JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

Online Hate Speech Detection for Afaan Oromo Using Deep Learning

BY: -

OLIYAD SEBOKA MESKELE

A THESIS SUBMITTED TO FACULTY OF COMPUTING AND INFORMATICS OF JIMMA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY

JIMMA UNIVERSITY

JULY 12, 2021

APPROVAL SHEET

This independent research entitled as ―Online Hate Speech Detection for Afaan Oromo using Deep Learning ―has been read and approved as meeting the preliminary research requirement of School of Computing and Informatics in partial fulfillment for the award of the degree of masters in Information Technology.

Advisor	Signature	Date
Dr. Getachew Mamo	______________	_________
Co-Advisor	Signature	Date
Mr. Tesfu Mokonen	______________	_________
Internal Examiner	Signature	Date
Mr. Teferi Kebebewu	______________	_________
External Examiner	Signature	Date
Dr. Teklu Urgessa	______________	12/07/2021

DECLARATION

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

Oliyad Seboka Meskele

July 2021

DEDICATION

This paper is dedicated to my father Seboka Meskele Banti, my Mother Xajitu Reba Eyi, my Wife Tigist Shume, all my brothers and sisters who were able to reap the fruit of their own. I love you all.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God who helped me to succeed in all my life. Next, I would like to thank my Advisor Getachew Mamo (PhD) for his valuable assistance in providing his genuine, professional advice and encouragement that goes even beyond the accomplishment of this study. He initiated me to do by giving valuable comments on necessary points, especially on general knowledge of the research.

And also, I would like to acknowledge my co-advisor Tesfu Mekonen (MSc) for all his guidance at every step of the thesis work.

Besides, I would like to thank Mr. Kena Chala (MA in law), Magartu Seboka(MA in Afaan Oromo) and Mr. Alemu Kena(MA in Afaan Oromo) for helping on data labeling.

Lastly, I would like to thank Mettu University for giving me the opportunity for master degree study and also, I would like to thank Jimma University for teaching and providing for the research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BLSTM………………………………………………. Bidirectional Long Short-Term Memory

BoWV………………………………………………   Bag of Words vectors

CNN…………………………………………………. Convolutional Neural Network

GloVe………………………………………………   Global Vectors for Word Representation

GRU………………………………………………… Gated Recurrent Unit

LSTM……………………………………………… . Long Short-Term Memory

TF-IDF………………………………………………Term Frequency- Inverse Document Frequency

RNN………………………………………………... Recurrent Neural Network

SVM ……………………………………………… Support Vector Machine

ABSTRACT

Social networking has now days become a part of human life. People share their information, feelings, and emotions by using social sites like Facebook and Twitter. As social networking increasing day by day, cyber hate using these social sites are also increasing rapidly. Social media especially twitter and Facebook have a very big impact on the success or destruction of a person's image. Many of the social movements are done in social Medias, particularly Facebook and Twitter, all of which successfully affect the users. There is a well-targeted movement there is also a movement with the goal of evil that is spreading hatred to others. Hate speech can contain any form of appearance such as images, videos, songs as well as text. Detecting hate speech is the most important things to avoid the influence of hate speech on social media. Hate speech detection system will help to clean any hatred comment or post that creates the society to participate in the violent activities, and besides, it creates social media users to communicate without harm. In this research we presented hate speech detection for Afaan Oromo language to tackle hate speech on social media. To accomplish this research, we prepared a dataset of 14,077 label data to train and test our model. The collected dataset was labeled into three class's strong Hate, weak hate and neutral class. We trained three different deep learning models those are convolutional neural network, bidirectional long short-term memory neural network and the hybrid of the two neural networks models. We used the same dataset for each deep learning model. Additionally, word embedding was created by applying the word2vec algorithm with a CBOW model on a corpus collected from different social media. We explore the effect of using the pre-word embedding's with these models. Experimental results have shown that the use of word embedding's with neural networks effectively produces performance improvements in terms of run time and accuracy. The results achieved by CNN, BLSTM and CNN-BLSTM methods are 98.15%, 97.91% and 97.98% accuracy respectively. This research indicated that CNN model is more applicable to Afaan Oromo hate speech detection than CNN-BLSTM and BLSTM.

Keywords: hate speech, CNN, BLSTM, Deep learning, natural language processing, classification, neural network

CHAPTER ONE

INTRODUCTION

1.1. Background

"Natural language processing (NLP) is a branch of artificial intelligence (AI) that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. Natural language processing helps computers communicate with humans in their own language and scales other language-related tasks" [1]. For example, Natural language processing makes it conceivable for computers to read text, hear speech, understand it, measure sentiment and determine which parts are significant.

Social networking has now days become a part of human life. People share their information, feelings, and emotions by using social sites like Facebook and Twitter. As social networking increasing day by day, cyber hate using these social sites are also increasing rapidly. Social media especially twitter and Facebook have a very big impact on the success or destruction of a person's image. Many of the social movements are done in social networks, particularly Facebook and Twitter, all of which successfully affect the users. There is a well-targeted movement there is also a movement with the goal of evil that is spreading hatred to others [6]. "Hate speech covers many forms of expressions which spread, incite, promote or justify hatred, violence and discrimination against a person or group of persons for a variety of reasons" [54]. Hate speech can contain any form of appearance such as images, videos, songs as well as text.

Almost all countries throughout the World have laws that regulate Hate Speech. Hate speech is set in Article 55(1) of the Constitution of the Federal Democratic Republic of Ethiopia, it is proclaimed as follows: "speech that promotes hatred, discrimination or attack against a person or an identifiable group, based on ethnicity, religion, race, gender or disability" [2]. According to this proclamation, disseminating hate speech by means of broadcasting, the print or social media using text, image, audio or video is prohibited. We summarize the definitions of hate speech from different sources. According to the Constitution of united states of America: "Hate speech is speech that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity" [63]. Hate speech is defined in European Union law as the "public incitement to violence or hatred on the basis of certain characteristics, including race, color, religion, descent and national or ethnic origin" [64]. Different social media platforms like Facebook, Twitter, and YouTube have used different policies to handle hate speech. According to YouTube community guidelines, "free speech is encouraged while hate speech is not permitted. The content that encourages violence, hatred against individuals or groups depending on race, religion, disability, gender, age, veteran status, sexual orientation is defined as hate speech" [3]. Manual flagging and filing abuse reports are the options currently given by YouTube to report about hate content in YouTube. Twitter strives to provide an environment where people can feel free to express themselves. If hate behavior happens, Twitter makes it easy for people to report it. Multiple Tweets or Lists can be included in the same report, helping it to gain better context, while investigating the issues to get them resolved faster [4]. Facebook too has enumerated their policies for users about the content the user posts. In case of any policy abuse sending messages to the responsible person of posting, un-friend persons and block persons are few options given by Facebook to control promoting hate content through Facebook [5]. Not only these three-social media, but almost all the social media have come up with their own definitions, policies for hate speech. It signifies the importance of detection and removal of online hate content. The method and measurement used by some social media is interesting but it is not able to address the disseminations of hate speech completely.

In this research, we are proposing hate speech detection for Afaan Oromo language. Today there is a lot of Hate speech detection system developed for various languages including English, Arabic, Italian, German, Amharic, Vietnamese and other language. Though for the local languages or under resourced languages like Afaan Oromo language are not sufficiently implemented hate Speech detection. Text tweet are considered in this research since; the task of hate speech detection is considered as a text analytics task using natural language processing techniques. The dataset is collected targeting hate speech on Facebook and Twitter. The data was collected based on ethnicity, religion, race, gender and disability. Mainly, the dataset contains three classes; the strong hate, weak hate and neutral class. A Tweet is identified as Strong hate speech if it targets individuals or groups on the basis of their characteristics (based on ethnicity, religion, race, gender or disability). text is identified as Weak hate speech if it targets individuals or groups without contingent on their characteristics (without focusing on ethnicity, religion, race, gender or disability) or simply causing someone to feel resentful, upset, or annoyed. Neutral label is normal comments or posts on social media; it does not contain Strong hate speech or Weak hate speech. In this task, we focus on a solution for predicting hate speech on Afaan Oromo which is a low-resource language for natural language processing. In particular, we will implement deep learning to classify comments or posts on Facebook and twitter. The problem is stated as:

- Input: Given an Afaan Oromo Text on social media.
- Output: One of three labels (Strong hate, weak Hate and neutral) which is predicted by our system.

The word embedding technique is used for extracting a set of words features that can capture the hidden relations of words of the dataset. Word embedding was created by applying the word2vec algorithm with a CBOW model on a corpus collected from different social media and Wikipedia. We explore the effect of using the pre-word embedding's with these models. Keras deep learning framework is used for the implementation of the deep learning model. The proposed deep learning models are a Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BLSTM) and CNN-BLSTM (hybrid of CNN and BLSTM).

Tweet (Post or comment)	**Label**
Oromoon yoomu hin qoqqoodamu ***(oromo cannot be divide)***	Neutral
Gummuuzi fokkataa akka kee irra waan gaariin hin eegamu woyyaanen kufte Tigraay situ bulchaa jiraare bada goorillaa *(**good things can't be expected from a person like you ugly gumuz, woyane is failed are you leading Tigray now my chimpanzee**)*	Strong Hate
Hhh obbooloota akkookee warra gooftaakeef hojjetan suura kaaste hin maxxansiinkaa. Shaneen dhudhuuftuu akkakee miti. ***(hhh you are posting those who are working for your lord by taking your grandfather brothers photo. Shane is not farting like you do)***	Weak Hate

Table 1 : Hate speech category

1.2. Statement of the problem

Recently, Social media has become a source of harm in the form of misuse of private user details, hate speech and false information, specifically Facebook and Twitter. Social media can act as a transmission tool between online hate speech and real life violent crime. Hate speech is serious and growing problems in Ethiopia. It has contributed to the growing ethnic tensions and conflicts across the country that have created millions of internally displaced people in the first half of 2018 according to human right watch [7]. In the past year, speeches by government officials, activists and social media users in Ethiopia have disseminated quickly through social network and facilitated trigger or fuel violent conflicts in the country [7]. The most important step in this process needs detecting and tracking hate speech online.

Today there are a lot of Hate speech detection system developed for various languages including English, Arabic, Italian, German, Amharic, Vietnamese and other language. Why not for Afaan Oromo language? Because there is no generalized work since hate speech is language dependent. While every language has a different set of rules, all language obeys rules. All language has underlying structure rules that make meaning full communication possible. Though for the local languages or under resourced languages like Afaan Oromo language are not sufficiently implemented hate Speech detection.

Afaan Oromo is an official language of Oromia Regional State (which is the largest regional State among the current Federal States in Ethiopia) [8]. It is one of the major languages that are widely used as a working language of the Regional State of Oromia [9]. Afaan Oromo has a large number of native speakers when compared with the rest of Ethiopian languages [10]. As the consequence of the number of Afaan Oromo speakers in Ethiopia and Africa there are a large number of social media users, activist and groups. In Afaan Oromo language, like other languages, there exists hate speech propagation based on the ethnic, religion, race, demographic areas and political background between different groups. The dissemination of hate speech by using Afaan Oromo text through a social media, are growing from time to time and there is no enough research conducted previously, and this research is the first research to detect hate speech for Afaan Oromo text on social media.

Based on the above research problem, the following research questions will be raised:

1. How and when the performance of the proposed hate speech detection system increases?
2. From CNN, BLSTM and CNN-BLSTM which model is a better for Afaan Oromo hate speech detection?

1.3. Objectives

1.3.1. General objective

The general objective of this research is to develop online hate speech detection for Afaan Oromo Deep Learning.

1.3.2. Specific objectives

The following specific objectives are identified in order to achieve the specified general objective: -

- Review literatures on the concepts of hate speech detection and assess different techniques and approaches used so far hate speech Detection.
- To investigate and design a frame work of Afan Oromo hate speech detection model.
- To collect and compile necessary dataset for training and testing purpose.
- To develop Afan Oromo hate speech detection models.
- To test the models and report their results.

1.4. Scope and limitation

Here, we have described the main emphasis that our research study focused on as follows;

- The study is confined to detect hate speech Text on Facebook and twitter and it does not include video, image and audio data files because the majority of the hate speech is mainly propagated by using text.
- It is predominantly concentrate only on detecting Afaan Oromo hate speech and it is not multilingual.
- It did not include Afaan Oromo tagger, spelling checker, morphology analyzer, parser and etc.

1.5. Methodology

Under this section, different methods of conducting review, data collection and preparation, design framework, implementation tools and evaluation are discussed as follows.

1.5.1. Review of related literature

Deferent Literature reviews are conducted by concerning on different areas relevant to hate speech detection in a way that identifies the existing gaps. Thus, current computational methods of hate speech detection are reviewed from existing literatures and related works. Accordingly, for the identified gaps working approaches are identified and the one appropriate for the identified problems is selected.

1.5.2. Data collection and data preprocessing

Thousands of Afaan Oromo posts and comments on suspected social media pages of organizations and individual person's public pages are crawled as dataset for the evaluation of the proposed system. The collected data from the social media is manually annotated with the help of legal and linguistic professional to train and test the model.

1.5.3. Development Requirement

We used the keras 2.2.4 deep learning python library to develop the prototype. This library is used with python 3.7.x and it is uses tensor flow as its backend. Keras is a high level deep learning API which is very valuable in fast prototyping and ignores the detail implementations of back propagation (writing optimization procedures). This library is an open source and has large community support. Also, this library supports both types of neural networks (i.e. convolutional and recurrent).

1.5.4. Detection method

In this research work, three different deep learning models were used and those are CNN, BLSTM and the hybrid of CNN-BLSTM. A deep learning is a series of algorithms that endeavors to distinguish underlying relationships in a set of data through a process that mimics the way the human brain works. In this logic, neural networks refer to systems of neurons, either organic or artificial in nature [52]. NN can adapt to changing input; so, the network produces the best possible result without requiring redesigning the output criteria that is why we are selecting deep learning for hate speech detection. The concept of neural networks, which has its roots in artificial intelligence, is rapidly gaining popularity in the development of Hate Speech Detection.

1.5.5. Evaluation

The collection of result from the system, observing the result, statistically analyzing them to evaluate the system performance is necessary. Analyzing the proposed system with performance metrics is used to identify the contribution achieved in the study. We have evaluated our prototype automatically using keras accuracy metrics, on sample data.

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

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{F1 Score} = \frac{2*(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

Where,

- ✓ TN - True Negative: when a case was negative and predicted negative
- ✓ TP - True Positive: when a case was positive and predicted positive
- ✓ FN - False Negative: when a case was positive but predicted negative
- ✓ FP–False Positive: when a case was negative but predicted positive

1.6. Application of results

Detecting hate speech is the most important things to avoid the influence of hate speech on social media. Hate speech detection system will help to clean any hatred comment or post that creates the society to participate in the violent activities, and besides, it creates social media users to communicate without harm. Furthermore, detecting and punishing harmful behaviors on the internet are also necessary activities of every organization and authorities to protect the safety of the country and regions.

1.7. Structure of the Thesis

The rest of this thesis is organized as follows. In chapter 2, literature review and related work are described. The chapter explains a different type of approaches to Hate Speech Detection specifically it explains convolutional neural Network and Recurrent Neural Network based approaches. Related works particularly which is relevant to our work was discussed. Different types of Hate Speech Detection boundary identifications were also discussed in this chapter. In Chapter 3, Basic Afaan Oromo language structure was discussed. Here we presented about the word class, phrases, punctuation and sentences of Afaan Oromo. Chapter 4 presented Afaan Oromo Hate speech detection model using CNN, BLSTM and CNN-BLSTM. Chapter 5 presented the experimental results of the proposed model along with its discussion. Finally, Chapter 6 concluded the thesis with the research findings and future works.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

This chapter presents the state of the art in the hate speech detection System with overview of their components and the techniques developed. This part of study enlightens briefly on some of the basic concepts of the hate speech detection and the techniques applicable with the system.

