POLITICAL STANCE DETECTION ON AMHARIC TEXT USING MACHINE LEARNING

A Thesis Presented

by

Surafel Tadesse Guda

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements
for the Degree of Master of Science

in

Computer Science

January, 2023

ACCEPTANCE

POLITICAL STANCE DETECTION ON AMHARIC TEXT USING MACHINE LEARNING

By

Surafel Tadesse Guda

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

____

Internal Examiner

____

External Examiner

____

Dean, Faculty of Informatics

January 2023

DECLARATION

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

___Surafel Tadesse Guda__
Full Name of Student

____
Signature

Addis Ababa

Ethiopia

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

____
Full Name of Advisor

____
Signature

Addis Ababa

Ethiopia

January 2023

Acknowledgment

First of all, I would like to thank the Almighty GOD for giving me the strength, courage, and patience to accomplish this research.

Secondly, I would like to express my deepest gratitude to my advisor Dr. Alemebante M. on for his generous help, continual advice, constructive comments, and suggestions throughout the preparation of this research.

Finally, I would like to express my gratitude to my wife, Hana Delka and My Friend Mr. Abenezer Desta, who helped me in the process of data annotation and encouraged me to face the challenges with enthusiasm to complete this research properly.

Contents

List of Acronyms

AI	Artificial Intelligence
BOW	Bag-of-Words
CBOW	Continuous Bag of Words
DSR	Design Science Research
LDA	Latent Dirichlet Allocation
LSTM	Long short-term memory
ML	Machine Learning
NLP	Natural Language Processing
RNN	Recurrent Neural Network
TF	Term Frequency
TF-IDF	Term Frequency Inverse Document Frequency
URL	Uniform Resource Locator
SVM	Support Vector Machine
LR	Logistic Regression
RF	Random Forest

List of Figures

List of Tables

Abstract

Technology advancements, such as social media, are now essential tools for connecting with the rest of the world, including political figures, governments, and social media activists. Recently, people have used social media to express their opinions about a particular subject or target. There are numerous uses for stance classification or detection in the world of NLP. Such as automatic stance recognition of whether a community is for or against a specific point of view in relation to religious and political issues, either in favor or against the stated targets.in this study, we constructed our own dataset with a total of 3,126 comments, of which are targeted **Prosperity Party**. Once the data has been collected and annotated using annotation guidelines, after that, the data were preprocessed, and morphologically analyzed. Then, we have used 4 different types of feature extraction techniques: BOW, N-gram, TF-IDF and word2vec and we trained three different machine learning algorithms SVM, LR and RF using **e**ach feature extraction techniques. according to the results from the experiments, we achieved accuracy score of **0.82** using TF-IDF feature extraction and SVM**.** based on these results, we draw the conclusion that the political stance classifier performed better classification utilizing feature extraction techniques using TF-IDF and SVM machine learning algorithm.

Keywords – Stance, SVM, Natural Language Processing, Political Stance Classification, Stance Classification

CHAPTER ONE

INTRODUCTION

1.1. Background of the study

Nowadays, social media is becoming a key medium to communicate with the rest of the world, such as political leaders, governments, and activists operating within social media. According to DIGITAL 2022: ETHIOPIA, **29.83** million internet users in Ethiopia in January 2022. This is evidenced by the number of Ethiopian individuals who have created personal Facebook profiles, which reached 5.95 million early 2022. [1]

As a result, there are certainly more Facebook users, and higher political involvement occurs on social media. Naturally, social media platforms are quickly becoming Ethiopia's most popular instruments for political debate. Even though social media networking sites are a new mode of communication for political discussion and other social concerns, they can be easily adopted by a significant number of Ethiopian users. We have seen several Ethiopian social media political parties and activists with tens of thousands or more followers on various social media platforms such as Facebook, Twitter, and others. A significant component of social media is that it allows users to openly voice their opinions without interference. [2] [3]

Stance is defined as expressing the speaker's point of view and judgment on a given statement. Stance detection plays an important role in social media, especially in analytical research that measures public opinion on political and social issues. [4] The nature of these issues is usually controversial, with people expressing disagreement over distinct points. Social issues such as abortion, climate change, and feminism have become target themes for stance detection on social media. has been used frequently. Similarly, political issues such as referendums and elections have always been hot topics and have been used to study public opinion.

In general, the stance classification process is also known as point of view recognition, in which perspective is identified by expressing an attitude about the object of controversial issue, whereas stance classification is based on various personal, cultural and social. It is a sophisticated process related to aspects. For example, political stances depend on empirical behavior as indicated by. And also, social media users can directly express their opinions by posting about topics, or indirectly infer attitudes from interactions and preferences. [4], [5], [6]

Stance detection or stance classification in the field of NLP has a wide range of applications. such as Automatic stance recognition of whether a community supports or opposes a particular viewpoint related to religious and political topics. Stance identification has also been shown to be beneficial in the development of recommendation and market forecasting systems. Most notably, it has grown in relevance in the identification of rumors and fake news. [4], [7]

1.2. Motivation

Stance classification is a newly emerging research area in the field of natural language processing. Since this research area is new, limited works of literature exist, so, different social media platforms are available today, and people attempt to express their stance toward a certain topic by using these platforms. Analyzing and classifying the stance helps to support decision-making by knowing the current topics and identifying stances toward those topics, whether they are against or in favor.

1.3. Statement of the Problem

Recent advancements in mobile computing and the Internet have led to an increase in the usage of social networks to communicate and express all types of perspectives, which has enhanced engagement and idea exchange. While it offers such benefits. [8]

Stance classification has been mainly used to identify the attitude toward an entity under analysis and to allow the measurement of public opinion toward an event or entity. Analyzing the stance of the communities that are posted on social media platforms helps to understand and classify the stance in favor or against. [5], [6], [7], [8]

Nowadays, we have witnessed different problems arising in our country, like people being displaced from different regions due to civil war among different ethnic groups, unemployment, and the spiraling cost of living. Such problems are raised by not understanding people’s interests or viewpoints toward government policies and other public interests.

Due to context-specific linguistic, cultural, and technological differences, particularly in social media communication, most of the research on stance classification of social media to assess people's view point toward a particular topic has been concentrated on the English language and European languages. [9], [10], [11] As a result, the existing models and approaches for the majority of resource-rich languages cannot easily be adapted to Amharic. [12], so this study's aim is to use machine learning techniques to analyze various Amharic comments posted on the prosperity party page in order to comprehend people's attitudes regarding various issues. This study addressed the above problem by answering the following questions:

1. Which feature extraction technique can better represent the amharic political stance dataset?
2. Which machine learning approach and algorithm can better classify the stance?
3. To what extent proposed model is efficient?

1.4. Objective of the Study

1.4.1. General

The main goal of this research is to build a model that can classify social media users' political stance from a piece of Amharic text.

1.4.2. Specific Objectives

To achieve the general objective, the following specific objectives are identified:

- To review relevant previous studies to find suitable methods and techniques.
- To collect and prepare dataset.
- To identify and apply different feature extraction mechanisms that are useful for our classification.
- To develop an appropriate machine learning model for the proposed work.
- To compare and evaluate the generated classification model using different metrics.

1.5. Methodology

1.5.1. Research Design

DSR [13] , [14] is a problem-solving paradigm that aims to improve human knowledge through the creation of new artifacts. Simply said, DSR attempts to advance technology and scientific knowledge through the creation of novel objects that solve issues and enrich the environment in which they are realized. The outcomes of DSR include both freshly created objects and design knowledge (DK), which gives a more complete understanding of why the artifacts enhance (or disrupt) the relevant application contexts through design theories.

1.5.1.1. Design Science Research Processes

Activity 1. Problem identification and motivation.

This activity defines the specific research problem and justifies the value of a solution. Justifying the value of a solution.

Activity 2. Define the objectives for a solution

The objectives of a solution can be derived from the problem statement and knowledge of what is possible and practical.

Activity 3. Design and development

An artifact is created. Conceptually, a DSR artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact.

Activity 4. Demonstration

This activity demonstrates the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activities.

Activity 5. Evaluation

The evaluation measures how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from the use of the artifact in context. Depending on the nature of the problem venue and the artifact.

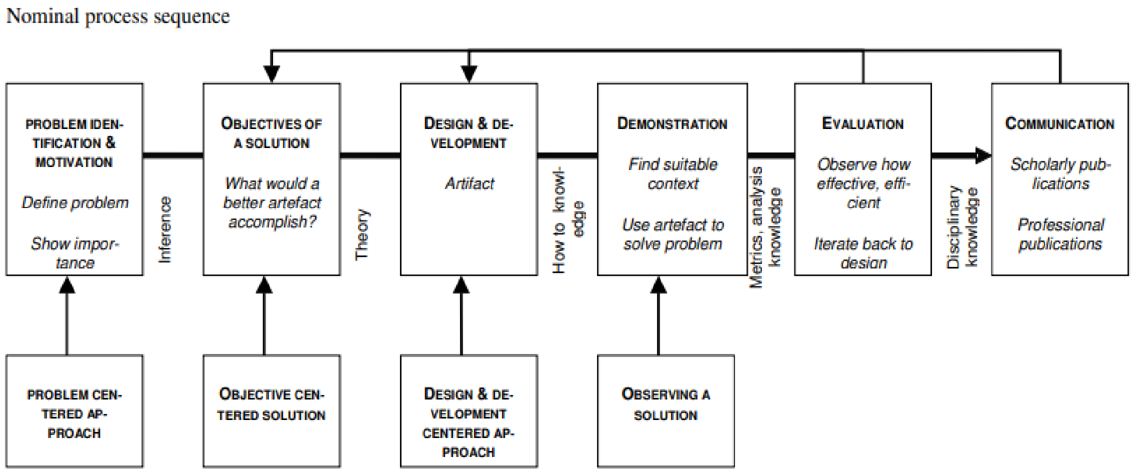


Figure 1-1 DSR Diagram [14]

1.5.2. Data Preparation

To examine the stances, we utilized datasets that contain ground truth labels for stances toward three topics The data were collected from different social media platforms using the tool Facepager, after the data collection the next step is the text pre-process step in this phase there are several steps to accomplish the task such as removal of unwanted characters, tokenization splitting sentences into words, stop word removal and stemming transformation of a word to its root form to reduce the computational cost using HornMorpho.

1.5.3. Tools and Techniques

in this study, a machine-learning technique is used to build the stance classification model, machine -learning technique can be from one of the three approaches. Supervised, unsupervised and semi-supervised, but a survey of stance detection studies shows that many scholars who used the supervised approach achieved a tremendous classification accuracy than other approaches. The supervised approach required a labeled dataset (a stance dataset is annotated using a predefined set of labels, either in favor or against) to predict new data classes based on its previously trained data. [5]

1.5.4. Evaluation

The four main metrics used to evaluate a classification model are accuracy, precision, recall and f1-score.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

Accuracy = Correct Predictions / All Predictions

Precision is defined as the fraction of relevant examples (true positives) among all of the examples which were predicted to belong in a certain class.

Precision = True Positives / (True Positives + False Positives)

Recall is defined as the fraction of examples that were predicted to belong to a class with respect to all of the examples that truly belong in the class.

Recall = True Positives / (True Positives + False Negatives)

F1-score is balancing precision and recall on the positive class.

F1-score = 2 * (Pression * Recall) / (Precision + Recall).

1.6. Scope of the Study

The study focuses only on detecting the political stance in Amharic language text using machine learning. A new dataset that is built by collecting Amharic text posts and comment from public Facebook page of prosperity party from 2021 to 2022 and annotating the post or comment into two different classes, which are in favor and against concerned with the target, **Prosperity parity** the study implements machine learning classifiers for the stance detection model and evaluates the result of the models based on the classification accuracy metrics.

1.7. Limitations of the study

This study comes across limitations in a different phase of the research process. Since there is a lack of other studies for comparison stance detection for the Amharic language, and also a lack of shared public dataset and model for Amharic stance detection.

- Our study not includes Emojis and idiomatic expression.
- Does not assess Latin Amharic language and multi-target stance.

1.8. Significance of the Study

1. It helps the government to improve the quality of services.
2. Used to make decisions.
3. Used to understand people's viewpoints toward a certain policy and help to identify and change the existing policies.
4. Help the government to have information about what are the current political questions or people's interests and answer those questions.

1.9. Thesis Organization

The study consists of six main chapters including this chapter. other chapters are organized as follows.

Chapter Two: deals with literature reviews on stance and stance detection.

Chapter Three: discusses the methodology and describes the general architectural design for Amharic stance detection.

Chapter Four: discusses the implementation of the stance detection model and the experiment and result discussion of the machine learning model are discussed.

Chapter Five: Conclusion and Recommendation of the research work with on possible future works related to this research are discussed.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

In this chapter, we review similar works to our research in the fields of machine learning and computational linguistics and methods and common text classification features for Stance classifications in prior works.

2.2. Amharic Language

Amharic, a Semitic language that belongs to the Afroasiatic languages family and is related to Ge,ez, is the official language of Ethiopia and the second most widely spoken Semitic language in the world after Arabic. also, Ethiopian. Each of the 33 fundamental characters has seven different variations depending on which vowel is to be pronounced in the syllable. Amharic was the only official language of Ethiopia up until 2023.

Amharic Alphabet

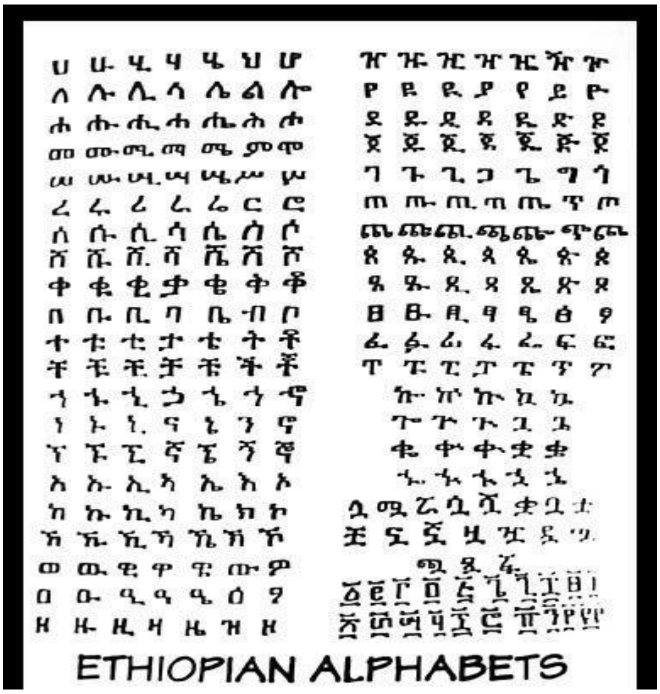


Figure 2-1 Ethiopian Field Characters [15]

Fidel, a Ge'ez script and alphabet where each character represents a consonant and vowel order in so much as the consonant dictates the form of the letter, makes it simple to write Amharic.

In Amharic, alphabets or letters are visible in a grid system where consonants appear vertically and vowels in a horizontal fashion. The grid below offers a simple and quick way of understanding Amharic and getting to comprehend it for the purpose of proper articulation of sounds. [16]

Despite this, the Amharic language is considered to be one of the "low-resource" languages in the world since it lacks the necessary tools and resources for NLP (natural language processing) and other techno-linguistic solutions. [17]

2.3. Natural Language Processing

NLP is a field of AI which is used for improving computers' ability to interpret human language. Databases are highly organized data types. The Internet, on the other hand, is fully unstructured with only a few structural components. In this scenario, the ultimate objective of NLP is to comprehend and model human language.

NLP tries to turn unstructured data into computer-readable language by utilizing natural language features. Complex algorithms are used by machines to break down any written material in order to extract useful information from it. The obtained data is then utilized to teach robots natural language reasoning. Natural language processing guides machines by finding and recognizing data patterns using syntactic and semantic analysis.

2.4. Stance Detection

The basic purpose of stance detection in sociolinguistics is to analyze the writer's point of view via their writing. The purpose of stance detection is to determine the writer's intrinsic viewpoint from the text by linking it to three factors: linguistic acts, social interactions, and human identity. The use of linguistic characteristics in stance detection is frequently associated with the usage of adjectives, adverbs, and lexical elements. [18]

Stance detection is a classification problem for a text input and a target pair in which the author's stance is sought in the form of a category label from this set: Favor, Against, Neither. Neutral is periodically added to the list of stance types, and the target may or may not be explicitly stated in the text. [5], [6], [19]

While sentiment analysis and stance identification are connected, they are not the same. Determine if a piece of text is positive, negative, or neutral. Determine the speaker's viewpoint and the target of the speaker's opinion from the text (the entity towards which opinion is expressed). In stance detection, on the other hand, requires systems to ascertain favorability towards a predetermined target of interest. It's possible that the text's aim of interest and target of opinion are not both addressed openly. [6], [19]

2.5. Stance detection according to target

The issue of establishing whether a given topic or entity has an in favor or against stance is known as stance classification. As a result, stance detection necessitates the presence of a predefined target in order to determine the attitude towards it. Stance detection may be divided into three categories based on the sort of target being evaluated.[5], [6], [19]

2.5.1. Multi-related-targets stance detection

The purpose of multi-target stance recognition is to understand the social media user orientation towards two or more targets for a single topic at the same time. The basic premise underlying this type of stance detection is that when a person presents his stance for one target, it conveys information about his attitude toward other connected targets. [18]

Multi-target stance detection is a classification problem for an input in the form of a piece of text and a group of related targets, in which the author's attitude is sought as a category label from this set Favor, Against, or neither for each target, and each stance categorization (for each target) may have an influence on the classifications for the other targets. [5]

Stance (T|U, Gn) = {(FavourG1, AgainstGn+1), (Favourof Gn+1, AgainstG1)}

2.5.2. Target-specific stance detection

The fundamental form of stance detection on social media may be created by employing the social actor's attributes. Thus, the key two inputs in this kind of stance detection are: Text T or user U, and Given Target G

The objective of analysis is the dependent factor in the stance identification task. One typical method in stance identification on social media is to infer the stance solely from the raw text, which reduces the stance detection issue to a textual entailment work. [18]

Stance (T, U|G) = {Favor,Against,None}

The primary approach for various stance research is target-specific stance identification; even for benchmark datasets such as SemEval stance 2016, which covers numerous themes, the majority of the published work on this dataset trained a distinct model for each topic (target) individually. [18], the below examples show target specific stance detection for the target that are used in this study.

Target: Prosperity Party

Comment: *“አሁን በሀገራችን እየተደረገ ያለው ሙሉ በሙሉ ትክክለኛ ብልፅግናን የሚያረጋግጥ ተግባር ነው ተስፋ አለን በቅርብ ግዜ ውስጥም የታዳጊዎች ሀገር ተርታ እንሰለፋለን”*

“What is being done in our country now is a complete proof of prosperity, it is a task. We hope that in the near future we will join the ranks of the developing countries.”

Stance: In favor

Target: Prosperity Party

Comment: *“የፈለጋችሁትን መልካም ሰው ሁሉ ብትሾሙ ይህ የጎሳ ፌደራሊዝምና ዘረኛ የህዋሀት ህገ መንግስት እያለ መተማመን አንችልም አንተማመንም በእኔ እምነት ህወሀት አድስ አበባ አራት ኪሎ ሆኖ ህግ ሆኖ እየመራን ነው”*

“Even if you appoint all the good people you want, as long as there exists the TPLF's tribal Federalism and Racist Constitution we cannot trust each other. In my opinion, TPLF is leading as a lawmaker in Addis Ababa at 4 Kilo and we don’t trust each other.”

Stance: Against

2.5.3. Claim-based stance detection

The focus of the analysis in claim-based, also known as open-domain attitude identification, is not an explicit entity like the ones outlined above, but rather a claim in a piece of news. The first stage in detecting the attitude is to determine the principal target assertion in the dialogue sequence or provided text. The claim (C), which might be the rumor’s post or the article title, is the major input to the claim-based stance identification algorithm. The prediction label sets usually take the form of either confirming or denying the assertion [18]

Stance (T, C) = {Confirming,Denying,Observing}

2.6. Stance Detection Approach

Classification/Detection is a process of categorizing a given set of data into classes. The process starts with predicting the class of given datapoint. the class are often referred to as target. Label or categories. According to several literature stance classification can be done is in to three approaches.

2.6.1. Supervised Learning

Supervised learning is used to explain prediction challenges since the objective is to forecast/classify a certain result of interest based on previously learned history. In this technique, the dataset is labeled using a preset set of classes that are either in favor of or against the given comment. Many research has been published utilizing this machine learning technique and various classification algorithms such as Naïve baye, SVM, Decision trees, and others. [20]

Machine learning (ML) is a subset of artificial intelligence (AI) that enables computer programs to grow increasingly effective at predicting outcomes without explicitly programming them to do so. Machine learning algorithms predict new output values by using past data as input. The following approaches are described in various stance detecting literature.

2.6.1.1. Support Vector Machine (SVM)

Support Vector Machine, or SVM, [36] is one of the most popular supervised learning algorithms. However, it is most frequently applied to classification and regression problems in machine learning. The SVM algorithm aims to generate the best decision line or boundary that can categorize the n-dimensional space, making it simple to add new data points in the future. The name of this ideal decision boundary is the hyperplane. SVM selects extreme points or vectors to construct a hyperplane. Because of the term "support vectors" that is used to describe these extreme cases, the algorithm is known as a support vector machine.

There are two types of SVM Linear SVM and Non-Linear SVM, Linear SVM is used for linearly separable data, which is defined as data that can be divided into two classes using just one straight line. The classifier used for such data is called the Linear SVM classifier. Whereas Non-Linear SVM is used for non-linearly separated data, which means that if a dataset cannot be classified using a straight line, it is considered to be non-linear data, and the classifier used is referred to as a Non-linear SVM classifier.

The best decision boundary for classifying the data points must be determined. There may be several lines or decision boundaries used to divide classes in n-dimensional space. This ideal boundary is referred to as the hyperplane of SVM. A straight line will be present if there is a hyperplane because the hyperplane's dimensions depend on the dataset's features. If there are three features, the hyperplane will be a two-dimensional plane. A hyperplane is always produced using the maximum margin, or the distance between the data points. Support Vectors: Support Vectors are the nearest data points or vectors to the hyperplane and those that affect the position of the hyperplane. These vectors are referred to as support vectors since they provide assistance to the hyperplane.

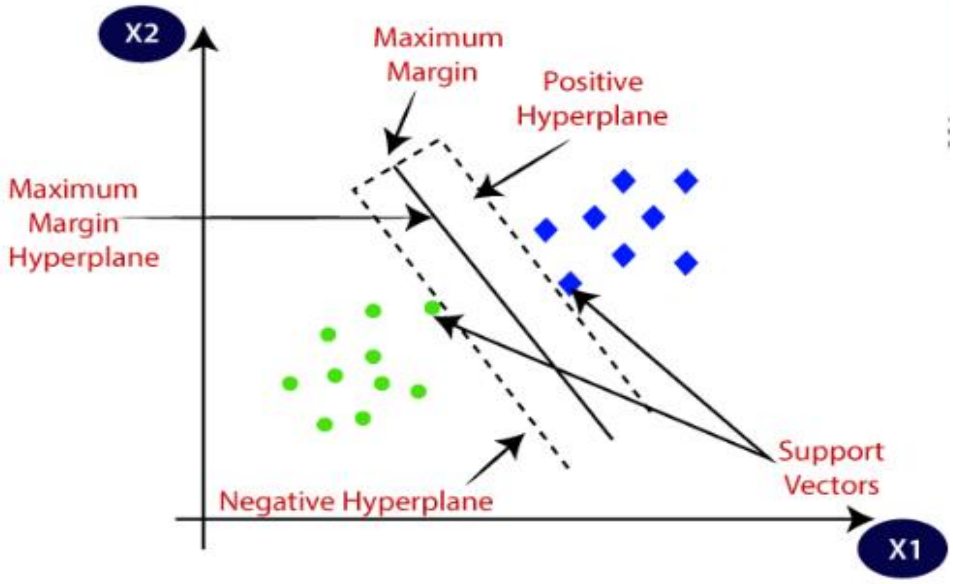


Figure 2-2 SVM [21]

2.6.1.2. Logistic regression

LR is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables and by analyzing the connection from a given collection of labelled data, it helps classifying data into distinct classes. It derives a linear connection from the provided dataset before introducing a non-linearity in the form of the Sigmoid function. [22] The authors [23] used Logistic regression algorithm for rumor stance classification by comparing the LR model with decision tree, random forest, and naïve bayes the model achieve accuracy score of 80%. In another study [24] the author tried to identify the public's stance on the bill on Twitter social media. The experiments were conducted using the Naïve Bayes, Support Vector Machine, and Logistic Regression, with unigram and bigram features and the Logistic Regression classifier using the unigram feature obtains a micro-F1 score of 71.8%.

2.6.1.3. Random Forest

Random forest is an ensemble of many decision trees. Random forests are built using a method called bagging in which each decision trees are used as parallel estimators and Samples from the original dataset are used as input for each tree in the classifications. The features used to build the tree at each node are then chosen at random. Until the exercise is complete and the forecast is clearly reached, no trees in the forest should be trimmed. In this manner, the random forest enables any weakly correlated classifier to produce a strong classifier. [25] the work [26] algorithm to open stance classification for rumor and veracity checking, and they tried to compare random forest with decision tree and IBk, and the random forest classifier achieved accuracy score of 79.02%

2.6.2. Unsupervised Learning

Unsupervised machine learning methods are very beneficial in description tasks since they try to uncover correlations in a data structure without having a quantifiable output. This type of machine learning is known as unsupervised because it lacks a response variable that may monitor the research and is usually used for clustering.[18]

Recently, there has been a focus on developing unsupervised stance detection algorithms. Clustering techniques are typically employed in this research, with an emphasis on user and subject representation on the social media platform. They developed an unsupervised model utilizing the clustering approach at the author and topic levels.

2.6.3. Transfer Learning

Some work in this field sought to combine unconstrained supervised approaches, such as transfer learning, weak-supervision, and distant supervision methods for stance identification, to solve the lack of labelled data for each target in the stance detection task. In transfer learning, an algorithm's knowledge from one task is applied to another, so that in the process of stance detection, transfer learning is used across many targets. [18]

Various works in this field attempted to apply transfer learning techniques to enrich the representation of targets in the dataset and improve overall stance detection performance to overcome the shortage of labeled data for each target in the stance identification problem. Transfer learning applies the information that an algorithm has acquired from one task to another, such that in task instance detection, transfer learning is used across distinct targets.

2.7. Feature Extraction Methods

Since different machine algorithms require a vector representation of a text as input, a feature is a numerical representation of a given data set. machine learning models can only understand numerical data, so we have to transform the text into some form of a numerical representation because features are relevant characteristics of the data that can be used during the analysis and this characterization is one of the biggest productions and also, it’s providing helpful information about the stored dataset by creating a meaningful future.

2.7.1. Bag-of-Words Model

The bag-of-words (BOW) model is a simplifying representation that represents a text as a bag of its words. This type of feature is called term frequency, the number of terms that appear in the text. As simple and clear as the BOW model is, it has some limitations. Like disregarding spatial information(grammar) and word order thus, the model fails to capture the semantic or syntactic relationship between two words. The output is document-term-matrix.

2.7.2. N-Gram Model

A (n 1)-order Markov model is used in an n-gram model, a sort of probabilistic language model, to predict the next item in a sequence. In probability, communication theory, and computational linguistics, n-gram models are currently often utilized. As an alternative to the BOW model, the N-Gram model can store spatial information in the text. By storing word co-occurrence information, n-gram models usually outperform BOW models.

2.7.3. TF-IDF

Term frequency-inverse document frequency (TF-IDF) is another statistical-based representation of text, term frequency is the number of terms that appear in the text. The BOW model and the N-gram model only consider term frequency, The TF-IDF model introduces Inverse Document Frequency, which measures how much information a word provides. By combing the term frequency, and inverse document frequency, TF-IDF reduces the effect of the commonly used words or stop words.

2.7.4. word2vec

word2vec is a word embedding strategy created in 2013 by Tomas Mikolov and other Google researchers for dealing with complicated NLP challenges. It can read through large amounts of text repeatedly to find word dependencies or connections.

word2vec uses the cosine similarity measure to detect similarities between words. If the cosine angle is one, it indicates that words are overlapping. If the cosine angle is 90, this indicates that the words are independent and have no contextual relationship. It pairs words with comparable vector representations.

2.9. Stance detection applications

The major use of stance detection has been to determine the attitude toward the object of investigation in order to gauge public opinion of a certain event or thing. [5], [6], [18] The discussion in this part, however, is focused on further stance detecting applications.

2.9.1. Analytical studies

Stance detection has proven its benefit as social sensing technique to measure public support related to social, religion and political topics. As examples of using stance detection for analyzing political topics are studies on analyzing public reaction towards certain political topics.

2.9.2. Applications related to social-based phenomena

On social media sites, stance detection has been utilized to address algorithmic problems. The social dynamics on these platforms are reflected in these problems. Echo chambers and homophily are the two most prevalent phenomena that are present on different social media sites.

The social phenomena known as homophily is concerned with people's propensity to make "like minded buddies." The cascade of specific knowledge among a group of people is what the echo-chamber is. Social media platforms have exacerbated this social behavior by amplifying particular opinions among small social networks. As a result, individuals are exposed to material that supports their own opinions. As a result, this reinforces the prejudices held by social media users and prevents them from seeing different points of view. In order to measure and address the issues caused by polarization on social media, stance detection has been deployed.

2.9.3. Veracity checking applications

People are encouraged to rely on social media platforms as their primary information source due to the quick distribution of news on these platforms. This type of information consumption raises serious concerns in social media about the veracity of the material shared, such as fake news and rumor identification. Classifying positions has received more attention recently as a means of establishing the first step in resolving the truthfulness verification problem.

2.10. Related Work

A probabilistic technique for stance identification was presented by J.Ebrahimi et al [9]. They employed a linear svm trained with word and character n-grams (n = 1 to 5). The authors provided a multi-way interaction log-linear model that incorporated stance, the target of stance, and tweet sentiment (STS). They wanted to be able to distinguish between emotional characteristics and target features when identifying the stance. They obtained an F-score of 71.03 percentile using the SemEval-2016 dataset.

In the field of stance detection, Liu et al [10]. suggested a supervised technique. Their RF-based method uses gradient decision trees to merge all of the included classifiers into an ensemble system that performs a decent job of obtaining minority classifications. While our method detects a wide spectrum of perspectives, it has trouble deciphering emotional political tweets. Furthermore, such tweets were topic-specific, with the same words perhaps having different meanings depending on the perspective chosen. because this model comprises a large number of irrelevant variables, the characteristics were chosen solely to aid in the assessment of any tweet with possible ambiguity difficulties. The reported F-score for the classification was 63.6%.

Using the SemEval-2016 dataset, Elfardy and Diab [11], suggested an svm-based supervised system that leverages lexical, sentiment, semantic, latent, and frame semantic variables to detect a tweeter's position for specified targets. They stated that the F-score in the competition was 63.6 percent.

Zotova [27], suggested a method based on TF-IDF and svm, with lemmatization used to construct the feature vector. Each word in the tweet is lemmatized; a word is eliminated if it lacks a root. To build the feature vector, the word list of each tweet is tokenized and mapped to the tf-idf score. The tf-idf scoring is based on the dictionary created from the training dataset's unigrams. The suggested method was tested on the SemEval-2016 dataset, with a reported F-score of 56%.

Schiller et al [28]. introduced a multi-dataset learning (MDL) method that uses the BERT architecture for its feature vector. BERT (Bidirectional Encoder Representations from Transformers) is an open-source NLP pre-training model created by Google. Ten datasets from five distinct areas were utilized to train their MDL model. When evaluated on the SemEval-2016 dataset, the model F-score was 71.62 percent.

In another work [29], an svm-based learning system learns in two phases; the first step in this system is to assess whether the tweets are subjective or objective to the position. The tweets are then categorized as either negative, positive, or neutral in the second stage. These stance detection models were built using a variety of models, including unigram and feature-based models. The F-scores obtained for unigram and feature-based models were 72.4 and 68 percent, respectively. suggested a tree kernel-based approach for tweet categorization that leverages tree representations. Their greatest performance F-score utilizing their kernel and senti-features model was 60.83 percent.