2.2. Hate Speech

The phenomenon that we nowadays refer to as "hate speech" is an ancient one. On numerous occasions in history, hate speech has been used to target individuals or groups with the purpose of stigmatization and to incite hatred and violence. Hate speech was used as an instrument in, for example, the German Nazi holocaust and in the genocide in Ruanda. Public concerns about hate speech grew specifically after the events of the World II War, and in the light of such events, several international agreements were signed, most importantly the Universal Declaration of Human Rights (UDHR) (1948), and the International Covenant on Civil and Political Rights (ICCPR) (1966).

These agreements now protect the rights of each individual to, for example, equality, personal dignity, security, and freedom of opinion and expression (UDHR, articles 1–3, 19), and ban all forms of discrimination in violation of those rights (UDHR, article 7). More explicitly, the agreements prohibit engaging in propaganda for purposes of war or pleasing to national, racial, or religious prejudices to incite hostility, discrimination, or violence (ICCPR, article 20). Other agreements legislate in contradiction of racial discrimination, genocide, and other violations of international law (Convention on the Prevention Punishment of the Crime of Genocide, 1948; International Convention on the Eradication of All Forms of Racial Discrimination (ICERD), 1965). The most severe forms of "hate speech" can be well-defined and recognized based on these international agreements.

2.3. Automatic approaches for hate speech detection

Most social media platforms have established user rules that prohibit hate speech; enforcing these rules, however, requires copious manual labor to review every report. Some platforms, such as Facebook, recently increased the number of content moderators. Automatic tools and approaches could accelerate the reviewing process or allocate the human resource to the posts that require close human examination. In this section, we overview automatic approaches for hate speech detection from text.

2.3.1. Keyword-based approaches

A basic method for detecting hate speech is using a keyword-based approach. By using an ontology or dictionary, texts that have potentially hateful keywords are identified. For example, Hatebase [32] maintains a database of derogatory terms for many groups across 95 languages. Such well-retained resources are valued, as terminology altered over time. However, as we observed in our study of the definitions of hate speech, simply using a hateful smear is not necessarily enough to organize hate speech.

These approaches are fast and straightforward to comprehend. But, they have numerous insufficiencies. Detecting only racial smears would consequence in a highly precise system but with low recall where precision is the percentage of relevant from the set detected and recall is the percent of relevant from within the global population. In other words, a system that depends largely on keywords would not identify hateful content that does not use these terms.

2.3.2. Machine learning approach

By using pre-labeled examples as training data, a machine learning algorithm can learn the different associations between pieces of text and that a particular output (i.e. tags) is expected for a particular input (i.e. text) [12]. The first step towards training a classifier with machine learning is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words. For example, if we have defined our dictionary to have the following words {*Hin ta'uu gambeella hundii isaani suudanitti deebi'uu qabu* }, and we wanted to vectorize the text "*suudanitti deebi'uu qabu*", we would have the following vector representation of that text: (0, 0, 0, 0, 0, 1, 1,1).Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. strong Hate, weak hate and neutral) to produce a classification model: Once it's trained with enough training samples, the machine learning model can begin to make accurate predictions. The same feature extractor is used to transform unseen text to feature sets which can be fed into the classification model to get predictions on tags (e.g. strong Hate, weak hate and neutral): hate speech detection with machine learning is usually much more accurate than human-crafted rule systems, especially on complex classification tasks. Some of the most popular classical machine learning algorithms for creating text classification models include the Support Vector Machine, Naïve Bayes Classifier, Logistic Regression Classifier, Decision tree Classifier and K-Means Clustering algorithm.

2.3.2.1. Supervised machine learning algorithms

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labeled training data, that is, a dataset that already holds a known value for the target variable for each record. Because you provide the machine learning with the correct answers for each record during training, the algorithm is able to "learn" how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data of hate speech.

However, successfully building, scaling, and deploying accurate supervised machine learning models for hate speech detection has historically required extensive time and technical expertise from a team of highly skilled, expensive data scientists. Moreover, data science groups must periodically rebuild hate speech detection models in order to make sure the insights they provide remain true to life as the input data changes [33]. Types of Supervised Machine Learning Algorithms: -

Regression

Regression technique predicts a single output value by using training dataset. Outputs always have a probabilistic clarification, and the algorithm can be regularized to evade over fitting. Logistic regression might underperform when there are multiple or non-linear decision boundaries. Regression technique is not flexible, so it does not capture more complex relationships.

Logistic Regression

Logistic regression technique used to estimate discrete values based on given a set of independent variables. It helps us to predict the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logistic regression. As it predicts the probability, its output value lies among 0 and 1.

Classification

Classification algorithm means to group the output inside a class. If the algorithm tries to label input into two different classes, it is called binary classification. Choosing between more than two classes is referred to as multiclass classification.

Naïve Bayes Classifiers

Naïve Bayesian model is simple to build and very useful for large data. The theoretical background of Naïve Bayes classifier assumes independent of features. For text classification scenario we will use the tokens of the text as a feature to classify it to the appropriate class. By using the maximum posteriori (MAP) decision rule, we come with the following classifier. The following formulas are according to the work done in [37].

$$Cmap = \arg\max \ \left(\mathrm{p}(c|d)\right) = \arg\max(p(c) \prod p(tk|c)\ ))$$

Where t_k is words of the text, C is the set of classes that is used in the classification, p(c/d) the conditional probability of class c given text d, p(c) the prior probability of class C and p(t_k /c) the conditional probability of token t_k given class C. It implies in order to find the target class in which the new instance is going to be labeled first we have to estimate the product of the probability of each word among the instance given that a particular class or probably likelihood of it multiplied by the probability of the particular class. The immediate task after calculating each of those class values is selecting the class with highest probabilities. As a solution to get rid of float point underflow of specific decimal point accuracy in place of maximizing the product of the probabilities it's better to maximize the sum of their logarithms:

$$Cmap = \arg max \left\{ \log\ p(c) + \sum_{1<K<n} \log\ p\ (tk|c) \right\}$$

After all, in place of taking the class with the greatest probability values take the class which is with the greatest log mark. The decision of MAP will remain the same in case of the algorithm function is monotonic. If a particular feature does not exist in a certain class, its conditional probabilities will be zero. The product becomes zero in case of using the first decision method, when we apply the second one the product log zero going to be undefined. To get rid of such headache we will use Laplace smoothing by adding one to each count: [36]

$$p(t|c) = \frac{(Tct + 1)}{\sum_{teV}(Tct + 1)} = \frac{(Tct + 1)}{\sum_{teV}(Tct) + \beta'}$$

Where β' is equal to the number of the words contained under the vocabulary V.

Decision Trees

Decisions trees classify instance by sorting them depending on the feature value. In this technique, each mode is the feature of an instance. It should be classified, and every branch represents a value which the node can hold. It is a widely used technique for classification. In this approach, classification is a tree which is known as a decision tree.

Support Vector Machine

SVM is a type of machine learning algorithm developed in 1990. This approach is based on results from statistical learning theory introduced by Vap Nik. Support Vector Machine machines are also closely linked to kernel functions which is a central concept for most of the learning tasks. The kernel framework and SVM are used in a variety of areas. SVM is an algorithm that determines the greatest decision boundary between vectors that belong to a given category and vectors that do not belong to it. That is, it. It can be applied to any kind of vectors which encode any kind of data. This means that in order to leverage the power of SVM hate speech classification, texts have to be transformed into vectors. Support Vector Machine machines determines the decision boundary we mentioned above, Support Vector Machine machines decides where to draw the best line that boundaries the space into two subspaces one for the vectors which belong to the given class and one for the vectors which do not belong to it [38]. So, provided we can find vector representations which encode as much information from our texts as possible, it is also possible to apply the SVM algorithm to hate speech detection problems.

Say, for example, the blue circles in the graph below are representations of training texts which represent about the strong hate class and the red triangles are representations of training texts which do not belongs to that (that means, weak hate class). What would the decision boundary for the hate speech category look like?

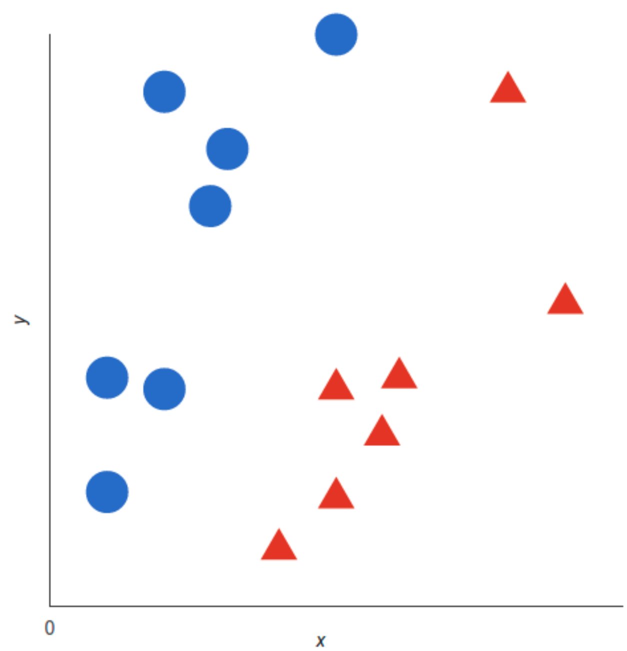


Figure 1: Support Vector Machine machines before decision boundary

The best decision boundary would look like this:

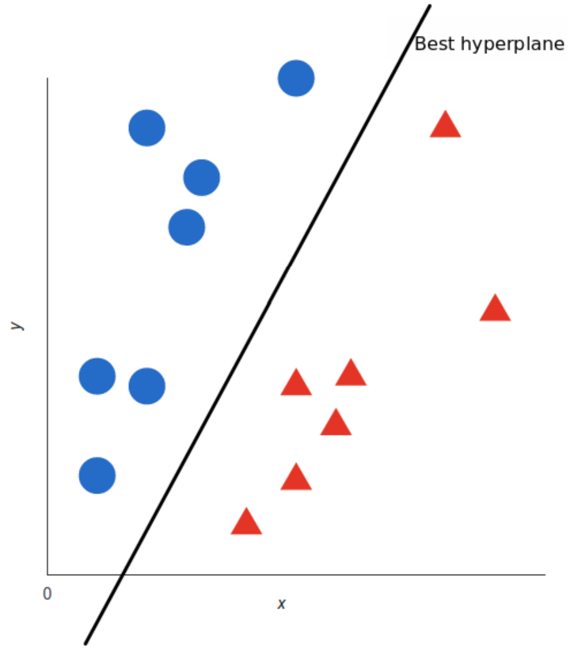


Figure 2: Support Vector Machine machines after decision boundary

Now that the SVM algorithm has determined the decision boundary for the category we want to analyze, we only have to take the representations of all of the texts we would like to classify and check what side of the boundary those representations are categorized into.

2.3.2.2. Unsupervised machine learning algorithms

Unsupervised learning is the training of machine using information that is neutral classified nor labeled and letting the algorithm to act on that information without supervision. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of dataset [34]. Here, are prime reasons for using Unsupervised Machine Learning: it finds all kind of unknown patterns in data, it helps to find features which can be useful for classification and It is easier to get unlabeled data from a source than labeled data, which require manual intervention. Unsupervised, machine learning problems further grouped into clustering and association problems.

Clustering

Clustering is a significant concept when it comes to unsupervised learning. It primarily deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process our dataset and find natural clusters if they exist in the dataset. We can also adapt how many clusters our algorithms should identify. It tolerates us to adjust the granularity of these groups. There are different types of clustering algorithm: - Hierarchical Clustering, K-means Clustering, K-NN (k nearest neighbors), Principal Component Analysis', Singular Value Decomposition and Independent Component Analysis.

Association

Association rules allow establishing associations between data objects inside large dataset. This unsupervised technique is about discovering interesting relationships between variables in large dataset.

2.3.2.3. Semi-supervised machine learning algorithms

Semi-supervised machine learning model resides between supervised learning and unsupervised. Semi-supervised learning accepts dataset that is partially labeled that means the majority of the data lacks labels. Semi-supervised learning determines the associations between the data points, just like unsupervised learning and then uses the labeled data to mark those data points. Finally, the whole model is trained based on the newly applied labels [35]. It is not good to use only the small labeled data to train the model because it is well known that when the proportion of the figure of training samples to the number of feature measurements is small, the training result is accuracy will be affected. Therefore, the model needs to combine labeled and unlabeled data during training to improve performance. The unlabeled data can be taken for density estimation or preprocessing of the labeled data, such as detecting inherent structure in the domain. In other words, the model extracts patterns from the annotated data, and labels the unannotated data automatically using the patterns. As a result, entry data are labeled for the training. It saves human effort while the performance can be as good as the performance of a supervised machine learning method.

2.3.3. Deep learning Approach

The exponential growth in the number of complex datasets every year requires more enhancements in machine learning methods to provide robust and accurate data classification. Lately, deep learning approaches are achieving better results compared to previous machine learning algorithms on tasks like image classification, natural language processing, face recognition, etc. The success of these deep learning algorithms relies on their capacity to model complex and non-linear relationships within the data [13].

Deep learning is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered features.

On the other hand, deep learning algorithms may require much more training data than traditional machine learning algorithms, i.e. at least millions of tagged examples. On the other hand, traditional machine learning algorithms such as SVM and NB reach a certain threshold where adding more training data doesn't improve their accuracy. In contrast, deep learning classifiers continue to get better the more data you feed them with.

2.3.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks are basically just numerous layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. In an old feed forward neural network we connect each input neuron to each output neuron in the next layer. That's also called a fully connected layer. In Convolutional Neural Network we don't do that. Instead, we use convolutions over the input layer to find the output. This results in local connections, where each region of the input is connected to a neuron in the output. Each layer performs different filters, typically hundreds or thousands. There's also something called pooling layers. The results of the convolution layer are "pooled" or aggregated to a representative number. This number (representative number) is nourished to a fully connected neural structure, which makes a classification decision depending on the weights assigned to each feature within the text [39].

In Natural Language processing, instead of image pixels, the input to these tasks are sentences or documents represented as a matrix. Each row of the matrix matches to one token, typically a word, but it could be a character. That is, each row of the matrix is vector that represents a word. Naturally, these vectors are word embedding's like word2vec or GloVe, but they could also be one-hot vectors that index the word into a vocabulary. For a 10-word Afaan Oromo sentence using a 100-dimensional embedding we would have a 10×100 matrix as our input [40].

The next figure illustrates how the convolutional "filter" slides over a sentence, two words at a time. It calculates an element-wise product of the weights of each word, multiplied by the weights assigned to the filter.

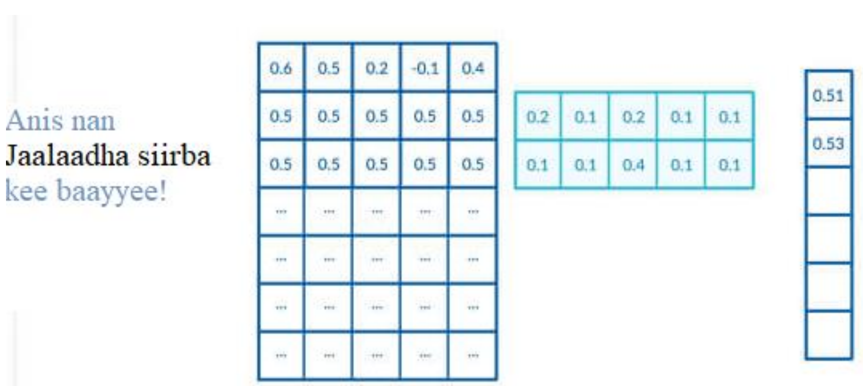


Figure 3: How the convolutional filter slides over a sentence

The sum of the products is used as a representation of the current textual feature 0.51 and 0.53 in the example. This is the “pooling” phase, decreasing the dimensionality of the word features and retaining only a simple probability score that imitates how likely they are to match a label.