The dataset [30] used by the authors contains 300 claims and 2,595 related news articles collected by journalists and evaluated for their veracity (true, false, or unverified). labeled with. Each relevant article is summarized in a headline and flagged to indicate whether its stance is for, against, or observation of the claim. Observation shows that the article simply repeats the claims. As such, Emergent provides a real-world data source for a variety of natural language processing tasks in the context of fact checking. After presenting the dataset, we move on to determining the stance of the article title in relation to the claim. To do this, we use a logistic regression classifier to develop a function that looks at the headline and how it matches the claim. The achieved accuracy was 73%.

2.11. Research Gap and Summary

In this chapter showed an overview of stance modeling on social media. First, we explained the different types of targets when applying stance detection, namely: target specific, multi-related targets, and claim-based. Later, the most used features for modeling stance and different machine learning approaches were discussed and compared, and we observed that, most of the studies on stance detection indicate, supervised machine learning approach is most effective for stance classification algorithm.

From the analysis of related work, we understand most of the research on stance classification on social media to assess people's attitude toward a certain target has focused on the English language and European languages. People write their feeling on local languages like morphologically reach languages such as the Amharic language. To the best of our knowledge, there is no work done related to this domain. The paper tries to review some efforts that were taken in the area of political stance classification of Amharic texts. The machine learning model will highly depend on the domain that is being applied in Natural language processing. Learned models often have poor adaptability between domains or different text genres because they often rely on domain-specific features from their training data. [12] Because we can't apply the model applied in another domain easily to our domain, we have to create a dataset related to the political area. The main aim of this research is to fill the gap observed by collecting data sets specific to in the Amharic language and assess the people's stance towards the political that can be used mainly by the government in making effective strategies.

CHAPTER THREE

DESIGN AND METHODOLOGY

3.1. Introduction

This chapter discusses the proposed solution for political stance detection or classification using machine learning techniques. in this research we are prepared new stance dataset from social media and uses them as key inputs to build a proposed solution in political stance classification.

3.2. Data Collection

The data collected from Facebook platform from Prosperity Party official Facebook pages using Facepager, this tool was made for fetching public available data from YouTube, Twitter and other websites on the basis of APIs and web scraping. [31]

3.2.1. Dataset Annotation

Annotation is a procedure for adding information to the collected data or document at some level. In this case, the annotation process needs to label comments for building stance dataset. The study uses a simple random sampling technique to select comments to be annotated. The technique allows all the filtered comments on each page to get an equal chance to be annotated. The annotation conducted based on the instruction guideline provided by the researcher.

3.2.1.1. Stance Annotation Guideline

The guidelines provided to annotators for assessing stance. In our annotation work, we only consider the comment when labeling for stance toward a certain target. Consider whether the comment explicitly supports or opposes the target.

Target: [Target entity]

Comment: [Facebook comment]

Question: Which of the following comments regarding the user's stance toward the target is most likely to be accurate based on the comments?

The comment suggests that the user is in favor of the target. Any of the following might be the cause of this:

- The comment is explicitly in support for the target

- The comment is in support of something/someone aligned with the target, from which we can infer that the user supports the target
- The comment is against something/someone other than the target, from which we can infer that the user supports the target
- The comment does not in favor or oppose anything, yet it contains facts from which, if we can estimate that the person is in favor of the target.
- if we are unable to determine the user's stance but, the comment is supporting another person's stance towards the target.

The comment suggests that the user is against of the target. Any of the following might be the cause of this:

- The comment is explicitly against the target
- The comment is opposed of something/someone aligned with the target, from which we can infer that the user against the target
- The comment does not in favor or oppose anything, yet it contains facts from which, if we can estimate that the person is against of the target.
- If we are unable to determine the user's stance but, the comment is supporting another person's stance towards the target.

3.3. Preprocessing

Preprocessing is the first step once we have collected and annotated the dataset. Annotated datasets are manually labeled. This preprocessing involves preparing the input text in a format suitable for further analysis. The preprocessing stages of the architecture include the procedures for cleaning, normalizing, balancing the dataset, and encoding the dataset.

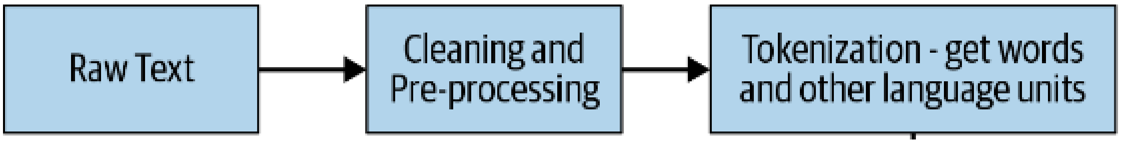

Raw Text
Cleaning and Pre-processing
Tokenization - get words and other language units

Figure 3-1 Preprocessing [37]

3.3.1. Data Cleaning

Text preparation involves cleaning the extracted data before stance detection is performed. It involves in identifying and eliminating non-textual content from the textual dataset, and any information that can have a privacy issue like source URL, commenter’s name, commenter’s location, comment date, numbers of likes, comments that is not relevant to the area of study, comments given on different language like English, removing numbers, stripping whitespace, removing punctuation and stop words (The most frequently used stop-words in Amharic documents are *እና (And),ወደ (to),የ(of),እኔ (I),ውስጥ (in),እሱ (he),እሷ (she),የእኔ (mine), እነሱ (they),እኛ (we)* was removed from the textual dataset. Moreover, the following list of tasks are done during the cleaning process.

Data Cleaning Algorithm

Input *Sentence (sequence of characters)*

Output *Filtered Sentences*

Process

I. *Read each word from sentences*

II. *Split in to tokens*

III. *If the token is punctuation, stop word, digits or irrelevant Unicode character perform replacement.*

IV. *Concatenate the words and form the original sentences*

V. *Return clean sentences*

3.3.2. Tokenization

Tokenization is the process of splitting input text into units called tokens, each of which can be a word or anything else like a number or a punctuation mark. Tokenization is applied to the input text based on the Amharic language characteristics used to retrieve the documents.

Tokenization Algorithm

Input *Words (sequence of characters)*

Output *Words*

Process

I. *Read each sentence from list*

II. *Then split them in to words*

III. *Return list of words*

3.3.3. Normalization

Since Amharic language has some characters that represent the same sound, use of these characters differs from user to user. Such characters include: ***ሀ, ሃ, ሐ, ሓ, ኀ, ኃ, ኻ, አ, ኣ, ዐ, ዓ, ሰ, ሠ, ጸ, ፀ.*** For example, ***ሃ, ሐ, ሓ, ኀ, ኃ*** and ***ኻ*** are replaced by ***ሀ*** and ***ሑ, ኁ, ኹ*** are replaced by ***ሁ***. and others. In order to exclude such variations in processing the Amharic text, one character representation for the same sound is necessary, so such representation was used for the text included in the experiment.

Normalization Algorithm

Input *Words*

Output *Normalized Characters*

Process

```
    I.    Read each word
   II.    Break the words in to characters
  III.    If the character is ኀ, ሐ then replaces with ሀ
          Else if the character is ሠ then replaces with ሰ
          Else if the character is ዓ, ዐ then replaces with አ
          Else if the character is ፀ then replaces with ጸ

              End if

    I.    Provide the normalized term
```

3.4. Morphological Processing

Morphological processing is the process of finding the constituent morphemes in a word. In order to perform this task, we use HORNMORPHO [32] is a Python program that performs morphology analysis and generation of words in three languages of the Horn of Africa: Amharic (አማርኛ), Oromo (Afaan Oromoo, Oromiffa), and Tigrinya (ትግርኛ). That is, it analyzes words into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the word's grammatical structure.

Morphological Processing Algorithm

Input *Words*

Output *Root word*

Process

I. *Read each word*

II. *Pass the word into Horn morpho*

III. *Select the root word from segmented word generated by Horn morpho*

IV. *Return the word*

3.5. Feature Extraction

Feature extraction methods based on state-of-the-art text mining; techniques applied for reducing redundant features and dimensionality and the main step in a machine learning text classifier is to transform the text into a numerical representation, usually a vector. The process of automatic feature extraction uses preprocessed text as an input. The input is large labeled data, and this data is tokenized, preprocessed and features are extracted before it's used for training. This stage is where the dataset is transformed into a vector of numbers. The output from this stage is a fixed-size vector representation for each word using BOW, N-gram, TF-IDF and word2vec model is used for generating word vectors.

3.5.1. Bag-of-Words Model

A representation of text called BOW counts the existence of known terms and builds a vocabulary of those words to explain the occurrences of those words inside a text.

The term "bag of words" refers to the fact that any information on the arrangement or structure of the words inside the document is ignored. The model doesn't care where in the document recognized terms appear; it is simply interested in whether they do. The decision of how to create the vocabulary of known words (or tokens) and how to evaluate the existence of known words is complicated.

BOW Algorithm

Input *Words*

Output *Set of vectors*

Process

I. *The comments will be divided to word level and morphological processing will be performed at word level and count the word frequency*

II. *Managing Vocabulary*

III. *Scoring Words*

IV. *Return Vectorize vocab*

3.5.2. N-Gram Model

There is a way to build a vocabulary of grouped terms. As a result, the bag-of-words is able to extract a little bit more meaning from the text. In this method, every word or token is referred to as a gram. A bigram model is the process of developing a vocabulary comprising two-word pairs. A word sequence of n tokens is known as an n-gram.

3.5.3. TF-IDF

Statistical method that reflects the importance of a word in a document or a set of documents. TF-IDF consists of two parts, Term Frequency (TF), which measures how frequently a word occurs in a document and Inverse Document Frequency (IDF), which measures how important a word is. Final score is then computed as:

$$TF_w = \frac{\text{number of times word } w \text{ appears in document}}{\text{total number of words in the document}}$$

$$IDF_w = \log_e \frac{\text{total number of documents}}{\text{number of documents with word } w \text{ in it}}$$

And the final score is computed as **TFw × IDFw**.

Second part of the equation (IDF) will effectively zero the probability for words occurring in most of the documents. And then the final TF-IDF score for words is a good measure of importance. [33]

TF-IDF Algorithm

Input *Words*

Output *Set of vectors*

Process

V. *The comments will be divided to word level and morphological processing will be performed at word level and count the word frequency*

VI. *Find TF for words*

VII. *Find IDF for words*

*VIII. And perform TF * IDF*

IX. Return Vectorize vocab

3.5.4. word2vec

This algorithm is a type of word embedding algorithm that converts every word in a text into vectors that convey the semantic meaning of the word. Word2vec is one of the neural network-based unsupervised learning applications. It is made up of a projection layer and a fully connected layer that is trained using stochastic gradient descent and a backpropagation technique. The projection layer converts the word in the context of an n-gram into continuous vectors. Words that exist in the context of an N-gram concurrently or repeatedly have a tendency to be triggered by the same weight, resulting in a correlation between words.

Weights connect the input layer with the projection layer as well as connect the projection layer with the output layer. Weights between the input layer and projection layer are represented by the V x N size W matrix, where V is the dimension of the input layer and N is the dimension of the projection layer. Between the projection layer and the output layer, matrix W is represented by a matrix measuring N x D, where N is the dimension of the projection layer and D is the dimension of the output layer. Word2vec relies on local information from the language The semantics learned from a particular word are influenced by the surrounding words. Wod2vec demonstrates the ability to study linguistic patterns as linear relationships between word vectors.

Two architectural models that can be used in word2vec, namely Continuous Bag-of-Word (CBOW) and Skip-Gram.

CBOW model word2vec uses words that are in preceding and following the target word and is limited to a window predicting the target word.

SKIP-GRAM model uses a word to predict words that are before and after the word that is limited by the window. A window is used as a kernel to obtain input and target words. The window is shifted from the beginning to the end of the wording. As an example, when the window size is given at 2, then word2vec will consider 2 words in before and 2 words after a word associated with it. [33]

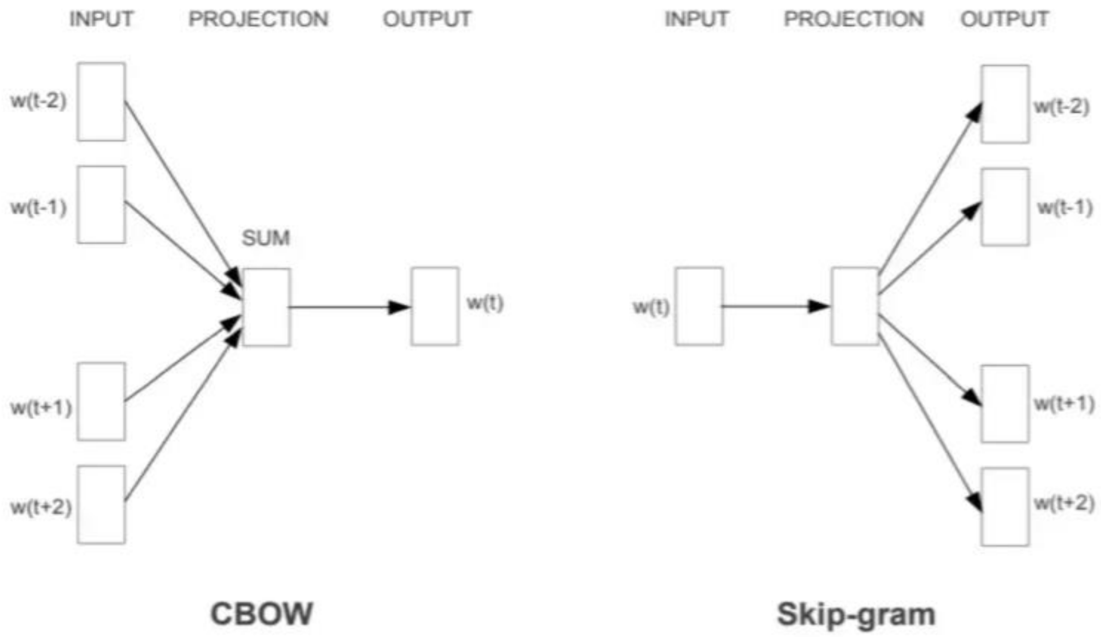


Figure 3-2 word2vec approach [34]

Word2Vec Algorithm

Input *Words*

Output *Set of vectors*

Process

I. *The comments will be divided to word level and morphological processing will be performed at word level*

II. *The word will be given as one-hot vectors, the vector length will be similar with the vocabulary length. The vector filled with 0's except the index that represents the word we want to represent, which is assigned 1.*

III. *The one hot encoded vector will go through hidden layer whose weights are the word embeddings. The weights are adjusted by minimizing the loss function and two edges exist between nodes(words) only if their corresponding vocabulary co-occurs in a window of length K, where K represents the size of the window.*

IV. *Outputs the candidate key words from vocabulary*

3.6. Machine Learning

There are a lot of state-of-the-art classification algorithm available for making stance classification including machine learning as well as deep learning models. But most of the studies on stance detection using supervised methods using **SVM, Logistic Regression and Random Forest** were found to be the most effective for different datasets. [18].