At the last phase, these scores are the inputs to a fully connected neural layer. The “fully connected” part of the CNN network goes through its own back propagation process, to conclude the most accurate weights. Each neuron obtains weights that prioritize the most appropriate label for example, “strong Hate”, “weak hate” and “neutral”. Lastly, the neurons vote on all of the labels and the victor of that vote is the classification decision [39].

2.3.3.2. Bidirectional LSTM

Bidirectional LSTMs are an extension to typical LSTMs that can boost performance of the model on sequence classification task. Where all time steps of the input sequence are obtainable, BLSTMs train two LSTMs instead of one LSTM on the input sequence. The first on the inputs sequence as is and the other on a reversed copy of the input sequence. By this additional context is added to network and results are quicker.

The idea behind Bidirectional Recurrent Neural Networks is actually straightforward. Which involves replicating the first recurrent layer in the network then providing the input sequence as it is as input to the first layer and providing a reversed copy of the input sequence to the duplicated layer. This overwhelms the limitations of an old RNN. Bidirectional recurrent neural network can be trained using all existing input info in the past and future of a specific time step Split of state neurons in regular Recurrent Neural Networks is responsible for the forward states and a part for the backward states. See figure (4)

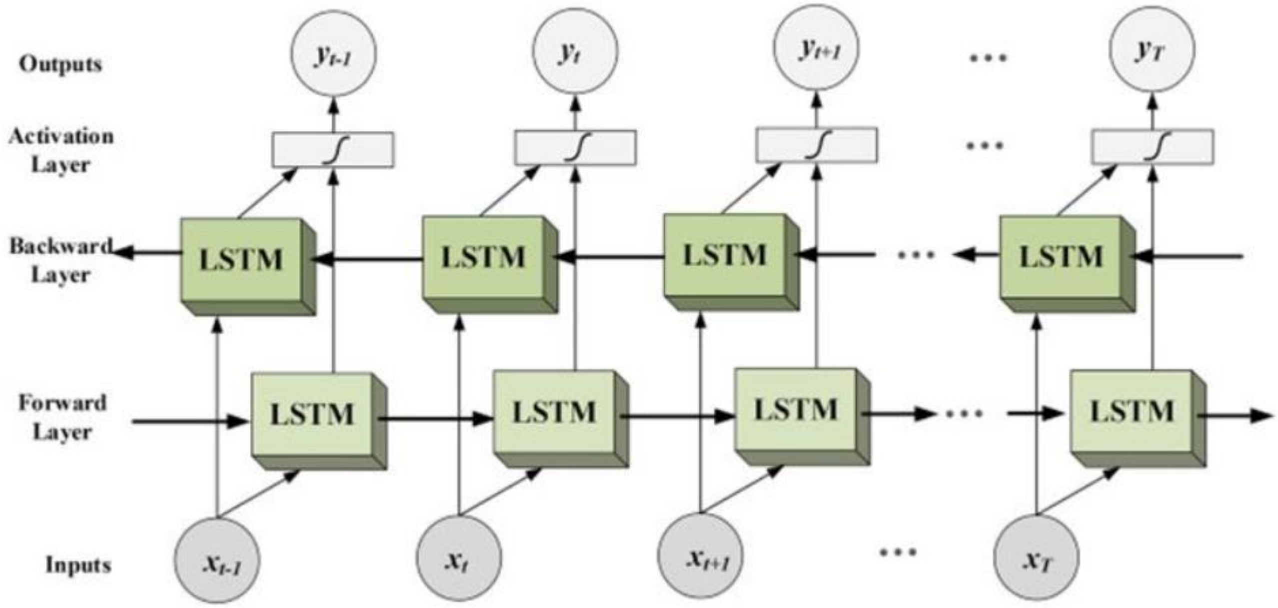


Figure 4: Bidirectional LSTM [41]

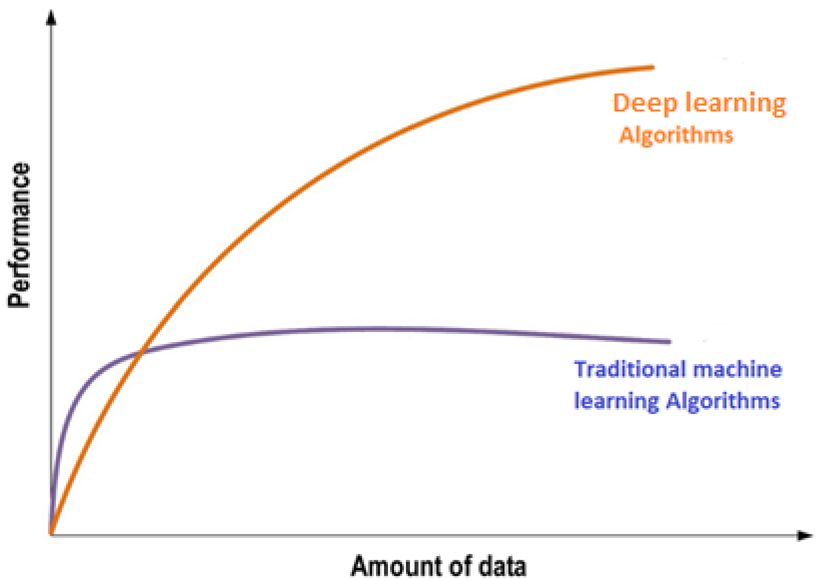
Bidirectional layer wrapper conveys the implementation of Bidirectional LSTMs in Keras. Bidirectional layer wrapper takes a recurrent layer (first LSTM layer) as an argument and we can also specify the merge mode, that describes how forward and backward outputs should be merged before being passed on to the next layer. The options are: Sum; the results are added together, mul; the results are multiplied together, Concat (the default); the results are concatenated together, providing double the number of outputs to the coming layer and ave; the average of the results is taken.

2.3.4. Comparison of machine learning and Deep learning

Strictly speaking, a deep learning is a type of machine learning model that is usually used in supervised learning [61]. By connecting together many different nodes, each one responsible for a simple computation, neural networks try to form a rough parallel to the way that neurons function in the human brain. Traditional Machine learning uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned. deep learning structures algorithms in layers to create an "artificial neural network" that can learn and make intelligent decisions on its own. This section presents some important points of deep learning and machine learning then compare the two methods.

Data Reliance

Deep learning algorithms may require much more training data than traditional machine learning algorithms, i.e. may be millions of tagged examples. On the other hand, traditional machine learning algorithms such as SVM and NB reach a certain threshold where adding more training data doesn't improve their accuracy. In contrast, deep learning classifiers continue to get better the more data you feed them with. This is because deep learning algorithms need a large amount of data to understand it perfectly or in order to make a brief conclusion. Due to the complex multi-layer structure, a deep learning system needs a large dataset to eliminate fluctuations and make high-quality interpretations, while traditional machine learning algorithms have a rather simple structure.



Hardware Reliance

On the subject of the hardware reliance, deep learning algorithms greatly depend on high-end machines, contrary to traditional machine learning algorithms, which can work on low-end machines. This is because the requirements of deep learning algorithm comprise GPUs which are an essential part of its working. deep learning algorithms fundamentally do a large amount of matrix multiplication tasks. These tasks can be efficiently optimized using a GPU because GPU is built for this drive [61].

Training Time

Concerning the training time, typically a deep learning algorithm takes a long time to train. This is for the reason that, there are so many parameters in a deep learning algorithm that training them takes longer than usual. Whereas machine learning comparatively takes much less time to train, ranging from a few seconds to a few hours, but this is absolutely inverted on testing time. At test time, deep learning algorithm doesn't take much time compare to machine learning to run [61].

Problem solving capability

Deep learning has the potential to solve the most challenging problem than machine learning. Relative to machine learning techniques, deep learning has four key advantages; Its ability to detect complex interactions among features, the ability to learn low-level features from minimally processed raw data, the ability to work with high-cardinality class memberships the ability to work with unlabeled data. Taken together, these four strengths mean that deep learning can produce useful results where other methods fail; it can build more accurate models than other methods; and it can reduce the time needed to build a useful model [62].

2.4. Related works

A number of researches have been conducted on the hate speech detection in different natural language such as Amharic, Arabic, Italian, English, Vietnamese and German. The researchers used different techniques as we have discussed in chapter two, to develop the hate speech detection for the above listed natural languages. We consider the conducted researches on local languages and on some international languages. Here, below we have discussed the different hate speech detection system of the above listed natural languages as follows.

2.4.1. Hate speech detection for different languages

2.4.1.1. Amharic language

Z. Mossie and J. Wang [11] studied social network hate speech Detection for Amharic language. The studies were focus only on hate speech detection from social media posts and comments. They aimed at classifying the hate level across Facebook for Amharic language users. They used Word2Vec and TF-IDF for feature selection and Naïve Bayes and Random forest machine learning algorithms for hate speech detection. They collected a dataset of 6,120 Amharic posts and comments out of this 4,882 to train the model and 1,238 for testing. According to the author, the model was tested to classify whether the post and comments are hate or not and able to detect and classify in an accuracy of 79.83 % and 65.34% for Naïve Bayes with word2vec feature vector and Random Forest with TF-IDF feature modeling approach respectively. Finally, they recommended technical improvements that can be made for the language in terms of: the number of dataset, analyzing the different aspect of the category of hate, utilize the information provided by Facebook so that, classification can be improved by expanding the feature space with profile information, list of followers and geo location and including other social media like Twitter, forums and other homepages beyond Facebook.

2.4.1.2. Vietnamese language

Q. Pham Huu, S. Nguyen Trung and H. Anh Pham [12] studied Automated Hate Speech Detection on Vietnamese Social Networks. They suggested a novel method for solving this problem by a multi-class classification model to classify content into three labels: HATE, OFFENSIVE, and CLEAN with the Vietnamese dataset. The dataset consists of 25431 items with three classes. The algorithms used are: Logistic Regression, Support-vector machine, Neural Network, Nearest Neighbors, Decision Tree, Naive Bayes and Random Forest. With experiment result of 0.6675 and 0.6197 F1-score on the test data and private test data respectively.

2.4.1.3. Arabic language

H. Faris, I. Aljarah, M. Habib and P. Castillo [13] they conducted Hate Speech Detection using Word Embedding and Deep Learning for Arabic Language. The proposed deep learning model was a recurrent convolutional network, which is a combination of convolutional network layer and LSTM network. They used word embedding technique for extracting a set of words features that can capture the hidden relations of words of the dataset. The utilized word embedding’s are the Word2Vec and the AraVec implementations. The proposed approach is evaluated based on different performance evaluation measures including the accuracy, precision, recall, and F1 measure. AraVec accomplishes superior results in terms of all metrics. They achieved accuracy of (66.564%), recall of (79.768%), precision of (68.965%), and F1 measure of (71.688%). Word2Vec achieved relatively close performance regarding the recall and F1 measure, but AraVec still superior.

2.4.1.4. Italian language

G. Bianchini, L. Ferri and T. Giorni [14], they presented a system based on neural networks for the hate speech detection in social media messages in Italian language. The system works in four phases: preprocessing of the initial dataset, encoding of the preprocessed dataset, training of the Machine Learning model, testing of the trained model. There are different types of ANN, the one used in this research is a feed-forward. They achieved Accuracy: 83.73% trained with the dataset of 3000.

2.4.1.5. German language

T. Scheffler, E. Haegert, S. Pornavalai and M. Lee Sasse [16] in this work, they presented a hate speech classifier for German tweets. The models used are Logistic Regression, Linear SVM and Adaboost classifiers using character n grams as well as additional textual features. During investigation, SVM performed consistently better than the other Two but not by much. They achieved a macro F1-score of 0.77 (95% confidence interval: ±0.04) in cross validation. The training data for this activity consisted of 5009 German tweets. The tweets were annotated in to two classifications

OFFENSE and OTHER. In future work, the researcher plans are to use a larger amount of data set may be helpful for training classification systems. Because, it would be particularly helpful for the second, fine-grained tasks, where the current classifiers showed really poor performance.

2.4.1.6. English language

According N.D.T. Ruwandika and A.R. Weerasinghe [15] they conducted on Identification of Hate Speech in Social Media. The research was concentrated on a lexicon-based approach with the combination of a machine learning technique. Supervised learning and unsupervised learning algorithms were used to build the classifier models. They tested on five different classification models and they were Support Vector Machine, Naïve Bayes Classifier, Logistic Regression Classifier, Decision tree Classifier and K-Means Clustering algorithm. Dataset contained of user written comments from different articles in Colombo Telegraph website. According to the investigation done by the researcher on results it was well-known that Naïve Bayes classifier with Tf-idf features performed best out of all models with an F-score value of 0.719. At end they recommended and noticed that supervised learning models perform better than unsupervised learning models as it is better to try out other unsupervised learning techniques for the task since K-Means clustering model performed little bit better with few feature types.

The other one is a research conducted by S. Malmasi and M. Zampieri [16] which is Detecting Hate Speech in Social Media. According to this research, the aim is to establish lexical baselines for hate speech detection by applying supervised classification. They addressed the problem of hate speech detection using a dataset annotated with three labels: hate speech (HATE), offensive language but no hate speech (OFFENSIVE), and no offensive content (OK). They used a linear Support Vector Machine (SVM) classifier and used three groups of features extracted for these experiments: Surface n-grams, word skip-grams, and Brown cluster. They obtained results of 78% accuracy in identifying posts across three classes. The future work plan of the researcher is to investigate the performance of classifier ensembles and Meta learning for hate speech detection.

In Z. Zhang, D. Robinson and J. Tepper [17] Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network. They introduce a new method based on a deep neural network combining convolutional and long short-term memory networks and conducts an extensive evaluation of the method against several baselines and state of the art on the largest collection of publicly available datasets.

They introduced a technique for automatically classifying hate speech using a deep neural network model combining CNN, LSTM with drop-out and pooling that are found to empirically improve classification accuracy. The result shows that the proposed method outperforms state of the art on 6 out of 7 datasets by between 0.2 and 13.8 points in F1. The future work of the researchers is to further focusing on neural network to consider, e.g., stacking multiple convolutional layers which are good for extracting hierarchical features, to integrate user-centric features, such as the frequency of a user detected for posting hate speech and the user's interaction with others and to study and quantify the difference between hate speech detection and other related tasks such as offensive language, and cyber bullying.

Table 2: Summary of some previous work

Author	Year	Approach/Models	Data source and domain	Language	Future work
Mossie and Wang	2018	Naïve Bayes and Random forest	6,120, Facebook	Amharic	the number of dataset, the category of hate, utilize the information provided by Facebook so that, classification can be improved by expanding the feature space with profile information, list of followers and geo location and including other social media like Twitter, forums and other homepages beyond Facebook.
Huu, Trung and Pham	2019	Logistic Regression, SVM, NN, Nearest Neighbors, Decision Tree, Naive Bayes and Random Forest	25431, Facebook	Vietnamese	Using deep learning approaches for this task which includes the solution for the imbalance data to improve the performance.
Faris, Aljarah, Habib and Castillo	2020	Combination of CNN and LSTM	3696, Twitter	Arabic	To more deeply dive into deep learning approaches yet investigate the evolutionary optimization within the deep learning.
Bianchini, Ferri and Giorni	2018	feed-forward ANN	6000, Twitter and Facebook	Italian	The proposed system can certainly be improved, an idea can be to use clustering techniques to categorize the messages (cleaned and with the related features) in two subgroups (positive and negative) and

					then, for each comment, calculate how much this is more similar to negative comments or positive comments and add it as a feature.
Scheffler, Haegert, Pornavalai and Lee Sasse	2018	Logistic Regression, Linear SVM and Adaboost	5009, Facebook	German	To investigate the performance of classifier ensembles and Meta learning for hate speech detection.
Ruwandika and Weerasinghe	2018	SVM, Naïve Bayes, Logistic Regression, Decision tree and K-Means Clustering	6126, Facebook	English	To gain better results, the dataset should be expanded further. A semi-supervised classification approach can be used accomplish the task of annotating the dataset and training the models.
Malmasi and Zampieri	2017	linear SVM	14,509, Twitter	English	To investigate the performance of classifier ensembles and meta learning for this task.
Zhang, Robinson and Tepper	2017	Convolution-LSTM	6,594, Facebook and twitter	English	1) To further fine tune neural network to consider, e.g. stacking multiple convolutional layers which are good for extracting hierarchical features; 2) to integrate user-centric features, such as the frequency of a user detected for posting hate speech and the user's interaction with others; and 3) To study and quantify the difference between hate speech detection and other related tasks such as offensive language, and cyber bullying.