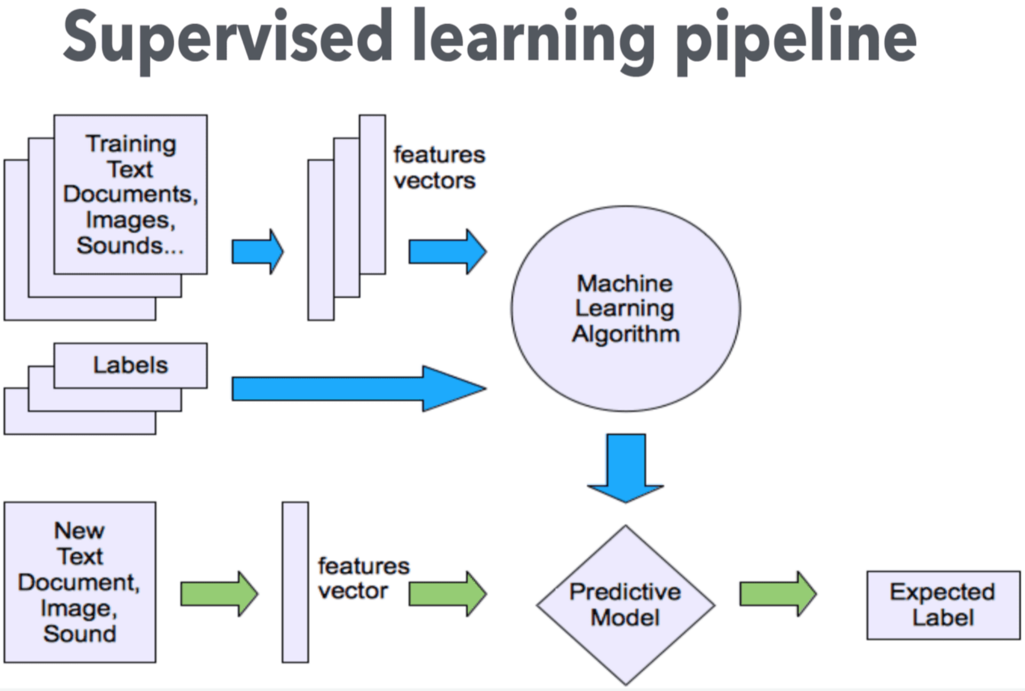

Supervised learning pipeline
Training Text Documents, Images, Sounds...
features vectors
Labels
Machine Learning Algorithm
New Text Document, Image, Sound
features vector
Predictive Model
Expected Label

Figure 3-3 Supervised Machine Learning [35]

3.7. Proposed Political Stance Detection Model Architecture

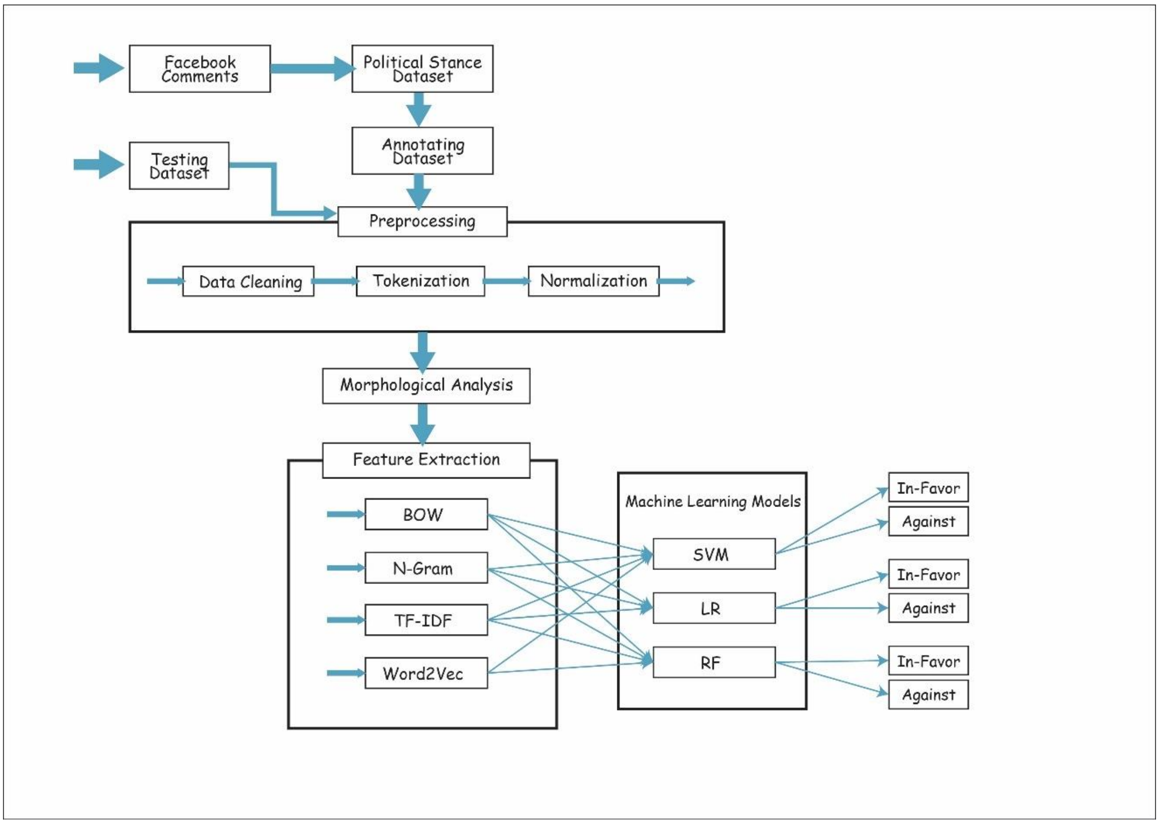


Figure 3-4 Stance Classification Model Architecture

3.8. Classification Tasks

In the analysis, the comments serve as a dataset. During the training phase, we used our data to progressively enhance our model's capacity to predict whether the submitted comments are in favor of or against the given target. We must send the feature extracted comment data with their labels to the machine learning model for training (in favor, against). The machine learning algorithm then attempts to comprehend the patterns necessary to categorize the comment to our target (output). The model will then be fed unseen data to anticipate the needed label. Unseen comments will be represented as features, and the machine learning model will classify them.

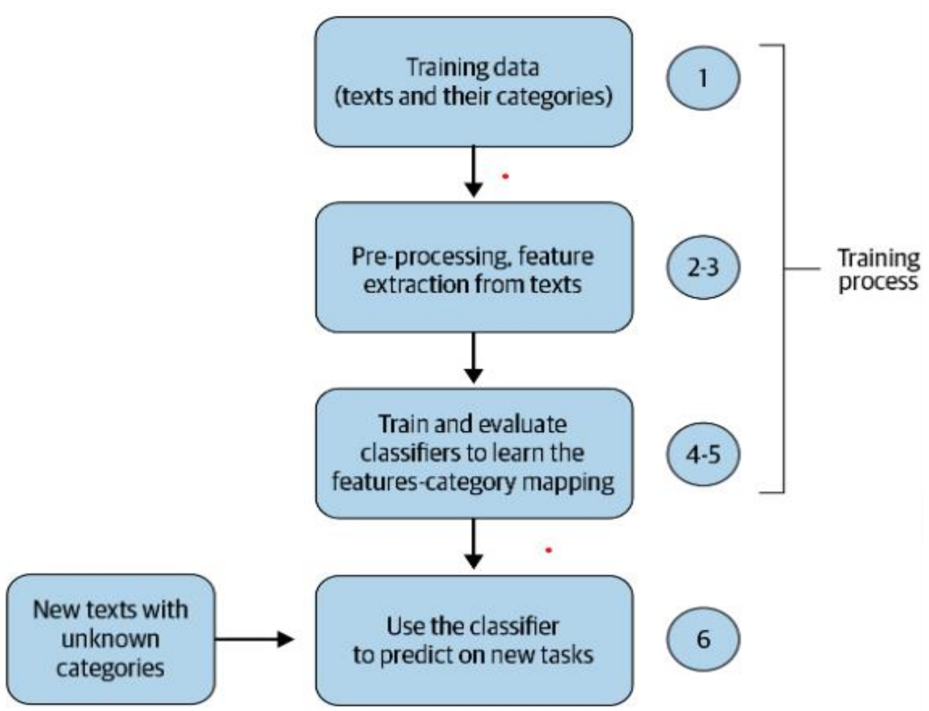

Training data
(texts and their categories)
1
Pre-processing, feature extraction from texts
2-3
Train and evaluate classifiers to learn the features-category mapping
4-5
Training process
New texts with unknown categories
Use the classifier to predict on new tasks
6

Figure 3-5 Classification Task [37]

3.8.1. Training Set

The training set is the actual dataset from which the model was trained. To forecast the result or arrive at the best decisions, the model observes and gains knowledge from this data. In this study, the Facebook data was the main source of training data, which was subsequently organized and preprocessed to ensure the model performed as intended. The model's capacity for generalization is strongly influenced by the nature of the training data. The model will perform better the higher the quality and diversity of the training data.

3.8.2. Testing Set

The dataset, which is used only after the model has finished being trained, is separate from the training set but shares some characteristics with it in terms of the probability distribution of the classes. A testing set is often a well-organized dataset including data from various scenarios that the model is likely to encounter when employed in the real world. It is not regarded as a good practice to frequently use the validation and testing set combined as a testing set. The model is considered to have overfitted if its accuracy on training data is higher than that on testing data.

3.8.3. Validation Set

The model's validation set, which is regarded as a component of model training, is used to fine-tune the model's hyperparameters. The model does not learn from this data; it merely uses it for evaluation, so it can be objectively and fairly evaluated. The validation dataset can also be used for regression by pausing model training when the loss of the validation dataset exceeds the loss of the training dataset, therefore minimizing bias and variance. the data is Approximately 10–20%.

3.9. Confusion matrix

The amount of accurate and wrong predictions made by a classifier is summarized in a confusion matrix, which is a table. It is employed to evaluate a classification model's effectiveness. By computing performance indicators like accuracy, precision, recall, and F1-score, it can be used to assess the effectiveness of a classification model. [38] [39]

The matrix is a 2*2 table for the classifiers' two prediction classes, a 3*3 table for the next three classes, and so on. The matrix is divided into two dimensions: the total number of predictions and the predicted values and actual values. The predicted values are those values that the model predicts, while the actual values are the real values for the provided observations. Since there were only two classes in this study, the confusion matrix was represented using a 2*2 matrix. [38] [39]

Table 3-1 confusion matrix example for binary classification

N = Total predictions	Actual: Against	Actual: in favor
Predicted: Against	True Negative	False Positive
Predicted: in favor	False Negative	True Positive

- **True Negative:** Model has given prediction against, and the real or actual value was also against.
- **True Positive:** The model has predicted in favor, and the actual value was also true.
- **False Negative:** The model has predicted against, but the actual value was in favor, it is also called as **Type-II error**.
- **False Positive:** The model has predicted in favor, but the actual value was against. It is also called a **Type-I error.**

3.10. Evaluation Metrics

The evaluation metrics of precision, recall, and F-score (or F-measure) are frequently employed. F-score is a combination measure that can be determined using precision (P) and recall (R), with the option to assign these two metrics different weights

This evaluation metric of F-score is also commonly employed in two-way stance detection (into one of the two classes: Favor, Against,). The most widely-used version of F-score is calculated as the macro-average of the F-scores for Favor and Against classifications as follows.

$$F = \frac{F_{Favor} + F_{Against}}{2} \qquad F_{Favor} = \frac{2 * P_{Favor} * R_{Favor}}{P_{Favor} + R_{Favor}} \qquad F_{Against} = \frac{2 * P_{Against} * R_{Against}}{P_{Against} + R_{Against}}$$

$$P_{Favor} = \frac{Correct_{Favor}}{Correct_{Favor} + Spurious_{Favor}} \qquad P_{Against} = \frac{Correct_{Against}}{Correct_{Against} + Spurious_{Against}}$$

$$R_{Favor} = \frac{Correct_{Favor}}{Correct_{Favor} + Missing_{Favor}} \qquad R_{Against} = \frac{Correct_{Against}}{Correct_{Against} + Missing_{Against}}$$

Accuracy is another metric used for stance detection and is calculated as follows:

$$Accuracy = \frac{Correct\ classifications}{All\ classifications}$$

3.11. Summary

Since we are employing the design science approach, the methodology section's primary result is the creation of artifacts. The artifact will be utilized to create stance classification model, using a machine learning model, predict people's comments toward the specific target called **Prosperity Party.** The procedure involves gathering data from the Facebook platform, which is followed by rigorous data preparation operations including data filtering, tokenization, and text normalization to reduce the amount of data representation and improve the system's effectiveness. Following the completion of the data preprocessing and morphological analysis, the system extracts features using BOWS, N-GRAM, TF-IDF and word2vec before creating a machine learning model. so, selecting appropriate feature extraction technique and machine learning algorithms for this study helps for achieving our goal determined in the objective.

CHAPTER FOUR

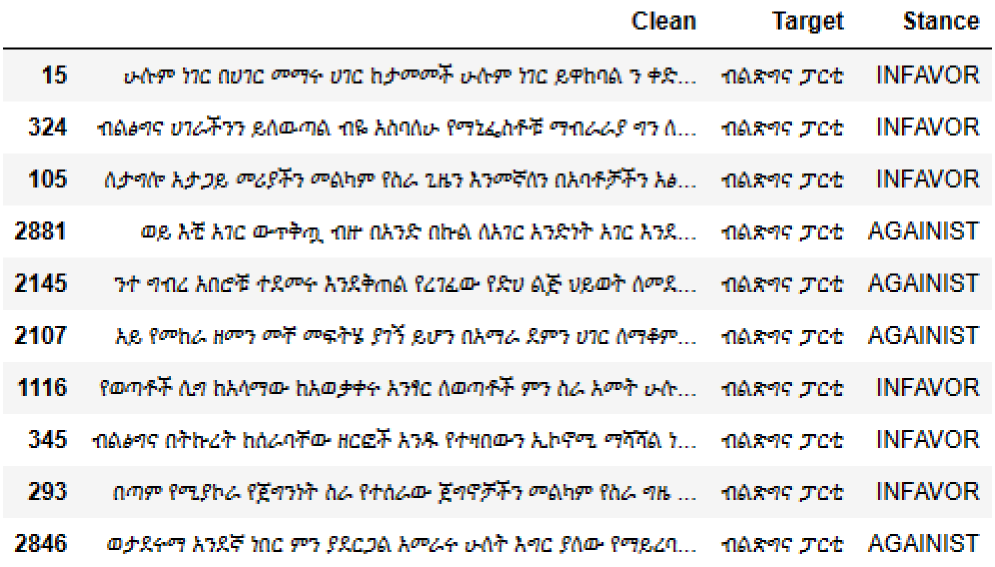
EXPERIMENTATION AND EVALUATION

4.1. Introduction

In this chapter, we use the approach covered in Chapter three, the proposed solution for Amharic Political Stance detection using a machine learning algorithm and find the ideal blend of the machine learning algorithm and feature extraction for the final models to build the artifacts, this study adopts a design science methodology. Furthermore, we used political stance dataset that are labeled with a binary-class (in favor or against).

4.2. Data Preparation

Facebook is by far the most popular platform, the Facebook page is used to share news articles with a wider audience, The corpus was collected manually and contains comments made on posts from 2021 to 2022 from prosperity party official Facebook page. Aside from the 6,000 comments target associate with **Prosperity party,** 3126 comments only extracted in preprocessing stages this research.



	Clean	Target	Stance
15	ሁሉም ነገር በሀገር መማሩ ሀገር ከታመመች ሁሉም ነገር ይዋከባል ን ቀድ...	ብልጽግና ፓርቲ	INFAVOR
324	ብልፅግና ሀገራችንን ይለውጣል ብዬ አስባለሁ የማኒፌስቶቹ ማብራሪያ ግን ለ...	ብልጽግና ፓርቲ	INFAVOR
105	ለታግሎ አታጋይ መሪያችን መልካም የስራ ጊዜን እንመኛለን በአባቶቻችን አፅ...	ብልጽግና ፓርቲ	INFAVOR
2881	ወይ እቺ አገር ውጥቅጧ ብዙ በአንድ በኩል ለአገር አንድነት አገር እንደ...	ብልጽግና ፓርቲ	AGAINIST
2145	ንተ ግብረ አበሮቹ ተደመሩ እንደቅጠል የረገፈው የድሀ ልጅ ህይወት ለመደ...	ብልጽግና ፓርቲ	AGAINIST
2107	አይ የመከራ ዘመን መቸ መፍትሄ ያገኝ ይሆን በአማራ ደምን ሀገር ለማቆም...	ብልጽግና ፓርቲ	AGAINIST
1116	የወጣቶች ሊግ ከአላማው ከአወቃቀሩ አንፃር ለወጣቶች ምን ስራ አመት ሁሉ...	ብልጽግና ፓርቲ	INFAVOR
345	ብልፅግና በትኩረት ከሰራባቸው ዘርፎች አንዱ የተዛበውን ኢኮኖሚ ማሻሻል ነ...	ብልጽግና ፓርቲ	INFAVOR
293	በጣም የሚያኮራ የጀግንነት ስራ የተሰራው ጀግኖቻችን መልካም የስራ ግዜ ...	ብልጽግና ፓርቲ	INFAVOR
2846	ወታደሩማ አንደኛ ነበር ምን ያደርጋል አመራሩ ሁለት እግር ያለው የማይረባ...	ብልጽግና ፓርቲ	AGAINIST

Figure 4-1 Sample Dataset

4.2.1. Defining our Label

Data annotation is the process of adding target variables to training data and labeling them so that a machine learning model can understand what predictions it is intended to make. This is one of the phases in the data preparation process for supervised machine learning. The labels utilized for the experiments in this study were “in favor” and “against”. The in-favor remarks toward the provided stance were represented by 1 and the against ones by 0. In this experiment, "in favor" refers to comments that support the provided target; "against" refers to comments that oppose the given target.

4.2.2. Manual Labeling

The activity is concerned with labeling the comments for experimental purposes. All **3126** comments are manually categorized by people into predefined categories against and in favor by using annotation guidelines discussed in chapter three. Since we employ supervised machine learning algorithms this step is crucial because the algorithm needs to learn from predefined labels then after identifying the relationship between the X (the comments) and y (stance) labels. This activity helped us to check whether the stance detection model properly categorize the comments accordingly to their labels.

4.2.3. Label Balancing

In this study, we will work with one dataset, the dataset contains 3126 comments with 1563 labeled as in favor and 1563 labeled as against for Prosperity Party target.

The Dataset [40] is balanced correct predictions are more important regardless of the d istribution of the various outcomes in the target feature. And this will help our stance d etection model to have good accuracy of prediction of unseen data.

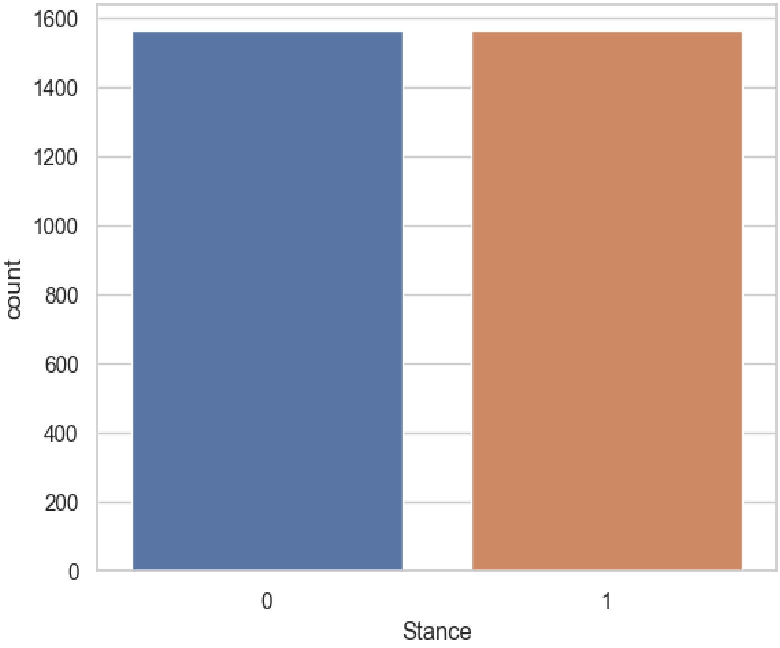


Figure 4-2 Distribution of the Dataset

4.3. Machine Learning Classifiers

One of the most important steps in NLP to be taken for a better comprehension of the context of what we are working with is feature extraction. After the initial text has been cleaned, it must be converted into its features so that it may be applied to modeling. comments must be converted into numerical data, such as a vector space model, as it cannot be computed directly. Feature extraction of document data is the common name for this transformation task. Because of the way they portray data in numerical form, feature extraction, also known as text representation, text extraction, or text vectorization, can have an impact on the machine learning model. The BOW, N-Gram, IF-IDF, and word2vec feature extraction methods are used. In Chapter two, we have discussed supervised machine learning algorithms and we will tried to compare each machine learning model by using evaluation metrics to compare the result for these experiments.

4.4. Feature Extraction

4.4.1. BOW

The BOW model estimates the frequency of occurrences of words in a text corpus. Bag of words is unconcerned with the sequence in which words appear in the text; rather, it is simply concerned with which words appear in the text. The words present in the manuscript are marked as 1 and the remaining as 0. We get a total of (3126, 2672) vector representations of the dataset when we use this feature extraction approach.

4.4.2. N-Gram

An N-Gram is a phrase that has a succession of N-words. N is an integer representing the number of words in the sequence. For example, if we enter N=1, the result is known as a uni-gram. If you enter N=2, it is a bi-gram. It becomes a tri-gram when N=3 is substituted. Bigram returns a total of (3126,6686) vector representations of the dataset using this feature extraction approach.

4.4.3. TF-IDF

This feature extraction method measures how important a particular word is with respect to a document and the entire dataset. Term Frequency is the measure of the counts of each word in a document out of all the words in the same document. IDF is a measure of the importance of a word, taking into consideration the frequency of the word throughout the corpus. It measures how important a word is for the corpus. using this feature extraction method gives us the total (3126, 3847) vector representations of the dataset.

4.4.4. word2vec

word2vec is word embedding techniques. The entire corpus is scanned, and the vector creation process is performed by determining which words the target word occurs with more often. In this way, the semantic closeness of the words to each other is also revealed, using this feature extraction method gives us the total (3126,200) vector representations of **Prosperity party** target.

Following feature extraction, the feature extracted representation of each comment used to train the machine learning algorithms. SVM, LR and RF is used for classification, as discussed in Chapter two. The experiment is built using Scikit-learn python library, one of the most well-known machine learning toolkits for classification. To train SVM, LR and RF and to measure the accuracy of each model.

4.5. Experimentation Result

The initial dataset was divided into two parts: training and testing, we used 80% of our dataset for training and 20% for testing (evaluation). According to the architecture of the proposed system, feature extraction was performed prior to the training phase in which machine learning was used. As a result, we created four distinct training and testing datasets for data extracted using Bag of words, N-gram, TF-IDF, and word2vec.

We train the machine learning models with those features to predict the class of 20% of the dataset based on 80% of the training dataset, and we observe the model's accuracy by comparing the prediction to the actual class of 20% of the test dataset, The experiment results obtained by SVM, LR and RF classifiers with all four-feature extraction are summarized below

Table 4-1 SVM Result

Feature	***SVM***				
	stance	**precision**	**recall**	**F1-score**	**accuracy**
BOW	0	0.77	0.71	0.74	0.75
	1	0.73	0.79	0.76	
N-GRAM	0	0.70	0.56	0.62	0.66
	1	0.63	0.76	0.69	
TF-IDF	0	0.81	0.84	0.82	**0.82**
	1	0.83	0.80	0.81	
Word2Vec	0	0.71	0.85	0.77	0.75
	1	0.81	0.64	0.72	

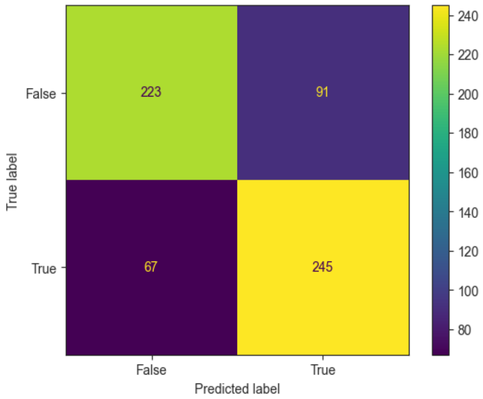


Figure 4-3 Confusion Matrix of BOW+SVM

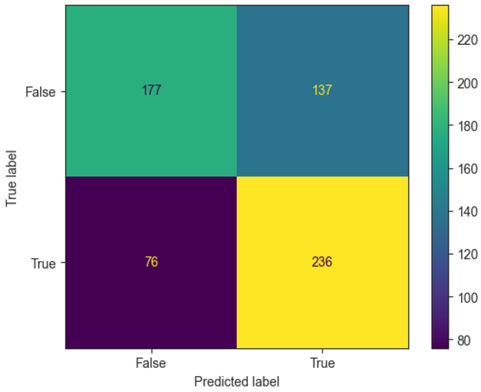


Figure 4-4 Confusion Matrix of Bi-gram+SVM

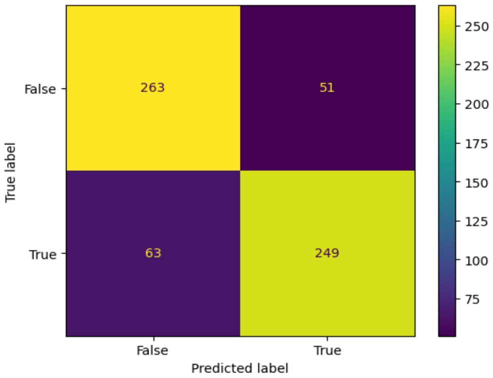


Figure 4-5 Confusion Matrix of TF-IDF+SVM

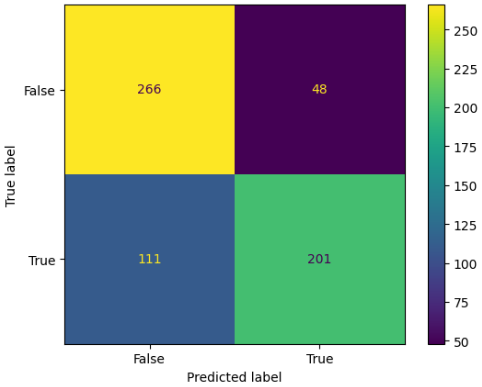


Figure 4-6 Word2vec+SVM

Table 4-2 LR Result

Feature	***LR***				
	stance	**precision**	**recall**	**F1-score**	**accuracy**
BOW	0	0.80	0.77	0.78	0.78
	1	0.77	0.80	0.79	
N-GRAM	0	0.71	0.61	0.66	0.68
	1	0.66	0.75	0.70	
TF-IDF	0	0.79	0.83	0.81	0.80
	1	0.82	0.78	0.80	
Word2Vec	0	0.76	0.77	0.77	0.77
	1	0.77	0.76	0.76	

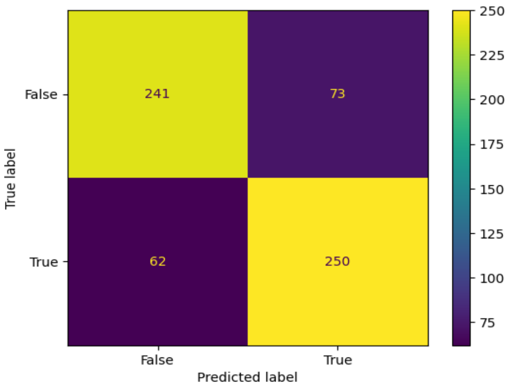


Figure 4-7 Confusion Matrix of BOW + LR

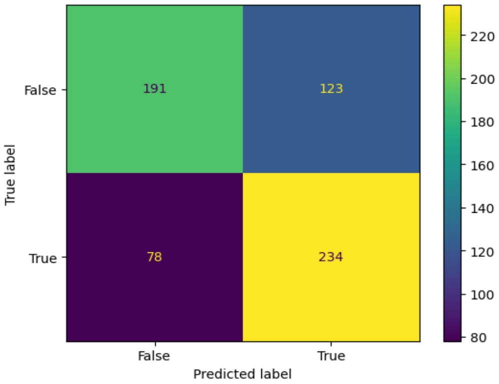


Figure 4-8 Confusion Matrix of Bi-gram + LR

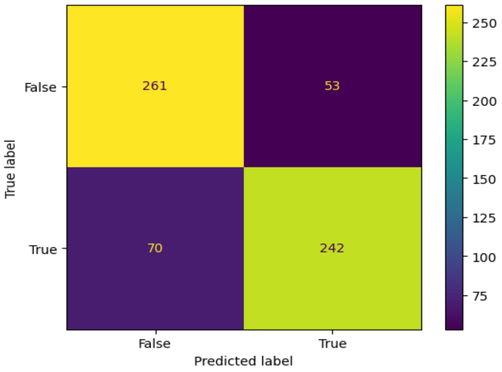


Figure 4-9 Confusion Matrix of TF-IDF + LR

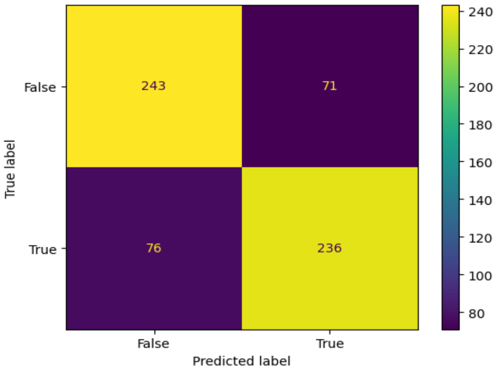


Figure 4-10 Confusion Matrix of word2vec + LR

Table 4-3 RF Result

Feature	***RF***				
	stance	**precision**	**recall**	**F1-score**	**accuracy**
BOW	0	0.77	0.83	0.80	0.78
	1	0.81	0.75	0.78	
N-GRAM	0	0.71	0.57	0.63	0.67
	1	0.64	0.77	0.70	
TF-IDF	0	0.76	0.83	0.80	0.78
	1	0.81	0.73	0.77	
Word2Vec	0	0.73	0.72	0.72	0.72
	1	0.72	0.73	0.72	

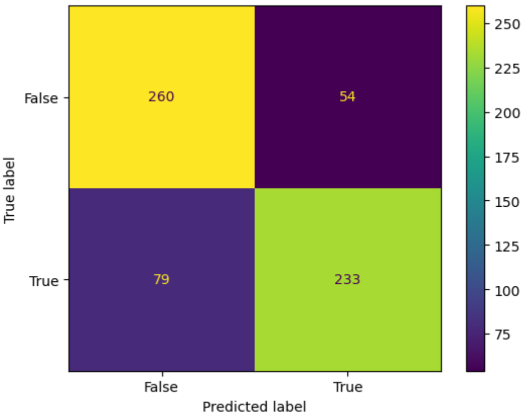


Figure 4-11 Confusion Matrix of BOW + RF

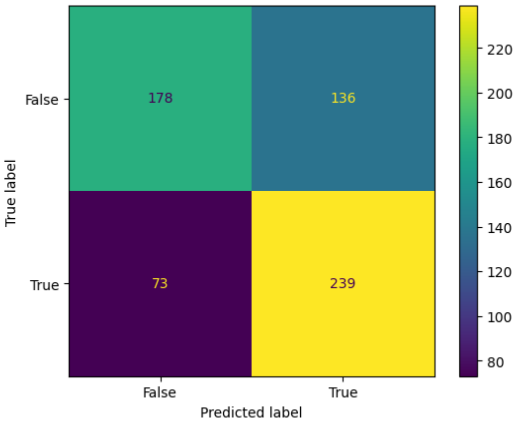


Figure 4-12 Confusion Matrix of Bi-gram + RF

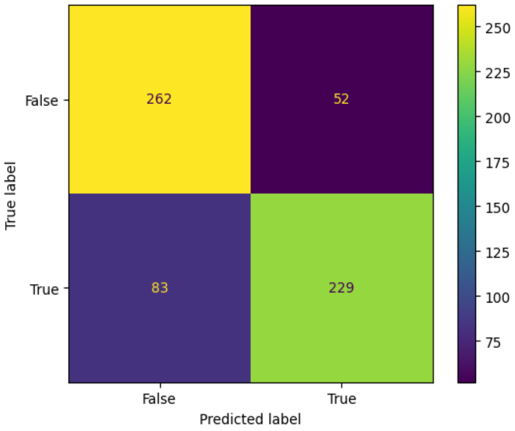


Figure 4-13 Confusion Matrix of TF-IDF + RF

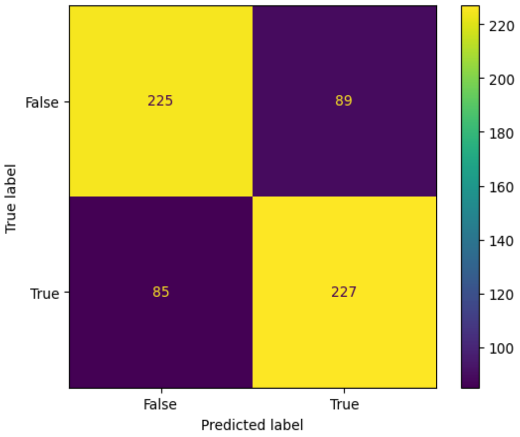


Figure 4-14 Confusion Matrix of word2vec + RF

4.6. Discussion of the Results

As we have presented the result in the previous section, we observed that different feature extraction method can affect the machine learning model, from the experiment the SVM machine learning model showed a better classification accuracy than the other two classification models. And we observed that the feature extractions technique can affect the machine learning model accuracy. For this study, TF-IDF feature extraction technique achieved a good representation of the dataset. As a result, the SVM trained with TF-IDF feature extracted dataset, the classification model achieved accuracy score of **82%.**

The experiment shows that TF-IDF feature extraction shows a better performance fitting the stance classification model using SVM with model accuracy of 82%, precession of 81%, recall of 84% and f1-score of 82% for classification of against, precession of 83%, recall of 80% and f1-score of 81% for classification of in favor. And from the experiment we sought that the stance classification model has good performance and the performance of the classification model depend on feature extraction method use to represent the target dataset. And when we look at the confusion matrix of classifiers, SVM stance classifier using TF-IDF feature extraction, showed a good result, according to the experiment from all three machine learning algorithms, SVM classification model has made a total of **626** predictions, out of 626 prediction **TF-IDF+SVM** classifier has made a **512** are true predictions, and **114** are incorrect.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

This thesis provides a comprehensive understanding of stance detection on social media. First, it covers the literature on stance detection on social media and provides an overview of the currently available approaches for handling stance modelling. Stance detection is important in analytical studies that measure public opinion on social media, especially on political and social subjects. In this study, we attempted to outline some preliminary techniques utilized to obtain comments from social media comment toward the **prosperity party,** once we prepare our dataset the next step is feature extraction, there are different techniques of feature extraction, but we have used **BOW, N-gram, TF-IDF** and **word2vec**, the output of this step is an input to the three different classification algorithms **SVM,LR** and **RF**, in this study, the classifiers were instantiated and trained with 80% training data. After building the model using the vector representation of the training dataset then we evaluate the model using the test set. in this study, TF-IDF feature extraction technique with SVM machine learning model achieved a better political stance classification with accuracy score of **82%**.