CHAPTER THREE

AFAAN OROMO LANGUAGE

3.1. Introduction

A natural language is used as a tool for communication and people use it for communication by combining phonologies to form words, by combining words to form phrases and by combining phrases to form sentences. Afaan Oromo language is a Cushitic language spoken by more than about 33.8% people in Ethiopia. There are more Oromo speakers in a foreign country, like In United States, Australia, Canada and different Europe cities people are speaking and communities are teaching their kids and foreigners those interested communications in Afaan Oromo also taking the Oromo class [18]. It contains own grammar rule like other natural languages. Most of them this grammar rules define characters of the languages. Afaan Oromo is written with a Latin alphabet called Qubee which was formally espoused in 1991[19]. Several versions of the Latin-based orthography had been used earlier, mostly by Oromo's outside of Ethiopia and by the Oromo Liberation Front (OLF) by the late 1970s (Heine 1986). In Oromia, Afaan Oromo is an official language of Oromia regional government. This chapter discusses the structure of Afaan Oromo word classes, phrases types and sentences formation with their types.

3.2. Word class in Afaan Oromo

Word class in Afaan Oromo language is some different from English word class. According to the [20], [21], Based on the context (galmaa) and form (uunka), In Afaan Oromo language there are five types of main word classes. Those are Noun (Maqaa), adjective (Maqibsi), Adverb (Xumibsi), Verb (Xumura) and pre- and post- position (Durduube).

3.2.1. Noun class

Like in other languages, Afaan Oromo nouns are words used to name or identify any of a class of people, places, or things. In Afaan Oromo, noun can be classified in to five categories proper noun, common noun, collective noun, material noun and abstract noun. In Afaan Oromo, Proper nouns are nouns that refer to specific people, places or things. For example, caalaa, sululta, jimma, ejersa, abdiisa and etc.

Common nouns refer to general, unspecific categories. For example **saree**(dog), **nama**(person), **hoolaa**(sheep), **biyya**(country), **laga**(river) and etc. are a common nouns.

In Afaan Oromo language, a collective noun describes a group of things, and it may be singular or plural, depending on how it is used. A singular collective noun refers to a group that functions as one unit or accomplishes the same action at the same time. **Uumata** (people), **bineensa** (animal,) **waldaa** (association) and etc. are examples of collective noun.

Material nouns refer to materials or substances from which things are made. For example, **damma**(honey). **Waraaqa**(paper), **dabboo**(bread), **qalama**(pen), **siree**(bed) and etc.
Abstract nouns are intangible ideas. They're not things people can see, smell, hear, or touch. Common examples comprise emotions, social concepts, political theories, and character traits (e.g., **jaalala**(love), **kallaqa**(creativity), and **diimmookraassii**(democracy)). Some example of Abstract nouns are, **bilisumma**(freedom), **hiyyummaa**(poverty), **dhukuba**(deasese), **fayya**(healthy),**laafina** (weakiness), **jabeenya**(strength), etc.

3.2.2. Verb class

Verb used to describe subjects action or state within a sentence. Therefore, without a verb any sentence cannot provide complete information. An Afaan Oromo verb consists minimally of a stem, representing the lexical meaning of the verb, and a suffix, representing tense or aspect and subject agreement. For example, in **dhufne** (we came), **dhuf-** is the stem 'come' and **-ne** shows that the tense is past and that the subject of the verb is first person plural. A main property of the Afaan Oromo verbs is that any word that comes at the end of a complete grammatical Afaan Oromo sentence is a verb [20].

Example in following sentences;

a) Caaltuun kaleessa kitaaba ishee **gate**
b) caalaan sanga diimaa guddaa **bite**
c) Margaan gara gaba **deeme**

In all above sentences the underline words **gate**"," **bite**"and "**deeme**" in respectively to sentences are verbs. Characters of these all words are to terminate the sentences within.

3.2.3. Adjective class

The main advantages of adjective are to give a clear explanation for a noun and to determine a noun itself. This means that, it can provide us more information about things represented by nouns and pronouns. In Afaan Oromo adjectives can come after noun or pronoun to modify noun or pronoun.

Example:

Isheen buna Mattu bite. (She bought Metu coffee).

In this sentence the word **Mattu** state type of **buna** (coffee) she bought. Also it has some position variance from English adjectives. It isn't come before noun or pronoun.

3.2.4. Adverb class

In Afaan Oromo language, adverbs are used to modify the coming verbs. Adverbs always come before the modified verb but it should be noted that any words come before verbs can't be always well-thought-out as an adverb.

Example:

Caalaan kaleessa dhufe. (Chala was come yesterday)

In this example, the adverb **kaleessa** (yesterday) go before the verb **dhufe** (came) that it modifies. However, it should be noted that any word that comes before a verb is not certainly an adverbs.

3.2.5. Pre- and Post- Position class

Prepositions in Afaan Oromo give meanings only if they combine with other words such as noun, adjective, verb, etc., unless they have no meaning. Pre- and post- positions tie with nouns, pronouns and with other words in a sentence. The main properties of pre- and post- positions are: they never use affixes and they don‘t contribute to form other words. According to [20], a preposition in Afaan Oromo ties a noun to an action (e.g., -achirraa deemi) or to another noun (-kopheen minjaalarraa jira). For the purpose of simplicity, this section will divide Afaan Oromo prepositions into two classes: prepositions and postpositions, with prepositions coming before the noun and postpositions coming after the noun they relay to. **Ala** (out, outside), **bira** (beside, with, around), **booda**(after), **cinaa**‘ (beside, near, next to) are example Postpositions. Some examples of prepositions are, **gara** (towards), **erga** (since, from, after), **hanga** (until), **hamma** (up to, as much as) and etc.

3.3. Punctuation marks

Punctuation is positioned in text to make meaning clear and reading at ease. Exploration of Afaan Oromo texts make known that different punctuation marks follow the same punctuation pattern used in languages that follow Latin writing system. Like English, the following are some of the most commonly used punctuation marks in Afaan Oromo language.

3.3.1. Full stop (.)

In Afaan Oromo Full stop (**Tuqaa)** is punctuation mark used at the end of a sentence and in abbreviations. Full stop is always written at the end of the sentence. We can use this punctuation mark to transmit meaningful information.

Example

a) Tolan adaama deeme.

b) lakkoofsa =Lakk.

c) Fakkeenya=Fkn.

d) Lakkoofsa sanduuqa poostaa=L.S. P

3.3.2. Question mark (?)

In Afaan Oromo, Question mark (**Mallattoo Gaafii**) is used in interrogative or at the end of a direct question. This must write at the end of the forceful speech. Question mark is used to ask question for others.

Example

a) Qorri ganna Kun akkam sitti jira?
b) Enyu atti?

3.3.3. Exclamation mark (!)

In Afaan Oromo, Exclamation mark (**Rajeffannoo**) is used at the end of command and exclamatory sentences. We can use this exclamation mark to express our emotions to others.

Example

a) Callisi!
b) Illa, hin turiin!
c) Ishoo ilma koo!
d) Of egii!

3.3.4. Comma (,)

Comma (**Qooduu**) is used to separate listing in a sentence or to separate the elements in a series. We can use this punctuation mark to separate a list of words.

Example

a) Boqonnaa 3,fuula 250, bara 2000
b) Naqamtee, wallaggaa
c) Barsiisaa Abdii Boruu, godina shawaa bahaa, aanaa dugdaa booraa, maqii
d) Guyyaa Jimaataa, Mudde 20, 1970.

3.3.5. Colon (:)

In Afaan Oromo, **Tuqlamee** (colon) is used to separate and introduce lists, clauses, and quotations, along with several conventional uses, and etc.

Example

a) Manni wantoota adda addaa irraa ijaarama: muka, dhagaa, cirracha, dhoqqee fi citaadha.
b) Ganama sa'a 4:45 irratti du'e.

3.4. Types of phrases

A phrase in Afaan Oromo is a structure which is constructed from one or more words in the Afaan Oromo language. Afaan Oromo Phrases are composed of either only head word or other words or phrases with the head combination. The other words or Afaan Oromo phrases that are combined with the head in phrase construction can be specifies, modifiers and complements. As linguistic book [20] report, the phrase is part of sentence. In Afaan Oromo language, there are five types of phrases which are described below:

3.4.1. Noun phrase (NP)

In Afaan Oromo, Noun phrases are consisting of a noun or pronoun and other related words that modify the noun or pronoun. A noun phrase consists of noun or pronoun as head word and other words which come after or before the noun. The simplest Noun phrases contains of a single noun (e.g. Gammadaa) or pronoun such as sii '(you), Isheen'(she), nutii'(we]), etc. A complex Afaan Oromo NP, can consists of a noun (called head) and other constituents (like complements, specifers, adverbial and adjectival modifiers) that modify the head from different aspects.

Example

- Caalaan muka xiqqoo kallessa mure.

In this example the head word is noun muka(tree) that indicates phrase is noun phrase. But the remaining words are used as relate words to more modify head word.

3.4.2. Verb phrase (VP)

It is composed of a verb as a head. In Afaan Oromo sentence structure verbs are found at the end of sentences.

Example

- caaltuun eerga barreessite.'(chaltu wrote a message)

In the above sentence, **barreessite**(wrote) is head word and **eerga** (message) is predicate. When categories sentence **caaltuun eerga barreessite** in to noun phrase and verb phrase, **caaltuun** is noun phrase **eerga barreessite** is verb phrase. The head word to phrase **eerga barreessite** is verb **barressite** that decided phrase is verb phrase.

3.4.3. Adjective phrase (AdjP)

In Afaan Oromo, Adjective phrase is composed of an adjective as head, and other constituents such as complements, modifiers, and specifiers. To decide part of sentence is as adjective phrase or word or group of words are adjective phrase; head word of that phrase must be adjective.

Example

margaan akkuma haadha cimaadha (marga is strong as his mother).

In the above example **akkuma haadha cimaa** is adjective phrase in verb phrase **akkuma haadha cimaadha**. For this Afaan Oromo adjective phrase head word is a word **cimaa**. Because; concentration of adjective phrase **akkuma haadha cimaa** is based on strengthens of marga not about marga like his mother. That is why **cimaa** is head word to the above phrase.

3.4.4. Pre- and post- positional phrase (PP)

Pre- and post- positional phrase is constructed from a pre- or post-position (PP) head and other constituents such as nouns, noun phrases, verbs, verb phrases, etc. in Afaan promo language. That means there is no phrase that construct from only pre- or post- position in Afaan Oromo. In Afaan Oromo prepositional phrases are not the same to other language like English. According to [25], there are not many possible forms for prepositional phrases in English, though adverbs can act as modifiers to prepositional phrases.

Example: prepositional and post positional phrase

a) Ulee**dhan**
b) Muka**rra**
c) Farda**an**
d) Mana **jala**
e) **Gara** muka

3.4.5. Adverbial phrase (AdvP)

In Afaan Oromo, an adverb phrase is a phrase that is adverb is head word to phrases. In this language, adverbs are used to modify the coming verbs. In Afaan Oromo, Adverbs always come before the modified verb but it should be noted that any words come before verbs cannot be always considered as an adverb. In Afaan Oromo, the key difference of adverb phrases from other phrase is adverb phrase is never come with other words [20]. Adverbs specify manner, time, place, cause, or degree. Sometimes it is single word in sentences. In following example all the underline one is adverb phrases.

Example

a) caalaan **kallessa** dhufe. (chala was come yesterday.)
b) tolaan **harra** ajjefame. (tola is killed today.)
c) Magaartun **haalaan** fiigdi.

In above sentence the underlined words **harra, haalaan** and **kallessa** are adverbs.

3.5. Sentence in Afaan Oromo

In Afaan Oromo, A sentence is group of words that are correctly structured to give one meaning. This definition is common in whichever language. But may be the structure is different depending on the language. In language like, English sentence must have subject, object, verb (S+O+V). But in Afaan Oromo, there is some dissimilarity. According to [20] [21], sentence is a complete thought or idea-subject + predicate. So, to be a sentence in Afaan Oromo it obligation to have a subject(S) and a Verb (V).

3.5.1. Types of sentences

Fundamentally Afaan Oromo sentences can be classified in to different types based on structurally and functionally [21].

3.5.1.1 Structurally

Structurally, Afaan Oromo sentences can be classified in to four categories. Specifically, simple, Compound, Complex and Compound Complex sentences.

A) Simple sentence

In Afaan Oromo, A simple sentence contains a subject and a verb, and it may also have an object and modifiers. But, it contains only one independent clause.

Example

a) Caalaan dhufe.
b) Margan gara mana buna deeme.
c) Margan buna dhuge.

B) Compound sentence

In Afaan Oromo like other language, a compound sentence contains at least two independent clauses. These two independent clauses can be combined with a comma and a coordinating conjunction or with a semicolon in this language.

Example

a) Margan daree seenee bahe.
b) Magaartuun ganama kaatee, fuula dhiqattee, ciree ishee nyaattee mana barumsaa deemte.

C) Complex sentence

In Afaan Oromo language, a complex sentence contains at least one independent clause and at least one dependent clause. This Dependent clause can refer to the subject (who, which) the sequence/time (since, while), or the causal elements (because, if) of the independent clause.

Example

a) Yoo adaama deemte, meeshaa naa bitta.
b) Bokkaa cimaa waan roobeef, lagni guutee riqicha cabsa.

D) Compound-Complex Sentences

Sentence types can also be combined from different types. In Afaan Oromo, A compound-complex sentence contains at least two independent clauses and at least one dependent clause.

Example

a) Yoommuu deemtuu fi yommuu deebitu na dubbisii darbi.

b) qillensarras kaattu, lafarras arreedu walgahin finfinneedha.

3.5.1.2 Functionally

On the basis of the meaning that they convey (function), in Afaan Oromo sentences can be divided into four main types. They are:

A) Declarative Sentence

When a sentence tells a plain statement or expresses an opinion in Afaan Oromo, it is termed as a declarative sentence. In simple words, an ordinary statement is identified as a declarative sentence.

Example

a) Foonichi cooma.

b) Barattootni daree jiru.

B) Imperative Sentence

While a sentence is in the form of order, command, instruction or a request, it is known as an imperative sentence. Imperative sentence can end with a period or exclamation mark or a question mark.

Example

- Hojii manaa kee hojjedhu [Do your homework]

C) Interrogative Sentence

When an Afaan Oromo sentence denotes an interrogative characteristic and ends with a question mark, it is known as an interrogative sentence.

Example

a) Caalaan essa jira?

b) Ati yoom dhufte?

c) Ati yoom dhalatte?

D) Exclamatory sentence

In Afaan Oromo, A sentence that denotes different expressions like shock, surprise, anger, etc. is known as an exclamatory sentence. Exclamatory sentence should always end with an exclamation mark.

Example

a) Nan bade!

b) Yaa rabbi na baasi!

CHAPTER FOUR

SYSTEM DESIGN AND IMPLEMENTATION

4.1. Introduction

In this Section we discuss the overall design of our system, Online Hate Speech Detection for Afaan Oromo language. At the start of this Section we explain a little more about our data set which is used for training and testing. The next Section discusses the general overview of the proposed system architecture from the perspective of the system's flow of operations. Finally, detailed explanations of each model along with subcomponents in each phase are discussed.