In this study we observed that, the feature extraction technique and data preprocessing are crucial for achieving good classification accuracy in machine learning algorithms. the overall performance of the models is determined using the scikit accuracy test function and scikit classification report. As a generalization, we may state that the experiment produced positive findings, however the research might be more engaging by using various feature extraction technique, deep learning and by utilizing additional datasets to increase accuracy.

5.2. Recommendation

As future work, the researchers recommended the following issues to be addressed in future work.

- Our model was trained using a small dataset; however, by adding additional targets, it is possible to extend the work to a larger dataset. As a result, the research can be enhanced by utilizing deep learning algorithms to produce more accurate models.
- Our model only trained using Ge’ez alphabets comment the research can be expanded by adding Latin Amharic alphabets.
- Our Classification model predict only single target stance, it is possible to extend the work by identifying multi target stance in a single comment.
- Transfer learning techniques to enrich the representation of targets in the dataset and improve overall stance detection performance to overcome the shortage of labeled data for each target in the stance identification problem.
- Exploring other word embedding techniques BERT, Glove, fastText word embedding feature extraction can achieve a good performance.
- Exploring and identifying more target or topics other than discussed in the study, using Automatically identify topic using Topic modeling from the comment using LDA.
- Emojis and idiomatic expression can be used in future research to evaluate people's opinions in addition to their stance because they can reveal vital details about the true meaning of the comment.

References

[1] S. Kemp, "Digital 2022: Ethiopia — DataReportal – Global Digital Insights," 04 Feb 2022. [Online]. Available: https://datareportal.com/reports/digital-2022-ethiopia. [Accessed 04 Feb 2023].

[2] A. Tazeb, "The Framing of Political Uprising on Social Media: The Case of Amhara and Oromiaa a Regional States," *Journalism and Communication Thesis and Dissertations The Framing of Political Uprising on Social Media: The Case of Amhara and Oromiaa a Regional States,* pp. 1-120, 2017.

[3] S. Lloyd, S. Lindberg and K. Tronvoll, ""The best and worst of times: The paradox of social media and Ethiopian politics,"," 26 Jun 2020. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/download/10862/10498. [Accessed 16 Nov 2022].

[4] D. Küçük and F. C. SIGIR, "Stance detection: Concepts, approaches, resources, and outstanding issues," in *Proceedings of the 44th International ACM; 2021*, 2021.

[5] D. Küçük and F. Can, "Stance detection: A survey," *Journal of ACM Computing Surveys,* vol. 53, no. 1, pp. 1-37, 2021.

[6] A. ALDayel and W. M. Management, "Stance detection on social media: State of the art and trends," *Information Processing Management,* vol. 58, no. 4, p. 102597, 2021.

[7] M. Hardalov, A. Arora, P. Nakov and I. Augenstein, "ASurvey on Stance Detection for Mis- and Disinformation Identification," *arXiv preprint arXiv:2103,* pp. 1-19, 8 May 2022.

[8] S. O. Edosomwan, "The history of social media and its impact on business," *Article in The Journal of Applied Management & Entrepreneurship,* vol. 16, no. 3, 2011.

[9] J. Ebrahimi, D. Dou and D. Lowd, "A joint sentiment-target-stance model for stance classification in tweets," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, USA, 2016.

[10] C. Liu, W. Li, B. Demarest, Y. Chen and S. Couture, "An Ensemble Model for Stance Detection in Twitter," in *Proceedings of SemEval-2016*, San Diego, 2016.

[11] H. Elfardy and M. Diab, "Ideological Stance Detection in Informal Text," in *Proceedings of SemEval-2016*, 2016.

[12] N. Alturayeif, H. Luqman and M. Ahmed, "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,* 2016.

[13] P. Offermann , O. Levina, M. Schönherr and U. Bub, "Outline of a design science research process," in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology,* 2009.

[14] K. Peffers, T. Tuunanen, C. E. Gengler, M. Rossi, W. Hui, V. Virtanen and J. Bragge, "Design Science Research Process: A Model for Producing and Presenting Information Systems Research," in *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology,* Claremont, CA, USA, 2006.

[15] pinterest, "Amharic language - official language of Ethiopia | Communication letter, Ethiopia, Alphabet," [Online]. Available: https://www.pinterest.com/pin/503558802058171204/. [Accessed 04 Feb 2023].

[16] amharicalphabet, "Amharic Language, Origin, History, & Characteristics | AmharicAlphabet," 2023. [Online]. Available: https://www.amharicalphabet.com/amharic-language. [Accessed 5 Feb 2023].

[17] F. Gereme, W. Zhu, T. Ayall and D. Alemu, "Combating Fake News in "Low-Resource" Languages: Amharic Fake News Detection Accompanied by Resource Crafting. Information 2021, 12, 20," *Information (Basel),* vol. 12, no. 1, p. 20, 15 Dec 2021.

[18] A. Aldayel, "Stance characterization and detection on social media," *Unpublished Ph.D dissertation ,Institute for Language, Cognition and Computation,University of Edinburgh,* pp. 1-179, 2021.

[19] D. Küçük and F. Can, "Stance Detection on Tweets: An SVM-based Approach," *arXiv preprint arXiv:1803.08910,* pp. -13, 3 2018.

[20] T. Jiang, J. L. Gradus and A. J. Rosellini, "Supervised machine learning: a brief primer," *Behavior Therapy,* vol. 51, pp. 675-687, 2020.

[21] skilltohire, "Support Vector Machines. Introduction to margins of separation," 26 Jul 2020. [Online]. Available: https://medium.com/@skilltohire/support-vector-machines-4d28a427ebd. [Accessed 13 Jan 2023].

[22] GeeksforGeeks, "Advantages and disadvantages of logistic regression," GeeksforGeeks, 25 Aug 2020. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/. [Accessed 04 Feb 2023].

[23] K. Xuan and R. Xia, "Rumor stance classification via machine learning with text, user and propagation features," in *International Conference on Data Mining Workshops*, China, 2019.

[24] A. H. Nababan, R. Mahendra and I. Budi, "Twitter stance detection towards Job Creation Bill," in *Information Systems International Conference*, 2021.

[25] CFI Team, "Random Forest - Overview, Modeling Predictions, Advantages," 20 Dec 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/random-forest/. [Accessed 4 Feb 2023].

[26] A. Aker, L. Derczynski and K. Bontcheva, "Simple Open Stance Classification for Rumour Analysis," *Recent Advances in Natural Language Processing Meet Deep Learning,* pp. 31-39, 11 2017.

[27] E. Zotova, R. Agerri and G. Rigau, "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.,* pp. 1-63, 2019.

[28] B. Schiller, J. Daxenberger and I. Gurevych, "Stance Detection Benchmark: How Robust is Your Stance Detection?," *Künstl Intell,* vol. 35, no. 3-4, pp. 329-341, 11 2021.

[29] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," *Coling 2010: Posters,* pp. 36-44, 2010.

[30] W. Ferreira and A. Vlachos, "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*, San Diego, 2016.

[31] J. Jünger and T. Keyling, " Facepager: Facepager was made for fetching public available data from YouTube, Twitter and other websites on the basis of APIs and webscraping," Facepager, 2019. [Online]. Available: https://github.com/strohne/Facepager.

[32] S. Vajjala, B. Manjumader, A. Gupta and H. Surana, Practical natural language processing: a comprehensive guide to building real-world NLP systems, USA: O'Reilly Media,Inc.., 2020.

[33] M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromon, and Tigrinya," pp. 1-7, 9 Aug 219.

[34] E. M. Dharma, F. Lumban Gaol, H. Leslie, H. S. Warnars and B. Soewito, "The accuracy comparison among Word2vec, Glove, and Fasttext towards

convolution neural network (CNN) text classification," *Journal of Theoretical and Applied Information Technology,* vol. 100, no. 2, pp. 1-11, 2022.

[35] R. Kulshrestha, "NLP 101: Word2Vec — Skip-gram and CBOW | by Ria Kulshrestha | Towards Data Science," 24 Nov 2019. [Online]. Available: https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314. [Accessed 16 Jan 2023].

[36] D. Ofer, "Machine Learning for Protein Function," *Unpublished Masters Thesis,Department of Computer Science, Hebrew University of Jerusalem,* pp. 1-76, 3 2015.

[37] L. Wang, Support Vector Machines:, Springer, Berlin, 2005.

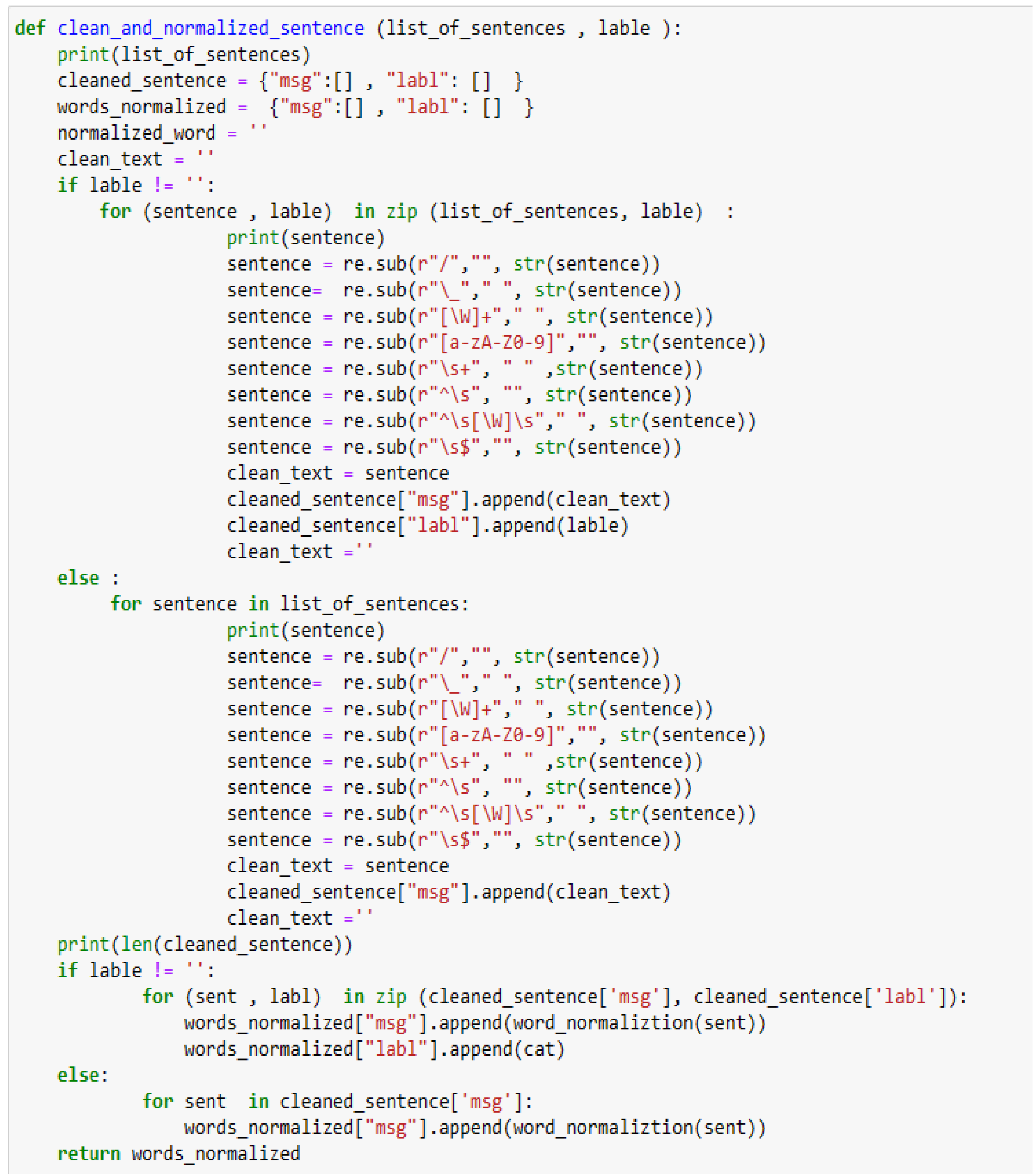
[38] S. Narkhede, "Understanding Confusion Matrix | by Sarang Narkhede | Towards Data Science," 9 May 2018. [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. [Accessed 17 Jan 2023].

[39] S. Jaiswal, "Confusion Matrix in Machine Learning - Javatpoint," 2021. [Online]. Available: https://www.javatpoint.com/confusion-matrix-in-machine-learning. [Accessed 17 Jan 2023].

[40] J. Cohen, "Machine Learning: Target Feature Label Imbalance Problems and Solutions | by Joseph Cohen | Towards Data Science," 26 Nov 2020. [Online]. Available: https://towardsdatascience.com/machine-learning-target-feature-label-imbalance-problem-and-solutions-98c5ae89ad0. [Accessed 11 Jan 2023].

Appendix A Sample Code

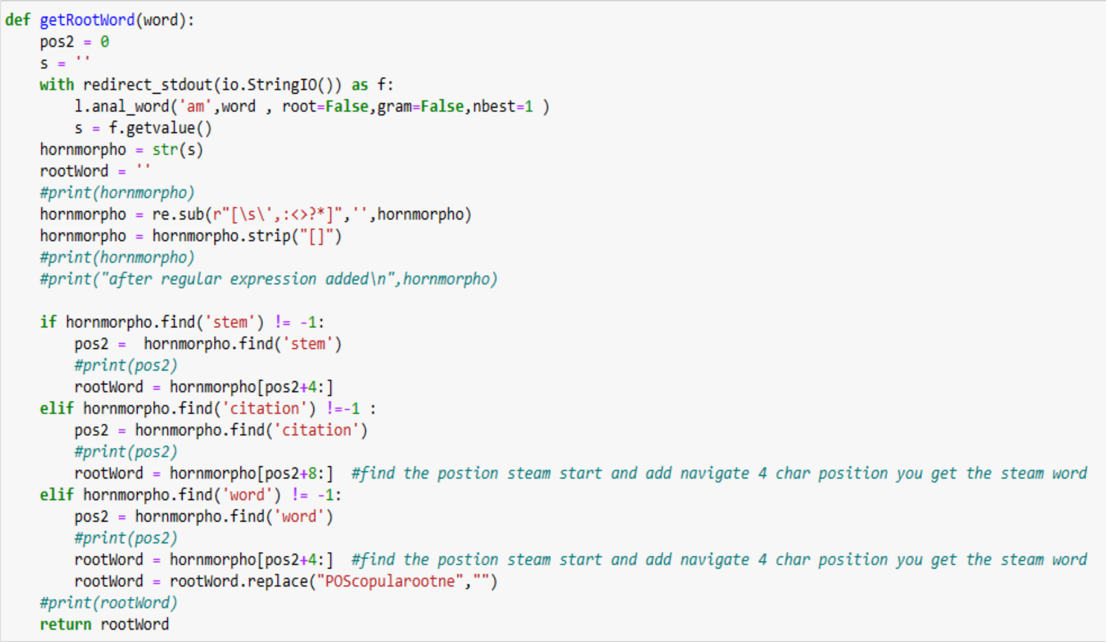
Data Cleaning and Normalization



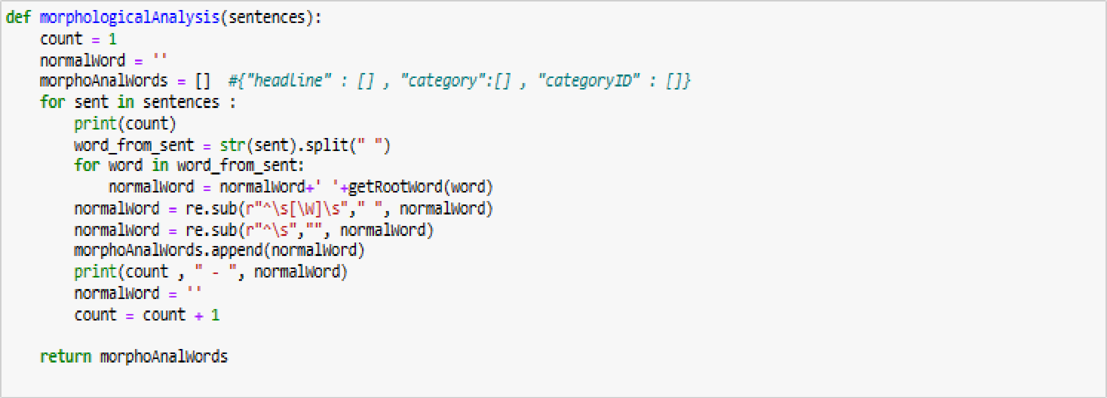
```
def clean_and_normalized_sentence (list_of_sentences , lable ):
    print(list_of_sentences)
    cleaned_sentence = {"msg":[] , "labl": []  }
    words_normalized =  {"msg":[] , "labl": []  }
    normalized_word = ''
    clean_text = ''
    if lable != '':
        for (sentence , lable)  in zip (list_of_sentences, lable)  :
                    print(sentence)
                    sentence = re.sub(r"/","", str(sentence))
                    sentence=  re.sub(r"\_"," ", str(sentence))
                    sentence = re.sub(r"[\W]+"," ", str(sentence))
                    sentence = re.sub(r"[a-zA-Z0-9]","", str(sentence))
                    sentence = re.sub(r"\s+", " " ,str(sentence))
                    sentence = re.sub(r"^\s", "", str(sentence))
                    sentence = re.sub(r"^\s[\W]\s"," ", str(sentence))
                    sentence = re.sub(r"\s$","", str(sentence))
                    clean_text = sentence
                    cleaned_sentence["msg"].append(clean_text)
                    cleaned_sentence["labl"].append(lable)
                    clean_text =''
    else :
         for sentence in list_of_sentences:
                    print(sentence)
                    sentence = re.sub(r"/","", str(sentence))
                    sentence=  re.sub(r"\_"," ", str(sentence))
                    sentence = re.sub(r"[\W]+"," ", str(sentence))
                    sentence = re.sub(r"[a-zA-Z0-9]","", str(sentence))
                    sentence = re.sub(r"\s+", " " ,str(sentence))
                    sentence = re.sub(r"^\s", "", str(sentence))
                    sentence = re.sub(r"^\s[\W]\s"," ", str(sentence))
                    sentence = re.sub(r"\s$","", str(sentence))
                    clean_text = sentence
                    cleaned_sentence["msg"].append(clean_text)
                    clean_text =''
    print(len(cleaned_sentence))
    if lable != '':
            for (sent , labl)  in zip (cleaned_sentence['msg'], cleaned_sentence['labl']):
                words_normalized["msg"].append(word_normaliztion(sent))
                words_normalized["labl"].append(cat)
    else:
            for sent  in cleaned_sentence['msg']:
                words_normalized["msg"].append(word_normaliztion(sent))
    return words_normalized
```

A-1

Find root word using Horn-morpho



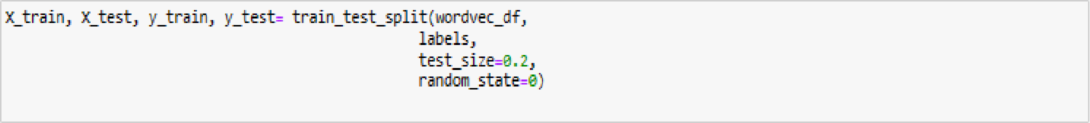
```python
def getRootWord(word):
    pos2 = 0
    s = ''
    with redirect_stdout(io.StringIO()) as f:
        l.anal_word('am',word , root=False,gram=False,nbest=1 )
        s = f.getvalue()
    hornmorpho = str(s)
    rootWord = ''
    #print(hornmorpho)
    hornmorpho = re.sub(r"[\s\',:<>?*]",'',hornmorpho)
    hornmorpho = hornmorpho.strip("[]")
    #print(hornmorpho)
    #print("after regular expression added\n",hornmorpho)

    if hornmorpho.find('stem') != -1:
        pos2 =  hornmorpho.find('stem')
        #print(pos2)
        rootWord = hornmorpho[pos2+4:]
    elif hornmorpho.find('citation') !=-1 :
        pos2 = hornmorpho.find('citation')
        #print(pos2)
        rootWord = hornmorpho[pos2+8:]  #find the postion steam start and add navigate 4 char position you get the steam word
    elif hornmorpho.find('word') != -1:
        pos2 = hornmorpho.find('word')
        #print(pos2)
        rootWord = hornmorpho[pos2+4:]  #find the postion steam start and add navigate 4 char position you get the steam word
        rootWord = rootWord.replace("POScopularootne","")
    #print(rootWord)
    return rootWord
```



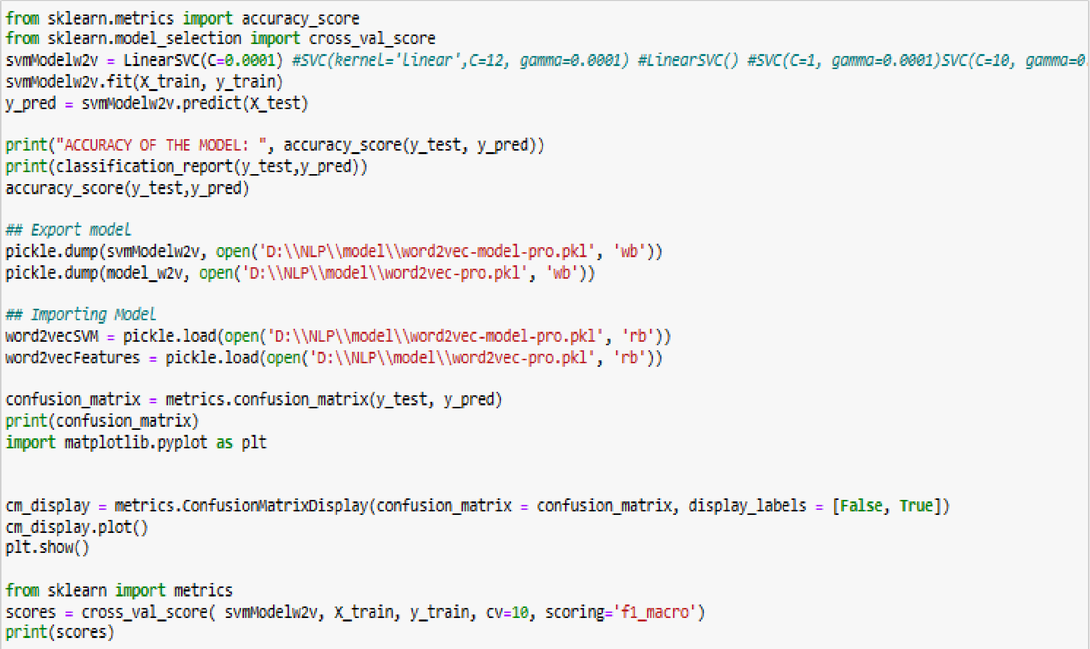
```python
def morphologicalAnalysis(sentences):
    count = 1
    normalWord = ''
    morphoAnalWords = []  #{"headline" : [] , "category":[] , "categoryID" : []}
    for sent in sentences :
        print(count)
        word_from_sent = str(sent).split(" ")
        for word in word_from_sent:
            normalWord = normalWord+' '+getRootWord(word)
        normalWord = re.sub(r"^\s[\W]\s"," ", normalWord)
        normalWord = re.sub(r"^\s","", normalWord)
        morphoAnalWords.append(normalWord)
        print(count , " - ", normalWord)
        normalWord = ''
        count = count + 1

    return morphoAnalWords
```

A-2

Sample code for Training SVM model using word2vec and performing k-fold cross validation

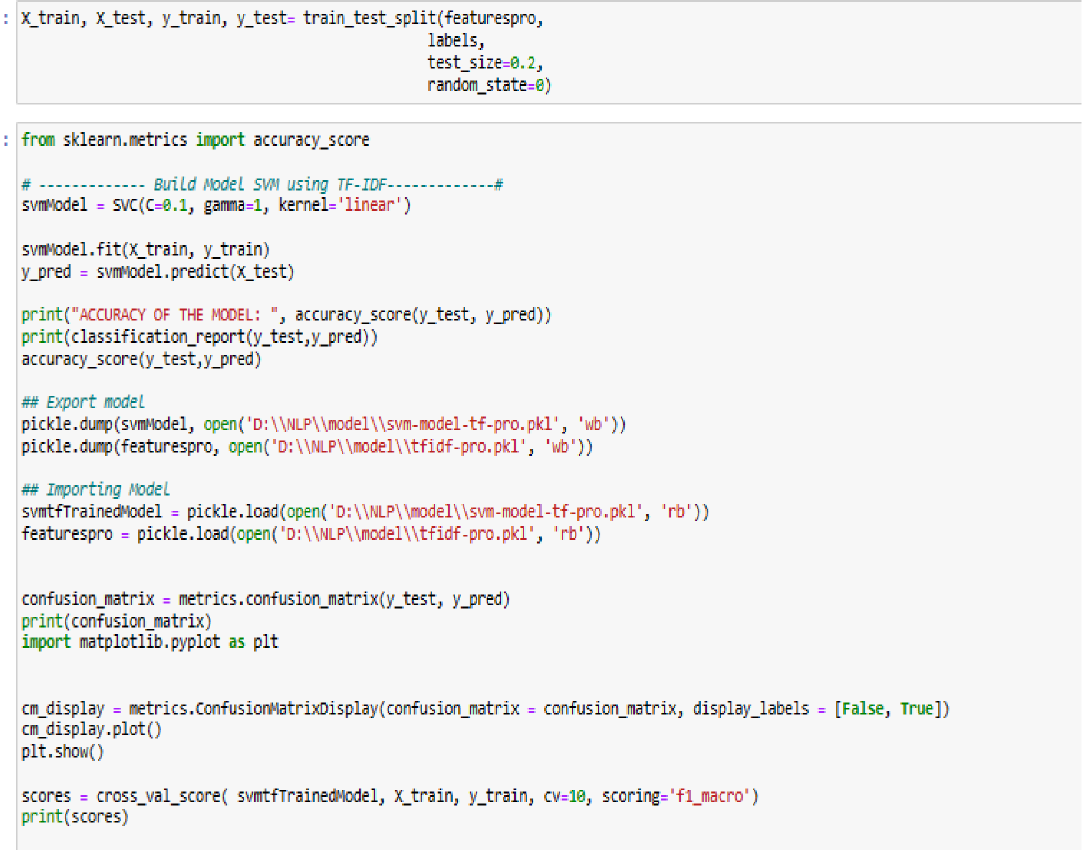


```
X_train, X_test, y_train, y_test= train_test_split(wordvec_df,
                                               labels,
                                               test_size=0.2,
                                               random_state=0)
```



```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
svmModelw2v = LinearSVC(C=0.0001) #SVC(kernel='linear',C=12, gamma=0.0001) #LinearSVC() #SVC(C=1, gamma=0.0001)SVC(C=10, gamma=0.
svmModelw2v.fit(X_train, y_train)
y_pred = svmModelw2v.predict(X_test)

print("ACCURACY OF THE MODEL: ", accuracy_score(y_test, y_pred))
print(classification_report(y_test,y_pred))
accuracy_score(y_test,y_pred)

## Export model
pickle.dump(svmModelw2v, open('D:\\NLP\\model\\word2vec-model-pro.pkl', 'wb'))
pickle.dump(model_w2v, open('D:\\NLP\\model\\word2vec-pro.pkl', 'wb'))

## Importing Model
word2vecSVM = pickle.load(open('D:\\NLP\\model\\word2vec-model-pro.pkl', 'rb'))
word2vecFeatures = pickle.load(open('D:\\NLP\\model\\word2vec-pro.pkl', 'rb'))

confusion_matrix = metrics.confusion_matrix(y_test, y_pred)
print(confusion_matrix)
import matplotlib.pyplot as plt


cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])
cm_display.plot()
plt.show()

from sklearn import metrics
scores = cross_val_score( svmModelw2v, X_train, y_train, cv=10, scoring='f1_macro')
print(scores)
```

Sample code for Training SVM model with TF-IDF and performing k-fold cross validation

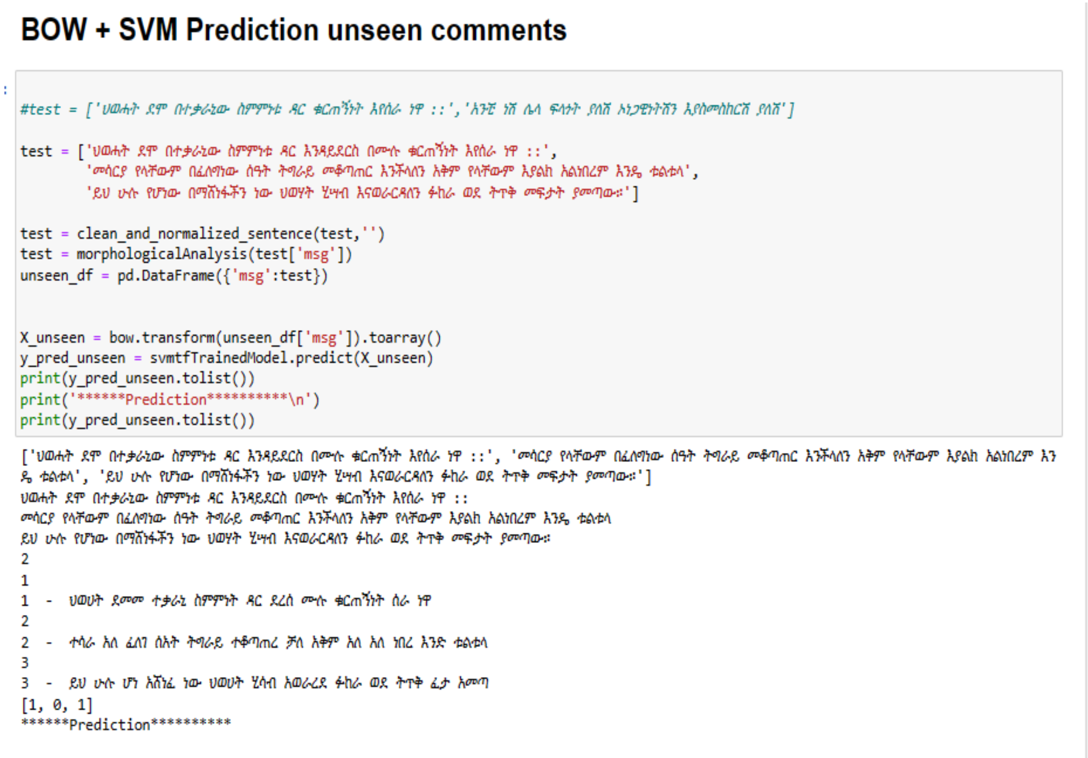


```
X_train, X_test, y_train, y_test= train_test_split(featurespro,
                                                 labels,
                                                 test_size=0.2,
                                                 random_state=0)
```

```
from sklearn.metrics import accuracy_score

# ------------- Build Model SVM using TF-IDF-------------#
svmModel = SVC(C=0.1, gamma=1, kernel='linear')

svmModel.fit(X_train, y_train)
y_pred = svmModel.predict(X_test)

print("ACCURACY OF THE MODEL: ", accuracy_score(y_test, y_pred))
print(classification_report(y_test,y_pred))
accuracy_score(y_test,y_pred)

## Export model
pickle.dump(svmModel, open('D:\\NLP\\model\\svm-model-tf-pro.pkl', 'wb'))
pickle.dump(featurespro, open('D:\\NLP\\model\\tfidf-pro.pkl', 'wb'))

## Importing Model
svmtfTrainedModel = pickle.load(open('D:\\NLP\\model\\svm-model-tf-pro.pkl', 'rb'))
featurespro = pickle.load(open('D:\\NLP\\model\\tfidf-pro.pkl', 'rb'))


confusion_matrix = metrics.confusion_matrix(y_test, y_pred)
print(confusion_matrix)
import matplotlib.pyplot as plt


cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])
cm_display.plot()
plt.show()

scores = cross_val_score( svmtfTrainedModel, X_train, y_train, cv=10, scoring='f1_macro')
print(scores)
```