4.2. Approach used

Based on related works we reviewed different approaches have been used throughout the state-of-art approaches of hate speech detection. Researchers have their own point of view with evidence to apply a particular approach in such domain. To make it precise we don't need to go detail explanation of each class of approaches such as keyword based, classical machine learning based and connectionist or neural network-based approaches, because we already stated them with their pros and cons in Section 2.3.

4.3. System Architecture

This section explains the overall architecture of the proposed system. The system has three phases the training, the testing and the prediction phase.

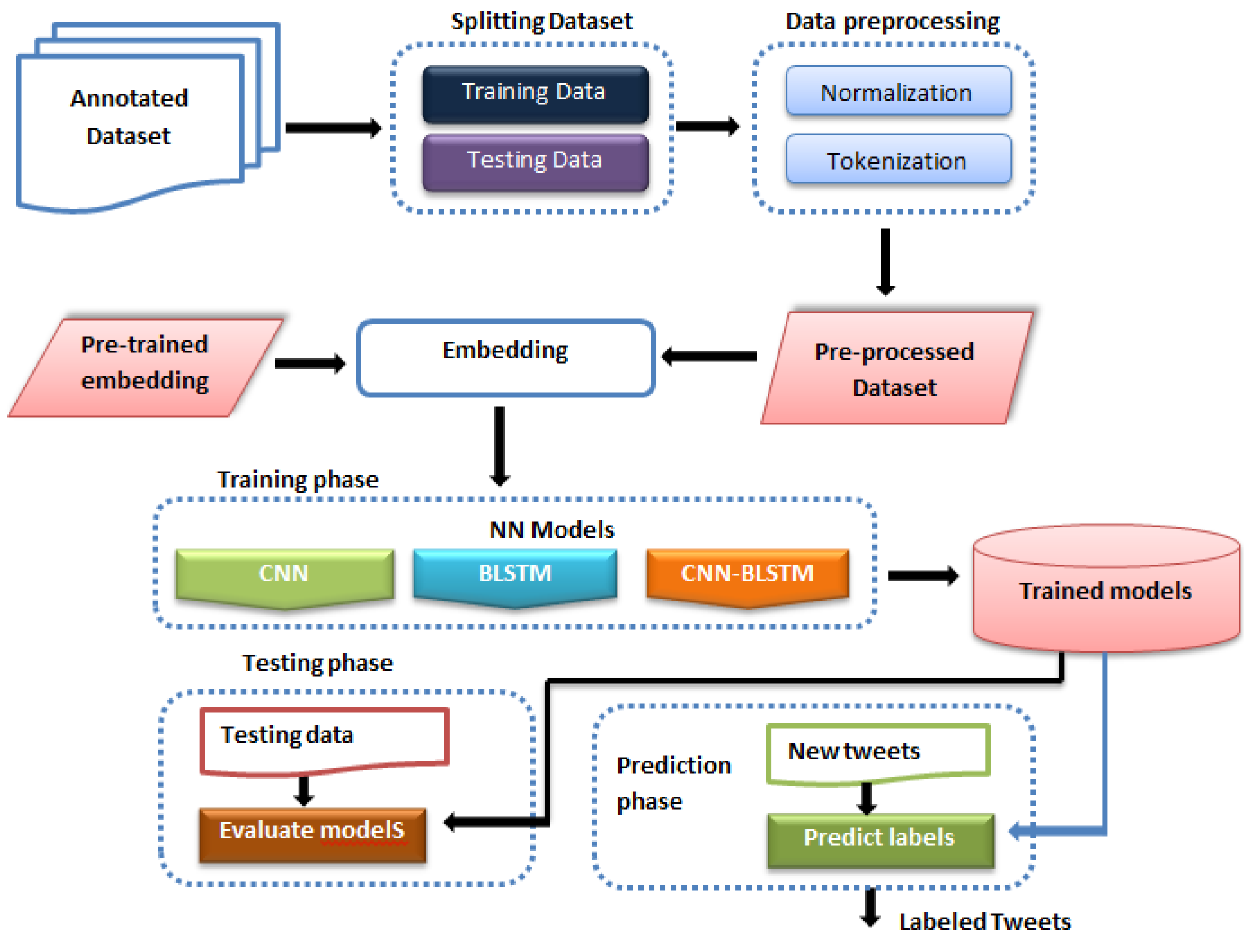


Figure 5: Architecture of our proposed work

4.3.1. Annotated Dataset

Annotated Dataset is a collection of text collected from Facebook and twitter. This text holds both the training and testing data of our model. Unlike other languages for Afaan Oromo, there are no any standardized annotated publicly available corpora like English or other languages for hate speech detection. Targeting at classifying the hate level across Facebook and twitter for Afaan Oromo language users, with the help of linguistic and legal professionals, we have prepared a corpus of text (post and comments) collected from Twitter and Facebook public pages of Ethiopian newspapers, individual politicians, activist, television, Radio Broadcast and groups. These pages typically post discussions passing across a variety of political, race, gender, disability and religious topics. By doing so, we could take both casual conversations and hated posts and comments going on Facebook and twitter. We have engaged a different Facebook and Twitter crawler, which adventure the Graph API to retrieve the content of the comments from Facebook posts and Twitter using Facepager and save them to a CSV/Excel file. Facebook and Twitter platforms are selected to collect data for the subsequent reasons. They are the most vital platform for reaching out to online viewers. Comparative studies have revealed how in countries with inadequate Internet service, like Ethiopia, specifically Facebook has become nearly a synonym for the Internet, a platform through which user's access information, services, and take part in online communications.

The entire number of collected dataset after removing duplicates and irrelevant tweets is 14,077. we have collected 9030 post and comments From Facebook and 5047 tweets from tweeter social media website. The collected data were annotated by two annotators one from legal profession and the other is from Afan Oromo profession. Based on the overall perceived meaning of the tweet (or Facebook post and comments) they annotated into strong hate, weak hate and neutral. The total number of Strong Hate Speech tweets is 2617, while the number of weak hate speech tweets is 4567 and neutral tweet is 6893.

Classes	Facebook	Tweeter	Total
Weak Hate	3025	1542	**4567**
Strong Hate	2107	510	**2617**
Neutral	3998	2895	**6893**
Total	**9030**	**5047**	**14,077**

Table 3: Summary of classes and Dataset Source

A tweet is annotated depending on the agreement of two annotators with the same label. Let's say, the first annotator may label tweet T with Strong hate and the second annotator may label tweet T with Strong hate, then Tweet T is taken as having Strong hate. Just in case of dissimilarity; if both annotator's allocated dissimilar labels to the same Tweet, third annotator was taken into account. Depending on the choice of the third annotator from legal profession, whether she/he agrees with annotator-1 or annotator-2, a tweet was labeled subsequently. Tweet was excluded, in case the third data annotator neither agrees with the first one nor with the second one. Moreover, we measure the inter-annotator agreement is calculated by using Kappa statistic. "Cohen's kappa "is a measure of the agreement between two raters who determine which category a finite number of subjects belong to whereby agreement due to chance is factored out. The two raters either agree in their rating (i.e. the category that a subject is assigned to) or they disagree; there are no degrees of disagreement (i.e. no weightings)" [62].

Annotator 2	Classes	Annotator 1			
		Weak Hate	Storing Hate	Neutral	**Total**
	Weak Hate	4300	100	167	**4567**
	Storing Hate	400	2006	211	**2617**
	Neutral	393	500	6000	**6893**
	Total	**5093**	**2606**	**6378**	**14077**

Table 4: Interrater reliability: The kappa statistic

Annotator-1 said 5093 tweets is Weak Hate, 2606 tweet is Storing Hate and 6378 tweets is neutral class. Annotator-2 said 4567 tweets is Weak Hate, 2606 tweet is Storing Hate and 6893 tweets is neutral. The labeling in agreement is located on the main diagonal of the table in table 4. Thus, the percentage of agreement (Po) is 87.42%. But this table includes agreement that is due to chance. E.g. Weak Hate represents 36.17% of Annotator-1 labeling and 32% of Annotator-2 labeling. Thus 11.57% of the agreement about this labeling is due to chance, i.e. 11.57% * 14077 = 1629.32 of the cases. Generally, the total labeling due to chance agreement (Pe) is 37.36%. Then, the Cohen's kappa is calculated with the following formula: -

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

Where, Po represents the actual observed agreement and Pe represents chance agreement. The Cohan's kappa statistic is often used to test interrater reliability. The significance of rater reliability lies in the fact that it represents the extent to which the data collected in the study are correct representations of the variables measured. Measurement of the extent to which data annotators (raters) assign the same score to the same variable is called interrater reliability. Then, we calculated the kappa and the inter-annotator agreement among 2 annotators have resulted in Cohan's kappa of 0.799, which is a very good level of agreement. The scale of Kappa value interpretation is the as follows:

Kappa < 0: No agreement

Kappa between 0.00 and 0.20: Slight agreement.

Kappa between 0.21 and 0.40: Fair agreement

Kappa between 0.41 and 0.60: Moderate agreement

Kappa between 0.61 and 0.80: Substantial agreement and

Kappa between 0.81 and 1.00: Almost perfect agreement.

Texts written by users on social network websites like Facebook and tweeters contain lots of noise that can significantly affect the results of the hate speech detection procedure. In social network, occasionally users use informal language while articulating their opinion. Typically, natural language like English uses automatic techniques to correct spelling and grammar, translate from one language to another language. Though, for Afaan Oromo such platforms and resource do not exist online even if developed for the needs of fulfillment for the Degree. So that, we considered this problems into account and addressed several concerns such as; correct spelling errors and grammar of some text manually, correct short form or contraction and replication of characters, because some of them may useful in the tweets label, and a word written in other language mixed with Afaan Oromo was converted into Afaan Oromo by linguistic professionals.

Afan Oromo is a lowland east Cushitic language which has tens of millions of native speakers in Ethiopia and in neighboring countries such as Kenya and Somalia. Afan Oromo is dialectic language. A dialect is a variety of Afan Oromo which is associated with a particular region and or social class. To state the comprehensible, speakers from different geographical regions speak Afan Oromo somewhat differently. While we are collecting dataset from social media, we considered the different variety of Afan Oromo dialects. So, we have collected our dataset from different classes of Afan Oromo dialects. And then, to train the deep learning in demand to handle the complexities of Afan Oromo dialects, which makes the hate speech detection system to a certain extent challenging. According to (Feda Negessa, 2015) Afan Oromo dialect is divided into eleven genetic groups such as western (Wollega, Jimma, Ilubabor), eastern (Harar), central (show), south-eastern (Arsi, Bale), south (Borana, Guji), north (Wollo, Raya). Our dataset consists of the following diversities of dialects.

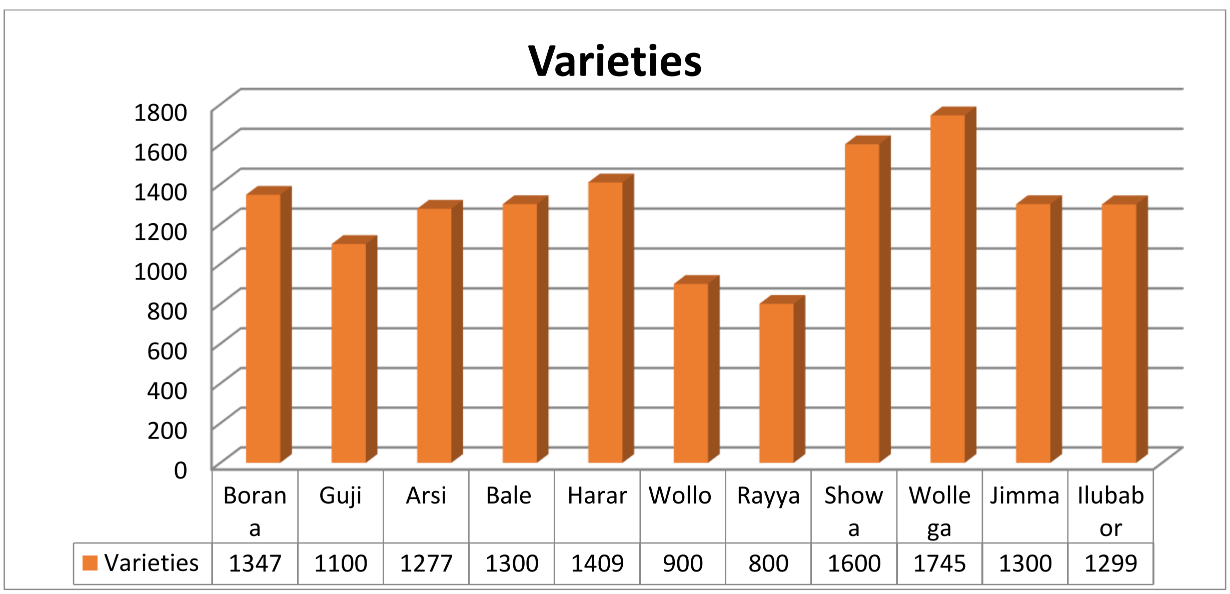


Figure 6: show a source of dataset from varieties of Afan Oromo dialects

4.3.2. Data splitting

Data splitting is the act of partitioning Annotated dataset into two parts, usually for cross- validator purposes. In this research, we split our dataset into training and testing data. Training portion of the dataset is used to train a predictive model. And the other to evaluate the hate speech detections model performance. We used sky learn python library to split our dataset into training and testing data.

4.3.3. Preprocessing

Preprocessing is the most significant and preliminary task of feature engineering approaches in many NLP applications which is used to clean and made ready the dataset for the further processing. The main role of data preprocessing is formatting or normalizing the input data, so that later tasks can be computed easily. The text preprocessing component handles different language specific issues that are imposed by the nature of the language to make the data ready for remaining phases. In order to get good results, language dependent text preprocessing should be performed before hate speech detection is implemented. Text preprocessing is the step by which the text is made comfortable to the learning algorithm or any other component which is in need of the relevant text to proceed to the next level or doing its action accordingly. The preprocessing step is simply a removal of non-informative words or characters from the text in order to save the systems computational resources as well to enhance its performance by getting rid of those unwanted jumbles.

In this section, we will talk about the basic steps of text preprocessing. These steps are needed for transferring text from human language to machine readable format for more processing.

4.3.3.1 Normalization

Normalization is a process that transforms a list of words to a more uniform sequence. This is useful in preparing text for further processing. By transforming the words to a standard format, other operations are able to work with the text and will not have to deal with issues that might compromise the process. The normalization process can increase text matching. After a text is acquired, we start with text normalization. Text normalization includes:

Convert text to lowercase: - Lowercasing all Afan Oromo text data, although commonly overlooked, is one of the easiest and most useful forms of text preprocessing. It is applicable to most text mining and NLP problems and can help in cases where your corpus is not very large and meaningfully helps with consistency of expected output. For example, if trainee a word embedding model for similarity lookups. It is easy to found that different variation in input capitalization (e.g. *'Gadaa' vs. 'gadaa'*) gave different types of output or no output at all. This was probably happening because the dataset had mixed case occurrences of the word ***'Gadaa'*** and there was insufficient evidence for the neural-network to effectively learn the weights for the less common version. This type of issue is bound to occur when the dataset is honestly small, and lowercasing is a great way to deal with sparsity issues.

Remove numbers: - numbers are removed if they are not relevant to the analyses. Typically, regular expressions are used to remove numbers.

Remove Punctuation: - Punctuation also will be removed. Punctuation is basically the set of symbols [!” #$%&()*+,-./:;<=>?@[\]^_{|}~]:

Remove HTML Tags: - since, our dataset is web scraped, there is chances that our dataset will contain some HTML tags. Since these tags are not useful for our natural language processing tasks, it is better to remove them.

4.3.3.2 Tokenization

Tokenization is the process of tokenizing or splitting Afaan Oromo string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph. This is the task of splitting Afaan Oromo texts in to piece of tokens, which are disjoint and meaning full texts. Sometimes it can be defined as given a sequence of Afaan Oromo characters and a defined document unit, tokenization is the task of chopping it up into pieces, perhaps at the same time throwing a way certain characters such as punctuations. A token is an instance of a sequence of Afaan Oromo characters in some particular document that are grouped together as a useful semantic unit for processing.

In order to get our computer to understand any Afaan Oromo text, we need to break that word down in a way that our machine can understand. That’s where the concept of tokenization in Natural Language Processing comes in. As tokens are the building blocks of Natural Language, the most common way of processing the raw text happens at the token level. For example, Transformer based models the State of The Art neural network architectures in Natural Language Processing (NLP) process the raw text at the token level. Similarly, the most popular neural network architectures for NLP like RNN, GRU, and LSTM also process the raw text at the token level. Hence, tokenization can be broadly classified into three types’ word, character, and sub word (n-gram characters) tokenization [27]. In our work, word level tokenization is used.

Word Tokenization: - it is the most commonly used tokenization algorithm. It splits a piece of a given text into individual words based on a certain delimiter. Depending upon delimiters, different Afaan Oromo word level tokens are formed. Pre-trained Word Embedding such as Word2Vec and GloVe comes under this type of tokenization. For example, the sentence “*Mootiin tiissisaa balfa irraa* *waan ooltuuf yaada gaarii yaaduu hin dandeessu.*" When we tokenize this sentence, it produces *('Mootiin', 'tiissisaa',' balfa',' irraa',' waan',' ooltuuf',' yaada',' gaarii',' yaaduu',' hin',' dandeessu'*). One of the major problems with word tokens is dealing with Out of Vocabulary (OOV) words. Out of Vocabulary words refer to the new words which are encountered at testing. These new words do not exist in the vocabulary. Hence, these methods fail in controlling OOV words. Another problem with word tokens is connected to the size of the vocabulary. Generally, pre-trained models are trained on a large volume of the text data. So, just imagine building the vocabulary with all the distinct words in such a large corpus. This explodes the vocabulary! `

Character Tokenization: - it splits apiece of Afaan Oromo text into a set of characters. For example, a word '*abaaramaa'* can be tokenized in to character *('a-b-a-a-r-a-m-a-a').* It overwhelms the drawbacks we saw in Word Tokenization. It handles OOV words comprehensibly by preserving the information of the Afaan Oromo word. Character Tokenization breaks down the OOV word into characters and denotes the word in terms of these characters. Character Tokenization also restricts the size of the vocabulary. The Drawbacks of Character Tokenization is it can solve the OOV difficult but the length of the input and output sentences increases promptly as we are representing a sentence as a sequence of characters. As a consequence, it becomes challenging to learn the relationship among the characters to form meaningful words.

Sub word Tokenization: - it splits the piece of text into sub words (or n-gram characters). For example, words like *koreewwan* can be segmented as *koree-wwan, daandiiwwan as daandii-wwan,* and so on.

4.3.4. Word Embedding

Word embedding is basically a form of word representation that ties the human understanding of language to that of a machine. They contain a learned representation of text in an n-dimensional space where words that have the same meaning have a similar representation. Meaning that two similar words are represented by almost similar vectors that are very closely located in a vector spaces. These are important for solving most Natural language processing problems specifically hate speech detection. Therefore, when using word embedding's, all individual Afaan Oromo words are represented as real valued vectors in a predefined vector space. Each Afaan Oromo word is tie to one vector and the vector values are learned in a way that resembles a neural network [28]. We have used the CBOW word2vec word embedding model and embedding layer of keras python libraries to carry out this representation. Therefore, this technique is used to embed our dataset to the corresponding vector representations and this vector representation is fed to the three different deep learning models.

The word2vec is a collection of similar models that are used to make word embedding's. These models are simple, neural networks with two-layer that are used to train and build linguistic contexts of phrases. Word2vec gets an input of large data of text and creates a vector space, generally hundreds of dimensions, where each individual word in the data is being nominated to a corresponding vector space. Those word vectors are placed in the vector space so that words with similar contexts in the corpus are positioned close one to another in the space. It was developed at Google by Tomas Mikolov and his group. The algorithm developed was subsequently analyzed and many other researchers have explained it too. The method used two different learning styles to learn the word embedding: CBOW model and Skip-Gram Model for distributed representations of words whose aim is to minimize computational complexity. The Continuous Bag-of-Words is a model that predicts the actual word based on its content while learning the embedding. The CBOW uses the average value of the word embedding of the context to predict the current word. And the skip-gram is similar to CBOW, but as an alternative of predicting the word from context, it tries to maximize the classification of a word based on another word in the same sentence. It is model that predicts the words around for a given actual word.

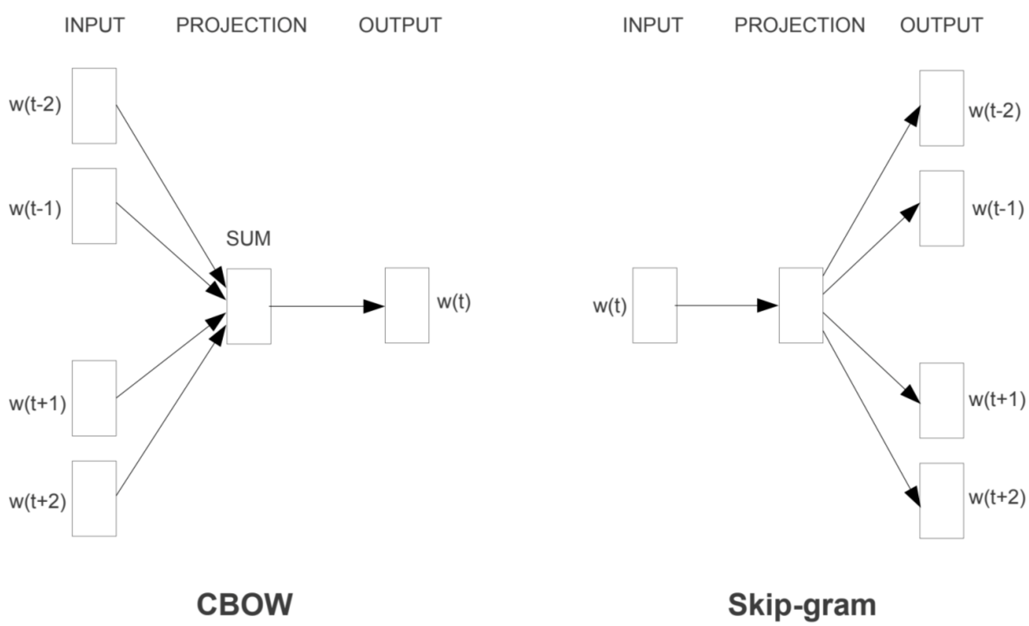


Figure 7: Word2Vec Training Models [56]

4.3.5. Training phase

The training dataset in deep learning is the data that is used to train the model for accomplishing various actions. This is training data is the ongoing development process models learn with different algorithm to train the device to work automatically. So that, the sample of data used to fit the model is called training data. This training stage is for building the model. For this research, we used the Convolutional neural network (CNN), Bidirectional Long Short-Term Memory (BLSTM) and the hybrid of CNN and BLSTM Neural Network to develop trained model.

4.3.5.1. Convolutional Neural Network Model

The first proposed deep learning is convolutional neural network (CNN): - which Inspired by P. Badjatiya et. al [15]'s work on using CNNs for hate speech detection, they leverage CNNs for hate speech detection. Convolutional neural network is a class of deep, feed-forward ANN where connections among nodes do not form a cycle & use a variation of multilayer perceptron's designed to need minimal preprocessing. Convolutional neural network is used in computer vision; however, they have recently been applied to various natural language processing tasks and the results were promising.

CNN models were developed for image classification, in which the model accepts a 2-dimensional input representing an image's pixels and color channels, in a process called feature learning. This same process can be functional to 1-dimensional sequences of Afaan Oromo data. The model extracts feature from sequences Afaan Oromo dataset and maps the internal features of the sequence. A 1D CNN is very effective for deriving features from a fixed-length segment of the whole dataset, where it is not so vital where the feature is located in the segment. In our task we implement 1D convolution layer, because our task is natural language processing.

The network we will build in this task looks roughly as follows: The first layers embed words into low dimensional vectors. The next layer makes convolutions over the embedded word vectors via multiple filter sizes. For example, sliding over 3, 4 or 5 words at a time. Next, we max-pool the result of the convolutional layer into a long feature vector, then add dropout regularization, and classify the result using a softmax activation layer.

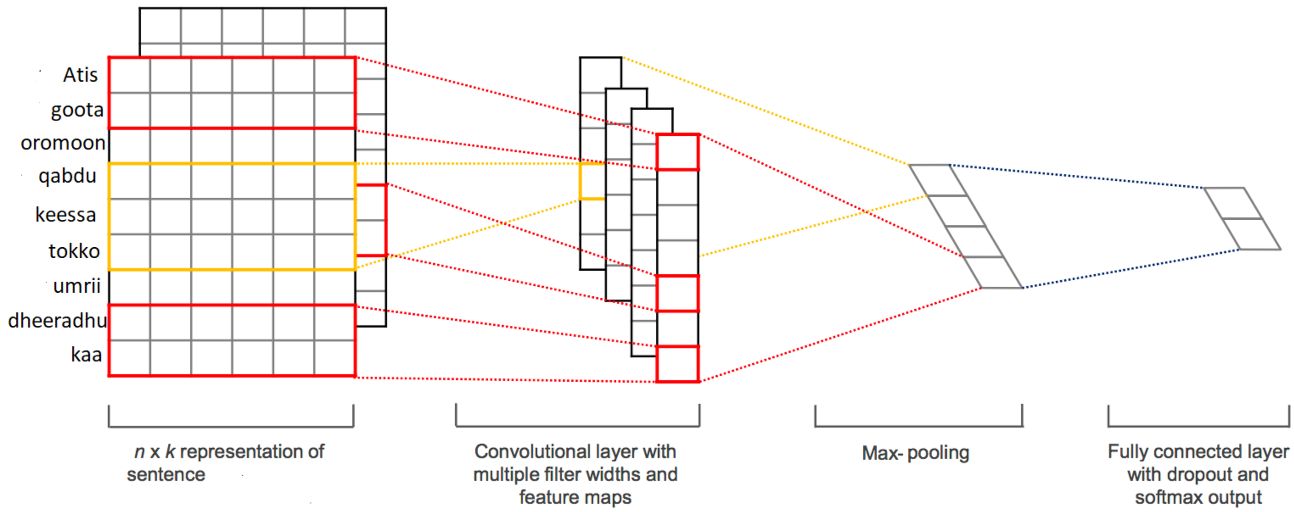


Figure 8: CNN model [58]

A CNN contains an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are covered by the activation function and the last convolution [57]. In a CNN, the hidden layers consist of a layer that performs convolutions. Typically, this includes a layer that does multiplication or other dot product, and its activation function is commonly ReLU. This is followed by other convolution layers such as pooling layers and fully connected layers.

Convolutional layer

The task of this layer is to reduce the number of weights by extracting higher level features from the input matrix. As known, each row of a matrix is a vector that denotes a Afaan Oromo word. Typically, these vectors are Afaan Oromo word embedding's or low-dimensional representations like word2vec or GloVe, but they can also be one hot vector that indexes the word into a vocabulary [42, 43].

Pooling layer

A pooling layer is another layer added after the convolutional layer, in convolutional neural network. Specifically, after a non-linearity has been applied to the feature maps output by a convolutional layer. The pooling layer operates upon each feature map distinctly to create a new set of the same number of pooled feature maps. These layers are intended to subsample their input, which is the output of the convolutional layer that is passed to these layers. The advantage of pooling is to convert the joint feature representation into a more functional one that preserves important information while discarding irrelevant details [45]. Furthermore, as classification needs fixed size output matrix, this layer also affords that. This lets the use of variable size sentences, and variable size filters, but all the time obtaining the same output dimensions to feed into a classifier [46]. In our work we used the 1D Global max pooling block which takes a 1-dimensional tensor of size (input size) x (input channels) and computes the maximum of all the input size values for each of the input channels, since global max pooling are supported by Keras via the GlobalMaxPooling1D classes.

Fully connected layers

Fully connected layers in neural networks are “those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are a full connected layer which compiles the data extracted by previous layers to form the final output” [59].

4.3.5.2. Bidirectional Long Sort Term Memory Model

The second proposed deep learning model is Bidirectional Long Sort Term Memory. Bidirectional Long Sort Term Memory is an extension of traditional LSTMs that can improve model performance on sequence classification problems. The notion of Bidirectional Recurrent Neural Networks (RNN) is straight forward. It involves duplicating the first recurrent layer in the network so that there are now two layers side by side, then providing the input sequence as is as input to the first layer and providing a reversed copy of the input sequence to the second recurrent layer [29]. Bidirectional LSTMs are an extension of old LSTMs that can progress model performance on sequence classification problems. In problems where all time steps of the input sequence are there, Bidirectional LSTMs train two instead of one LSTM on the input sequence. The first on the input sequence as is and the second on a reversed replica of the input sequence. This can provide further context to the network and result in faster and even fuller learning on the hate speech detection. Bidirectional LSTMs are supported in Keras via the Bidirectional layer wrapper. Bidirectional layer wrapper takes a recurrent layer (e.g. the first LSTM layer) as an argument.

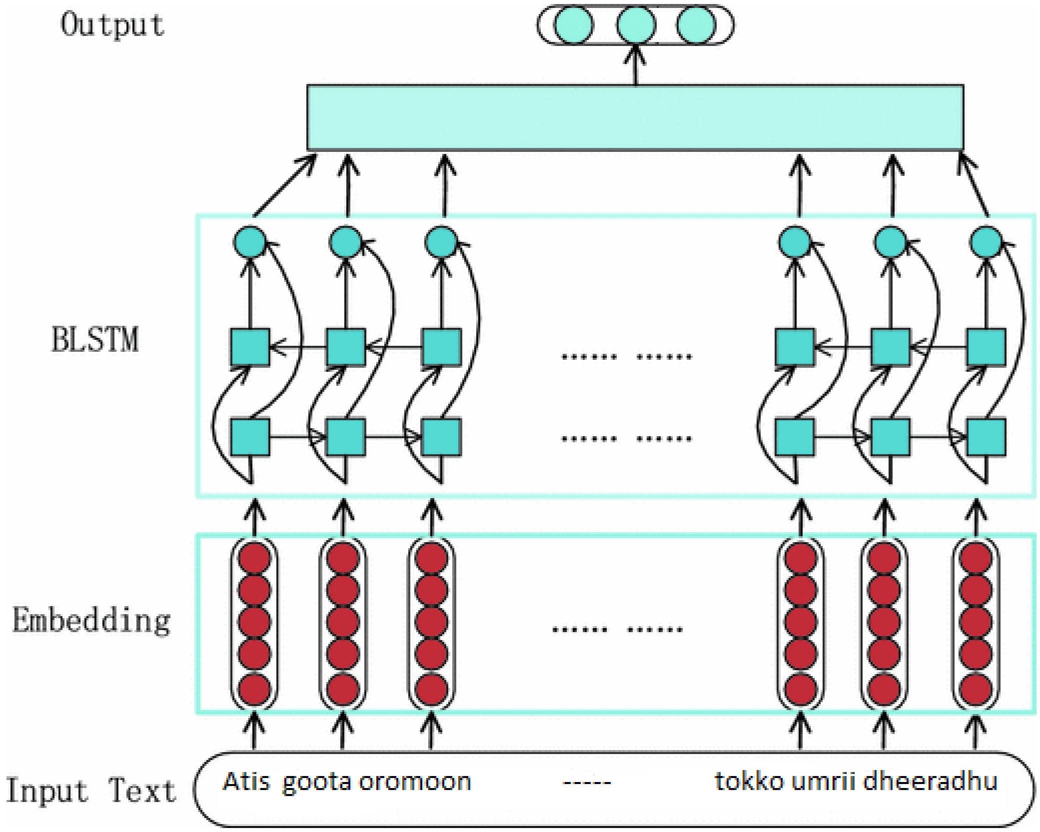


Figure 9: BLSTM model

We built bidirectional long short-term memory which contains a different layer and each layer does different works. The first layer, embedding layer is used to create word vectors for incoming words. It sits between the input and the BLSTM layer, i.e. the output of the Embedding layer is the input to the bidirectional long short-term memory layer. The next layer is Bidirectional layer wrapper. This wrapper takes a recurrent layer (e.g. the first LSTM layer) as an argument. It also allows specifying the merge mode that is how the forward and backward outputs should be combined before being passed on to the next layer. The third layer is global max pooling layer, which performs down-sampling by computing the maximum of the height and width dimensions of the input. The last layer is dense layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output. The layer has a weight matrix W, a bias vector b, and the activations of previous layer a. finally we add dropout regularization, and classify the result using a sigmoid activation.