A-4

Appendix B SVM classification

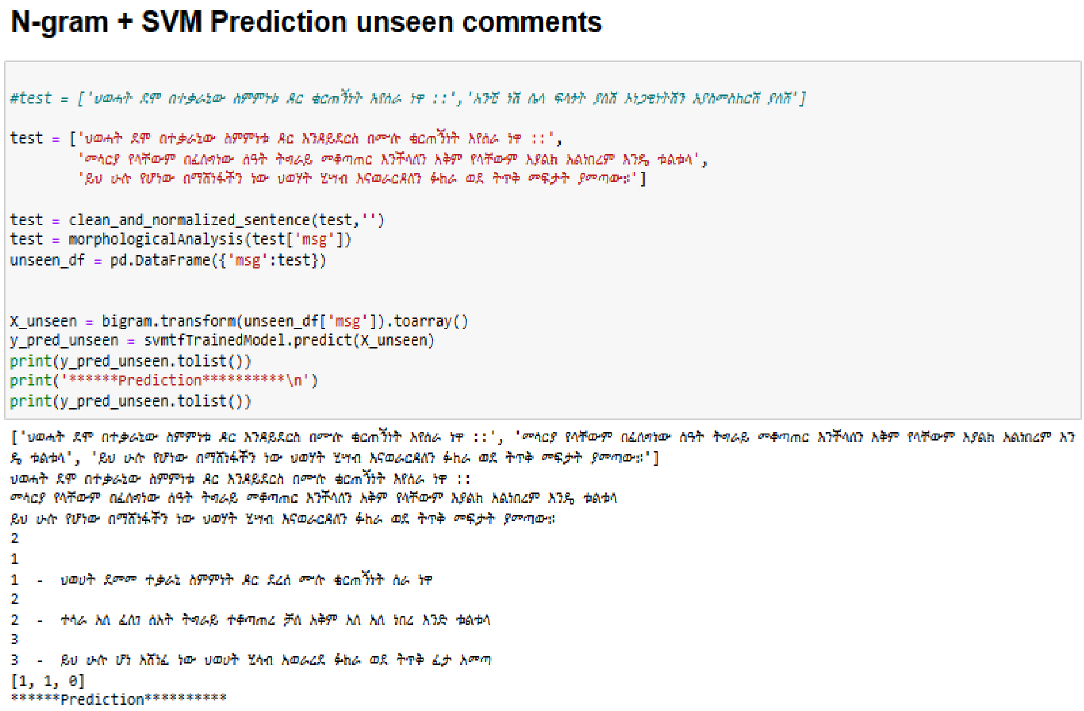


BOW + SVM Prediction unseen comments

```
#test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር ቁርጠኝነት እየሰራ ነዋ ::','አንቺ ነሽ ሌላ ፍላጎት ያለሽ ኦነጋዊትሽን እያስመሰከርሽ ያለሽ']

test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::',
        'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እንዴ ቁልቁላ',
        'ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።']

test = clean_and_normalized_sentence(test,'')
test = morphologicalAnalysis(test['msg'])
unseen_df = pd.DataFrame({'msg':test})


X_unseen = bow.transform(unseen_df['msg']).toarray()
y_pred_unseen = svmtfTrainedModel.predict(X_unseen)
print(y_pred_unseen.tolist())
print('******Prediction**********\n')
print(y_pred_unseen.tolist())
```

```
['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::', 'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እን
ዴ ቁልቁላ', 'ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።']
ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::
መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እንዴ ቁልቁላ
ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።
2
1
1  -  ህወሀት ደመመ ተቃራኒ ስምምነት ዳር ደረሰ ሙሉ ቁርጠኝነት ሰራ ነዋ
2
2  -  ተሳራ አለ ፈለገ ሰአት ትግራይ ተቆጣጠረ ቻለ አቅም አለ አለ ነበረ እንድ ቁልቁላ
3
3  -  ይህ ሁሉ ሆነ አሸነፈ ነው ህወሀት ሂሳብ አወራረደ ፋከራ ወደ ትጥቅ ፈታ አመጣ
[1, 0, 1]
******Prediction**********
```

B-1



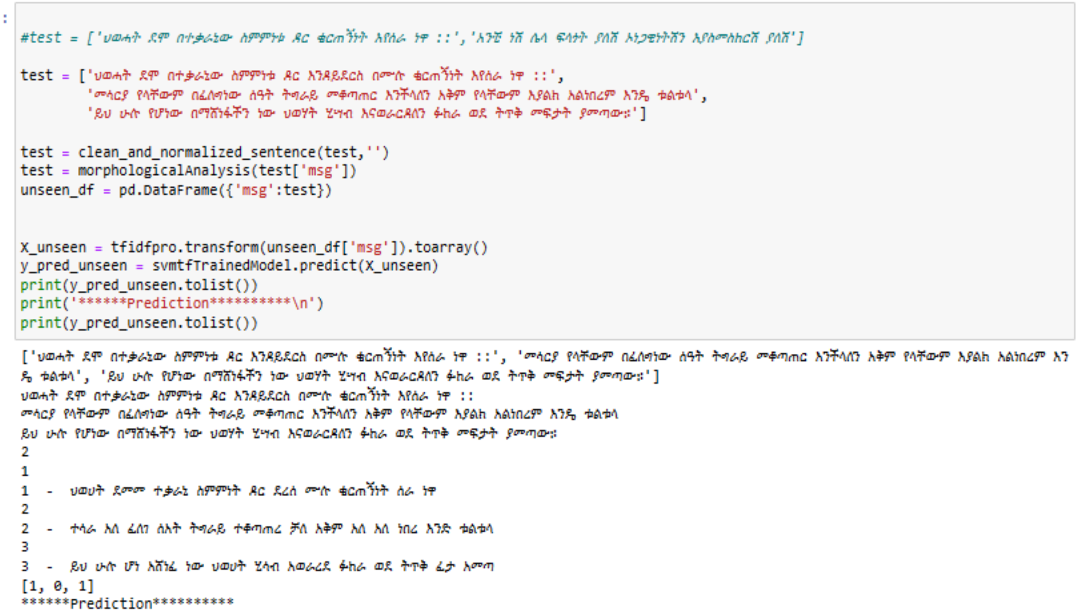
N-gram + SVM Prediction unseen comments

```
#test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር ቁርጠኝነት እየሰራ ነዋ ::','አንቺ ነሽ ሌላ ፍላጎት ያለሽ ኦነጋዊትሽን እያስመሰከርሽ ያለሽ']

test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::',
        'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እንዴ ቁልቁላ',
        'ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።']

test = clean_and_normalized_sentence(test,'')
test = morphologicalAnalysis(test['msg'])
unseen_df = pd.DataFrame({'msg':test})


X_unseen = bigram.transform(unseen_df['msg']).toarray()
y_pred_unseen = svmtfTrainedModel.predict(X_unseen)
print(y_pred_unseen.tolist())
print('******Prediction**********\n')
print(y_pred_unseen.tolist())
```

```
['ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::', 'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እን
ዴ ቁልቁላ', 'ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።']
ህወሓት ደም በተቃራኒው ስምምነቱ ዳር እንዳይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::
መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም እያልክ አልነበረም እንዴ ቁልቁላ
ይህ ሁሉ የሆነው በማሸነፋችን ነው ህወሃት ሂሣብ እናወራርዳለን ፋከራ ወደ ትጥቅ መፍታት ያመጣው።
2
1
1  -  ህወሀት ደመመ ተቃራኒ ስምምነት ዳር ደረሰ ሙሉ ቁርጠኝነት ሰራ ነዋ
2
2  -  ተሳራ አለ ፈለገ ሰአት ትግራይ ተቆጣጠረ ቻለ አቅም አለ አለ ነበረ እንድ ቁልቁላ
3
3  -  ይህ ሁሉ ሆነ አሸነፈ ነው ህወሀት ሂሳብ አወራረደ ፋከራ ወደ ትጥቅ ፈታ አመጣ
[1, 1, 0]
******Prediction**********
```

B-2

TF-IDF + SVM Prediction unseen comments

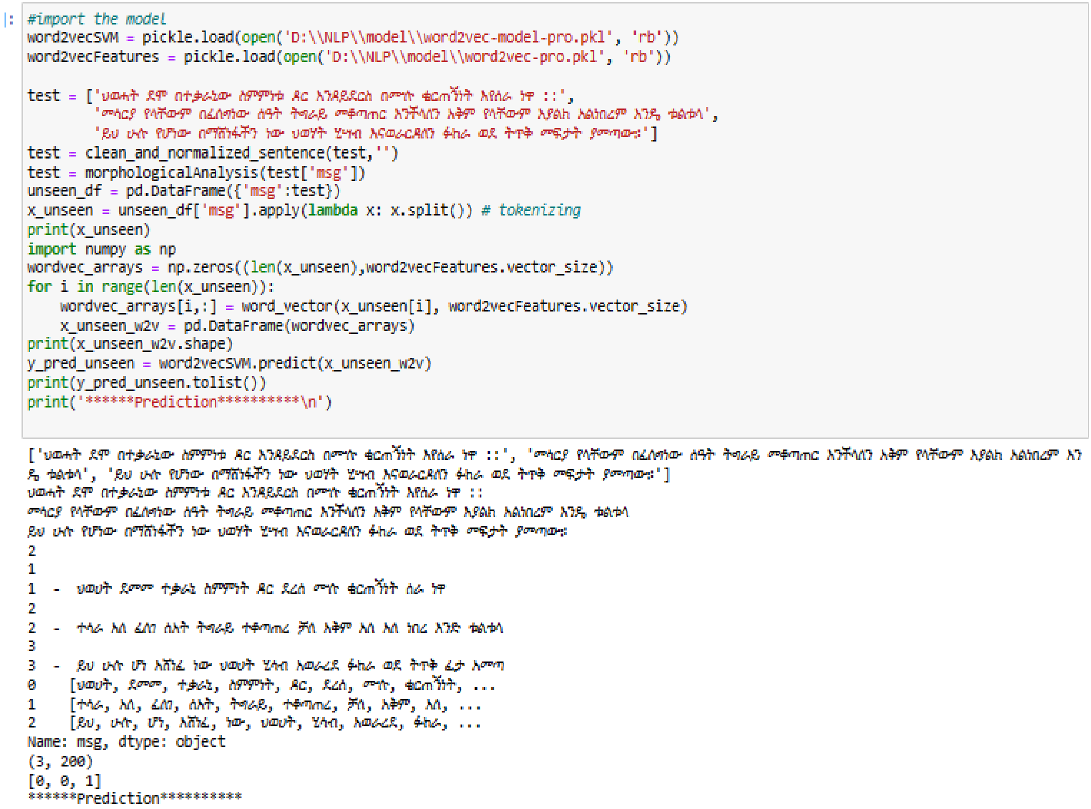


```
#test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ጸር ቁርጠኝነት እየሰራ ነዋ ::','አንቺ ነሽ ሌላ ፍላጎት ያለሽ እነጋቺትሽን አያስመስክርሽ ያለሽ']

test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::',
        'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እንዴ ቱልቱላ',
        'ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።']

test = clean_and_normalized_sentence(test,'')
test = morphologicalAnalysis(test['msg'])
unseen_df = pd.DataFrame({'msg':test})


X_unseen = tfidfpro.transform(unseen_df['msg']).toarray()
y_pred_unseen = svmtfTrainedModel.predict(X_unseen)
print(y_pred_unseen.tolist())
print('******Prediction**********\n')
print(y_pred_unseen.tolist())
```

```
['ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::', 'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እን
ዴ ቱልቱላ', 'ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።']
ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::
መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እንዴ ቱልቱላ
ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።
2
1
1  -  ህወሀት ደመመ ተቃራኒ ስምምነት ጸር ደረሰ ሙሉ ቁርጠኝነት ሰራ ነዋ
2
2  -  ተሳራ አለ ፈለገ ሰአት ትግራይ ተቆጣጠረ ቻለ አቅም አለ አለ ነበረ እንድ ቱልቱላ
3
3  -  ይህ ሁሉ ሆን አሸነፈ ነው ህወሀት ሂሳብ አወራረደ ፉከራ ወደ ትጥቅ ፈታ አመጣ
[1, 0, 1]
******Prediction**********
```

B-4

word2vec + SVM Prediction unseen comments



```
#import the model
word2vecSVM = pickle.load(open('D:\\NLP\\model\\word2vec-model-pro.pkl', 'rb'))
word2vecFeatures = pickle.load(open('D:\\NLP\\model\\word2vec-pro.pkl', 'rb'))

test = ['ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::',
        'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እንዴ ቱልቱላ',
        'ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።']
test = clean_and_normalized_sentence(test,'')
test = morphologicalAnalysis(test['msg'])
unseen_df = pd.DataFrame({'msg':test})
x_unseen = unseen_df['msg'].apply(lambda x: x.split()) # tokenizing
print(x_unseen)
import numpy as np
wordvec_arrays = np.zeros((len(x_unseen),word2vecFeatures.vector_size))
for i in range(len(x_unseen)):
    wordvec_arrays[i,:] = word_vector(x_unseen[i], word2vecFeatures.vector_size)
    x_unseen_w2v = pd.DataFrame(wordvec_arrays)
print(x_unseen_w2v.shape)
y_pred_unseen = word2vecSVM.predict(x_unseen_w2v)
print(y_pred_unseen.tolist())
print('******Prediction**********\n')
```

```
['ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::', 'መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እን
ዴ ቱልቱላ', 'ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።']
ህወሓት ደም በተቃራኒው ስምምነቱ ጸር እንጸይደርስ በሙሉ ቁርጠኝነት እየሰራ ነዋ ::
መሳርያ የላቸውም በፈለግነው ሰዓት ትግራይ መቆጣጠር እንችላለን አቅም የላቸውም አያልክ አልነበረም እንዴ ቱልቱላ
ይህ ሁሉ የሆነው በማሽነፋችን ነው ህወሃት ሂሣብ እናወራርጸለን ፉከራ ወደ ትጥቅ መፍታት ያመጣው።
2
1
1  -  ህወሀት ደመመ ተቃራኒ ስምምነት ጸር ደረሰ ሙሉ ቁርጠኝነት ሰራ ነዋ
2
2  -  ተሳራ አለ ፈለገ ሰአት ትግራይ ተቆጣጠረ ቻለ አቅም አለ አለ ነበረ እንድ ቱልቱላ
3
3  -  ይህ ሁሉ ሆን አሸነፈ ነው ህወሀት ሂሳብ አወራረደ ፉከራ ወደ ትጥቅ ፈታ አመጣ
0    [ህወሀት, ደመመ, ተቃራኒ, ስምምነት, ጸር, ደረሰ, ሙሉ, ቁርጠኝነት, ...
1    [ተሳራ, አለ, ፈለገ, ሰአት, ትግራይ, ተቆጣጠረ, ቻለ, አቅም, አለ, ...
2    [ይህ, ሁሉ, ሆን, አሸነፈ, ነው, ህወሀት, ሂሳብ, አወራረደ, ፉከራ, ...
Name: msg, dtype: object
(3, 200)
[0, 0, 1]
******Prediction**********
```

B-5