4.3.5.3. The Hybrid models

The third deep learning model is the hybrid of CNN-BLSTM based approach. The implemented approach is a hybrid of convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) network. The model starts with the embedding layer, the Convolutional layer, the max pooling layer, the bidirectional long short-term memory layer and lastly the dense layer.

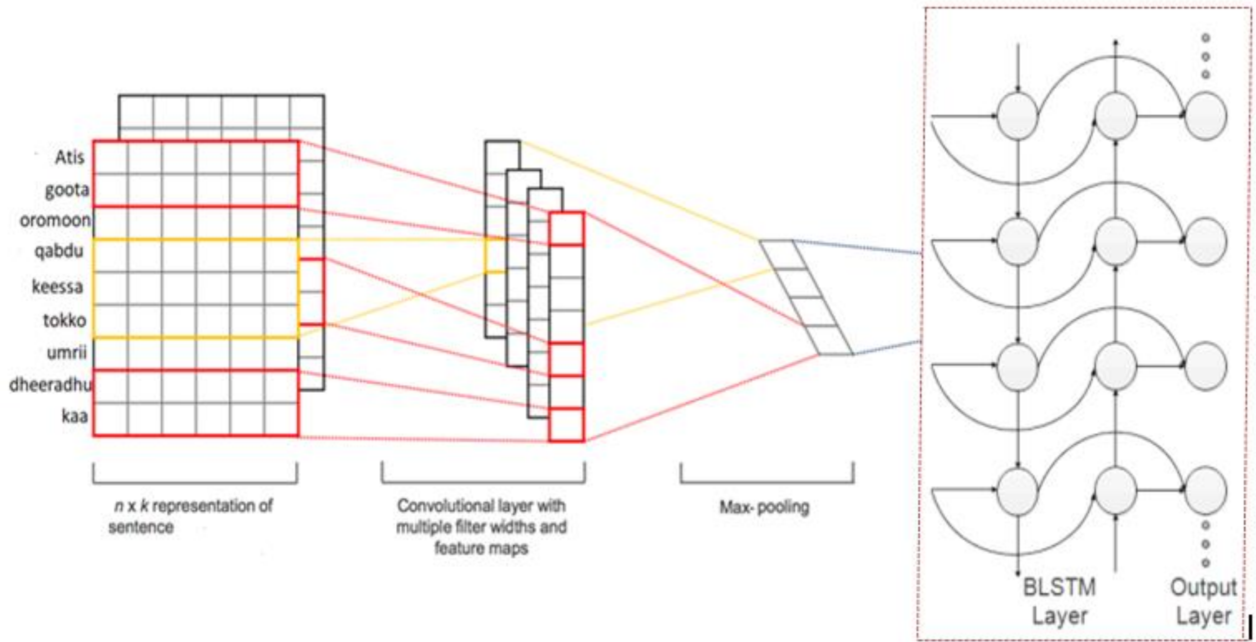


Figure 10: CNN-BLSTM based model

Dense layer

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer performs the following operation on the input and return the output. This means that we are using the dot product among our input tensor and whatever the weight kernel matrix is featured in our dense layer. Then, we add a bias vector (if we need to have a bias) and take an element wise activation of the output values (certain sort of function, linear or, more often, non-linear!) [31].

Activation (dot (input, kernel) + bias) =output

Where, input denote the input data, kernel denote the weight data, dot denote numpy dot product of all input and its corresponding weights, bias denote a biased value used in machine learning to optimize the model and Activation which denotes the activation function.

Dropout

Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" arbitrarily. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not useful to the neuron on the backward pass. As our deep learning learns, neuron weights settle into their context within the network. Weights of neurons are adjusted for specific features providing some specialization. Adjacent neurons become to depend on this specialization, which if taken too far can result in a fragile model too specialized to the training dataset. This dependent on context for a neuron during training is referred to complex co-adaptations. Dropout is easily implemented by randomly selecting nodes to be dropped-out with a given probability (example, 20%) each weight update round. This is how Dropout is implemented in Keras [30].

Optimizers

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. Optimizers are used to solve optimization problems by minimizing the function. Various optimizers are researched within the last few couples of years each having its benefits and drawbacks [60]. The varieties of optimizers are Gradient Descent, Stochastic Gradient Descent (SGD), Mini Batch Stochastic Gradient Descent (MB-SGD), SGD with momentum, Nesterov Accelerated Gradient (NAG), Adaptive Gradient (AdaGrad), AdaDelta, RMSprop and Adam. In our work we used Adam optimizer because, Adam realizes the benefits of both Adaptive Gradient (AdaGrad) and RMSProp. It is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Empirical results show that it works well in practice and compares favorably to other stochastic optimization methods.

4.3.6. Testing phase

The dataset which is not in training dataset that is used to evaluate the training model is called Testing data. The testing phase follows the same process as we follow in the training phase. The variance lies in the data we are going to use. We follow the same way and the same procedure starting from the preprocessing of the data. In our work, after trained model was saved, we used saved trained model to evaluate the testing data.

4.3.7. Prediction phase

Prediction phase is the process to input new tweet in the form of training feature and display the categorized tweets into labels classes. In the Afan Oromo hate speech detection system the input data is a new tweet. After trained models were saved and testing model was evaluated, we can predict new inputted tweet by using these models.

CHAPTER FIVE

EXPERIMENT

5.1 General pipelines

To perform the experiment, we followed the following pipelines: First we prepared the dataset with linguistic and legal professionals. Next, we designed and developed the functional BLSTM, CNN and CNN_BLSTM (the hybrid of BLSTM and CNN Neural Networks) models. We also selected the Adam optimizer. It is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data and Adam is computationally efficient and requires little memory [48]. And next, we selected the sparse categorical crossentropy loss function to estimate the loss (error) of the model during training. It is a loss function that is used in multi-class classification tasks. Sparse categorical crossentropy loss function preserve integer as chars'/ multi-class classification labels without transforming to one-hot label. Sparse categorical cross-entropy loss function can be used in keras for multiclass classification by using sparse_categorical_crossentropy while calling the compile () function.

5.2 Development Tool

To design and develop the hate speech detection for Afaan Oromo Language prototype we used different tools such as smart draw, anaconda, Jupyter notebook, and keras python library.

5.2.1 Python programming language

We used python 3.7.6 to develop the hate speech detection. We used keras deep learning python library to develop the prototype. Keras is a high level neural networks library that is running on the top of TensorFlow, CNTK, and Theano. Using Keras in deep learning lets for easy and fast prototyping as well as running seamlessly on CPU and GPU [50]. It is an Open Source deep learning library written in Python. In addition, this library supports both convolutional and recurrent neural networks.

5.2.2 Anaconda and Jupyter Notebook

Anaconda is a Python distribution (prebuilt and preconfigured collection of packages) that is commonly used for data science. The Anaconda distribution contains the Conda package manager in addition to the preconfigured Python packages and other tools. The Conda package manager can be used from the command line to set up Python environments and install additional packages that arisen with the default Anaconda distribution [53]. We use this python distribution to edit and develop our model because anaconda includes Jupyter Notebook. The Jupyter Notebook application allows us to easily create and edit documents that display the input and output of a Python language script. So, this helps us to minimize our effort and saves our time.

5.2.3 Facepager

Facepager enables to collect public data from platforms on the social media (such as Facebook, YouTube, twitter and other website on the bases of APIs and web scrapping). Facepager is open source, so it is freely accessible. Facepager reduces the technical obstacle to collect data which is particular relevant for scientific work. We used facepager version of 4.22 tools to collect dataset from Facebook and twitter in order to train our models.

5.2.4 Smart Draw

SmartDraw is a diagram tool used to make system architecture and another diagram. It is very easy to draw diagrams in smartDraw. This is very simple to draw the diagrams. As a result, this helps us to minimize our effort and saves our time.

5.2.5 Gensim

Gensim is an open source a python library for unsupervised topic modeling and naturel language processing (NLP), using modern statistical machine learning. It is licensed under the OSI approved GNU LGPLv2.1 license. This library supports all python versions that have not reached their end of life. The algorithm in Gensim such as word2vec, FastText and etc., automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training data. In our work we used gensim version of 3.8.0.

5.3 Model Training

For this research, we have prepared a corpus of text (post and comments) collected from Twitter and Facebook public pages of Ethiopian newspapers, individual politicians, activist, television, Radio Broadcast and groups. These pages typically post discussions passing across a variety of political, race, gender, disability and religious topics. The entire number of collected tweets after removing duplicates and irrelevant tweets is 14,077. The collected tweets were annotated based on the overall perceived meaning of the tweet into strong hate, weak hate and neutral. The total number of Strong Hate Speech tweets is 2617, while the number of weak hate speech tweets is 4567 and neutral tweet is 6893. After that, we have shuffled the data using the skylearn library in order to mix the dataset. After we have shuffled the data we have splitted 80 %(11,261) for training and 20 % (2,816) for testing data by using percentage split method of the skylearn python library.

5.3.1 Pre-trained word embedding's

Word embedding is an important input for training Neural Networks. Word Embedding has fundamental benefits in specifically, it is a more capable representation (dimensionality reduction) and also it is a more expressive representation (contextual similarity). In order to improve performance and obtain more accurate classification results in this excrement, we have trained neural networks using 200-dimensional word vectors to investigate the performance impact of using Pre-trained word embedding. Word2vec is a class of models that represents a word in a large text corpus as a vector in n-dimensional (typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space) conveying similar words closer to each other. It is similar to an auto-encoder, encoding each word in a vector, but rather than training against the input words through reconstruction, word2vec trains words against other words that neighbor them in the input corpus. Word2vec does so in one of two ways, either using context to predict a target word (a technique known as continuous bag of words, or simply CBOW), or using a word to predict a target context, which is called skip-gram.

Hereafter we have created a Word2Vec model for word embedding. In text classification tasks such as hate speech detection, word embedding's created with word2vec are used to improve performance and obtain more accurate classification results. We prepared Afan Oromo pre-trained data by using Gensim library. This library is fairly easy to use module which inherits CBOW and Skip-gram. Word vector embedding used in this research was obtained by applying a continuous bag of words (CBOW) word2vec model to hate speech detection of Afan Oromo. We use this method because CBOW is faster and represents frequent words better and when it comes to memory consumption, CBOW tends to consume Low memory. We used 400,000 of senetence collected from social media and Wikipedia to create Afan Oromo pre-trained data .in which, the total number of used vocabularies is 360,000.

Word Embedding	Embedding Size	min_count	Window	Iteration	Sg
CBOW	200	1	5	10	0

Table 5: Parameter setting

The word vector used in this experiment was embedded using CBOW in Word2vec, and the embedding size was 200. The window size of was set to 5, min count is1 and the iteration was set to 10 other parameter settings are given in Table 4.

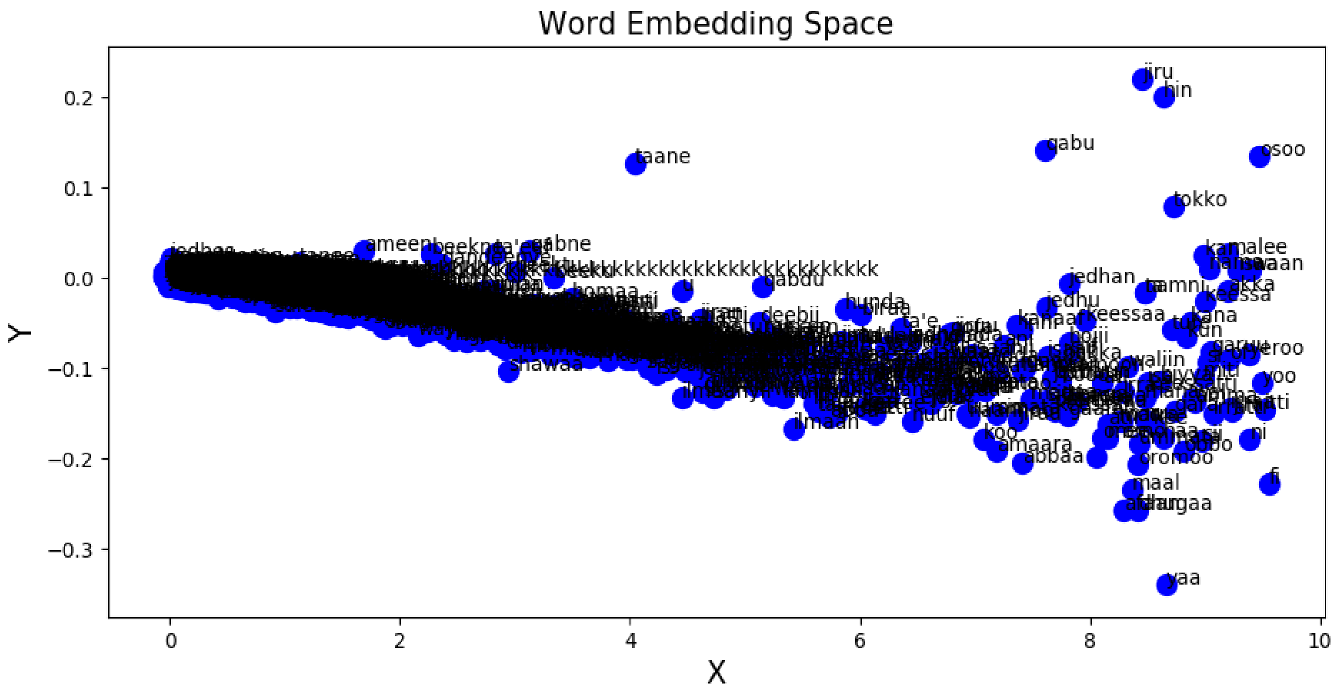


Figure 11: Visualization of the Word2Vec model trained

One of the techniques for checking model quality is to check if it reports high level of similarity between two semantically (or syntactically) equivalent words. Semantically similar words are expected to be near each other within our vector space. The gensim model.similarity() method is checking this type of proximity and returns a real number from 0 to 1 that measures the amount of proximity

```
In [50]: similar_words = word_model.most_similar(positive=['oromiyaa'], topn=3)
         print(similar_words,'\n')

         [('magaalaa', 0.9998647570610046), ('naannoo', 0.9998404383659363), ('finfinnee', 0.9998348951339722)]
```

Figure 12: show Sample snippet Word Similarity

5.3.2 Deep learning models

In this work, we report on three deep learning models for the classification task as strong hate, weak hate and neutral tweet. The three models were developed independently and show similar performance and we used the same data set for each model. The first model on this work is convolutional neural network. Convolutional Neural Networks are basically just numerous layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. The first layer in convolutional neural network is a word embedding layer, which maps each text tweet into a real vector domain. To do so, we map each Afaan Oromo word onto a 200-dimensional real valued vector, where each element is the weight for that dimension for that Afaan Oromo word. The embedding layer passes an input feature space with a shape of 200 × 100 to 1D convolutional layer with 128 filters with a window size of 5, padding the input such that the output has the similar length as the original input. The relu function is used for activation. This convolves the input feature space into a 196×128 representation, which is then more down-sampled by a 1D max pooling layer by producing an output of shape 1× 128. Finally, a dense layer with a softmax activation function takes this vector as input to predict probability distribution over all possible class, which will depend on individual datasets. We use the sparse_categorical_crossentropy loss function and the Adam optimizer to train the model on a batch size of 32 and 20 epochs.

Parameters	Value
Word embedding dimensions	200
No of filter size	128
Batch size	32
No of epochs	20
Learning rate	0.001
Dropout	0.25
Pooling size	4

Table 6: Hyper parameters for CNN training

Next, we have trained the created model with the fit () of the keras python library by calling the function. The model trained with the parameters such as the batches of the training data, number of epochs, steps per epoch, batch of validation data, validation steps, callbacks and verbose True. Lastly, we have evaluated the model with evaluate () keras python library and we have scored an accuracy above 98.15% for convolutional neural network on the test data.

The second model on this work is Bidirectional long short-term memory. The input of this model is a preprocessed Afaan Oromo data that is treated as a sequence of words. Like in convolutional neural network model, we map each Afaan Oromo word onto a 200-dimensional real valued vector, where each element is the weight for that dimension for that Afaan Oromo word. Then, the embedding layer passes an input feature space with a shape of 200 × 64 to Bidirectional LSTM layer. Bidirectional LSTMs are maintained in Keras through some Bidirectional layer wrapper. This wrapper uses a recurrent layer being an argument. Then we applied drop-out with a dropout rate of 0.25 and recurrent dropout of 0.1, the purpose of which is to regularize learning to avoid overfitting. Then, Bidirectional long short-term memory wrapper layer produces an output of in the shape 200x200. After that a global max pooling layer follows to ‘flatten’ the output space by taking the highest value in each timestep dimensions and produces the output of 1x200 shapes. Next, the first dense layer with a relu activation function with dropout rate of 0.25 and the second dense layer with activation function softmax takes this vector as input to predict probability distribution over all possible class. Finally, we use the sparse_categorical_crossentropy loss function and the Adam optimizer to train the Bidirectional long short-term memory model on a batch size of 32 and 20 epochs. Then, we have trained and evaluated the model with evaluate () keras python library and we have scored an accuracy above 97.98% on the test data.

Parameters	Value
Word embedding dimensions	200
Batch size	32
No of epochs	20
Learning rate	0.001
Dropout	0.25

Table 7: Hyper parameters for BLSTM training

The third model is a **hybrid** of convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) model. Like in convolutional neural network and Bidirectional long short-term memory model, the first layer is a word embedding layer, which maps each Afaan Oromo text tweet into a vector. After that, we map each Afaan Oromo word onto a 200-dimensional real valued vector, where each element is the weight for that dimension for that word. The embedding layer passes an input feature space with a shape of 200 × 64 to a drop-out layer with a rate of 0.25, the drive of which is to regularize learning to escape overfitting. Automatically, this can be supposed of as arbitrarily removing a word in sentences and forcing the classification not to depend on any individual words. The output feeds into a 1D convolutional layer with 64 filters with a Kernel size of 5, stride 1 and padding the input such that the output is in shape of 196 x 64. The rectified linear unit (Relu) function is used for activation. This convolves the input feature space into a 196 x 64 representation, which is then further down-sampled by a 1D max pooling layer with a pool size of 4 along the Afaan Oromo word dimension. A global max pooling layer follows to smooth the output space by taking the highest value in each timestep dimension, producing a 49× 64 vector. Each of the 49 dimensions can be considered as an ‘extracted feature’. These then feed into the bidirectional long short-term memory wrapper layer, which treats the extracted feature dimension as timesteps and yields 1 x 140 outputs to the next layer. Finally, a softmax layer takes this vector as input to guess probability distribution over all conceivable class, which will depend on individual datasets We use the sparse_categorical_crossentropy loss function and the Adam optimizer to train the model on a batch size of 64 and 20 epochs. After that, we have trained and evaluated the hybrid model, and we have scored accuracy above 96.91% on the test data.

Parameters	Value
Word embedding dimensions	200
No of filter size	64
Batch size	64
No of epochs	20
Learning rate	0.001
Dropout	0.25
Pooling size	4

Table 8: Hyper parameters for CNN_BLSTM training

5.4 Performance Evaluation

Once we fit a deep learning model, we must evaluate its performance on a test dataset. The Keras deep learning API model is very inadequate in terms of the metrics that you can use to report the model performance. To evaluate our system we use the Precision, Recall, F1score and accuracy classification which is measured automatically. During the training of the model the loss, accuracy, validation loss and validation accuracy is calculated automatically at each epoch (repetition).

5.4.1 Accuracy

The accuracy is one way to measure how often the deep learning model classifies a data point correctly. Accuracy can be calculated as the overall correctness of the model that means, the decisions that model got correct divided by all the total number of decisions made by the model and calculated by the following prescription.

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

True positive and true negative are the amount of correctly predicted tweet. False positive and false negative are the number of tweets of which the hate class was predicted incorrectly. The accuracy on the training and validation data is showed in the following figure this shows us how the tweet is correctly classified by model.

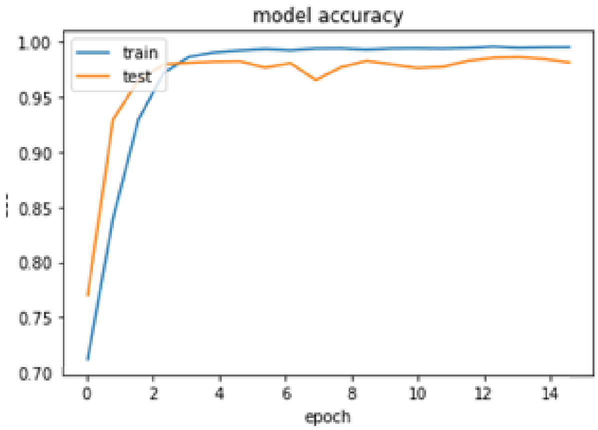


Figure 13: CNN model accuracy

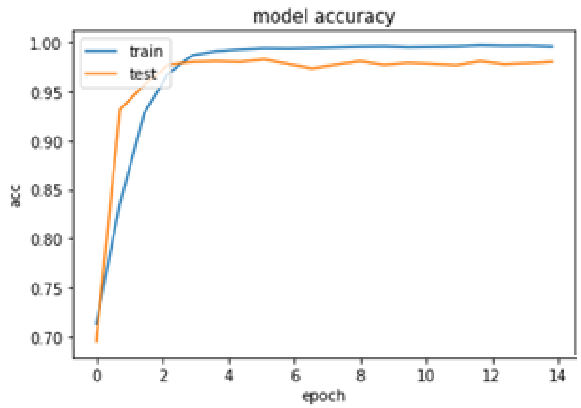


Figure 14: BLSTM model accuracy

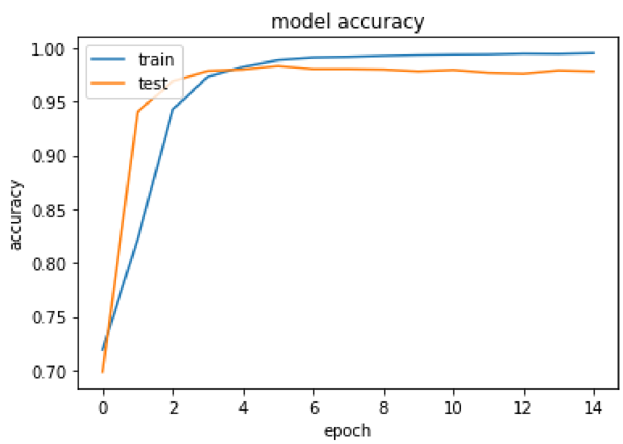


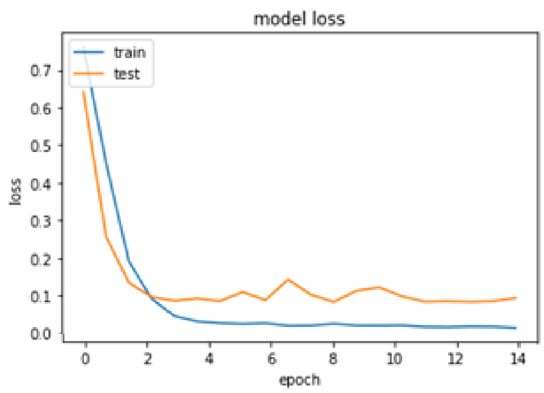
Figure 15: CNN-BLSTM model accuracy

The overall accuracy we scored is above 95.95%. This is incredible compared with the accuracy of the related works on hate speech detection prototypes which are developed by different researchers for different languages. Though, we compared only the accuracy score, nothing else as the language have their own syntaxes and semantics.

5.4.2 Loss

Loss functions are used to determine the error ("the loss") between the output of our model and the given target value. It is used in optimization problems with the goal of minimizing the loss. The loss on the training and validating data are showed in the following figure. We use the sparse categorical cross entropy loss function to calculate the loss value. Sparse categorical crossentropy is an integer-based kind of the categorical crossentropy loss function, which means that we don't have to convert the targets into categorical format anymore [55].

Figure 16: Convolutional neural network model



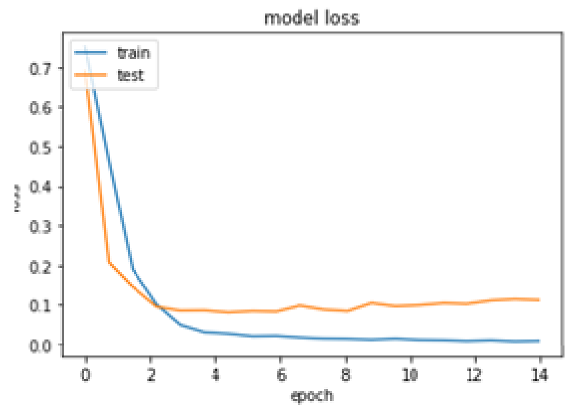


Figure 17: Bidirectional long short-term memory model

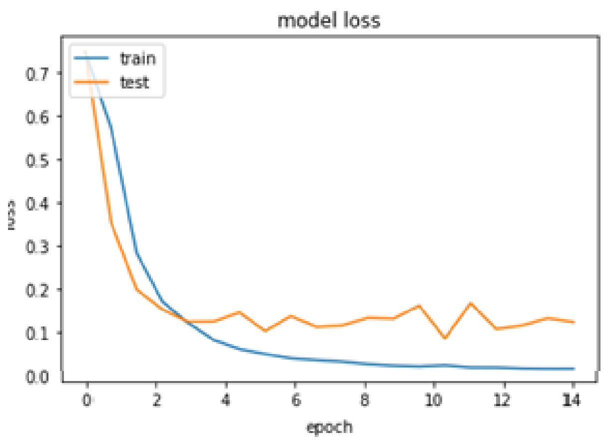


Figure 18: CNN-BLSTM model

5.4.3 Precision

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly).

$$\text{Precision} = \frac{TP}{TP+FP}$$

5.4.4 Recall

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected.

$$\text{Recall} = \frac{TP}{TP+FN}$$

5.4.5 F1score

F1score is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric.

$$\text{F1 Score} = \frac{2^* \text{(Recall * Precision)}}{\text{(Recall + Precision)}}$$

5.5 Experimental Result

To do Experimental Result, we prepared 14,077 Afaan Oromo dataset with the help of linguistic and legal professionals to train and evaluate the prototype. The collected dataset was annotated into strong hate, weak hate and neutral class. The dataset is then splitted into 80% for training and 20 % for testing. Pre-trained word embedding's can capture syntactic and semantic regularities in language. With Pre-trained word embedding's, neural networks achieve good performance in natural language processing tasks. In this study, we experimented three different deep learning models with Pre-trained word embedding's and analyzed the effects for hate speech detection tasks. It can be observed from the experiments and the results, which are given in Table 9, that the models initialized with Pre-trained word embedding's gives better training results. In this work, we used Afan Oromo word2vec pre-trained data created by collecting Afan Oromo text from different social media and Wikipedia. Then this created word2vec pre-trained data, can be used to improve the performance of the models.

The maximum result of Convolutional neural network was (98.15% accuracy, 98% precision, 95% recall and 96.9% F1score), while Bidirectional long short-term memory and the hybrid approach achieved (97.98% accuracy, 93% precision, 97% recall and 94.8% F1score) and (96.91% accuracy 88.5% precision, 96.4% recall and 91.6% F1score) respectively. Convolutional neural network has many promising result than Bidirectional long short-term memory and the hybrid approach by using the same dataset which discards 20% of training data for test.

	Models	Precision	Recall	F1score	Accuracy
Scenario 2	BLSTM	93.0	97.0	94.8	97.98
	CNN-BLSTM	88.5	96.4	91.6	96.91
	CNN	**98.0**	**95.9**	**96.9**	**98.15**

Table 9: Scenario 2 Experimental Result with Word2Vec

5.6 Discussion

The hate speech detection for Afaan Oromo prototype enables automatic online detection of hate speech.as we described above in section 6.5 we prepared a dataset of 14,077 label data to train and test our model. The collected dataset was labeled into three class's strong hate, weak hate and neutral. We trained three different deep learning models those are convolutional neural network, bidirectional long short-term memory neural network and finally the hybrid of the two neural networks models. We used the same dataset for each deep learning model. This method needs large set of corpuses and a long period of time to train. If the corpus is very large it may take long time to train. Though, the benefit is once we trained the model we can use that to predict a new data which means no necessity to train again. Once more, the model is used to know which word less important and which word is more important. This means we don't expect to throw the stop words of the input Afaan Oromo text. Here also, we use word embedding methods to represent the similarity of words. Word embedding gives similar value for semantically similar words. Hence, this technique aided us to build a low dimensional model that can preserve previous states. And yet again, no need of synonym wordlist preparation. This reduces our effort. When the model trained in large dataset it can have a good accuracy on predicting hate speech. However, still there are no prepared Afaan Oromo dataset for hate speech detection.so we prepared dataset, much as we can with the help of Afaan Oromo language and legal professional to train and test our model. Therefore, the accuracy we scored is good in the prediction even if our dataset is small. The other one is, we trained the three models without data preprocessing and got 64, 62.1 and 63.2 training accuracy and 67.2, 65.4 and 64.7, test accuracy for Convolutional neural network, Bidirectional long short-term memory and CNN-BLSTM respectively, which is small result in both training and testing case. Then after we have done some data prepressing methods to assure quality data in order to increase the accuracy of our model. Then after that we conclude that, using pre-trained data and applying data preprocessing can be answer for the first research questions ("*How and when does the performance of the proposed hate speech detection system increase?*"). The best solution for improving the performance of the model is data preprocessing. Preprocessing the data is very important, in order to realize satisfying results on the hate speech detection corpus. In our work, we have applied various techniques to enhance the quality of our data. We have done tokenization, removing numbers, removing HTML Tags, removing punctuation from words and removing all words that are not contained of alphabetical characters. And also, to answer the second research Questions ("*From CNN, BLSTM and CNN-BLSTM which model is a better for Afaan Oromo hate speech detection?"),* CNN deep learning method fit hate speech detection system for Afaan Oromo.

CHAPTER SIX

CONCLUSION, CONTRIBUTION AND RECOMMENDATION

6.1. Conclusion

Now days, there are a number of hate speech detection system that are developed in different language including English, Arabic, Italian, German, Amharic, Vietnamese and other language. Why not for Afaan Oromo language? Because there is no generalized work since hate speech is language dependent. While every language has a different set of rules all language obeys rules. All language has underlying structure rules that make meaning full communication possible. Though for the local languages or under resourced languages like Afaan Oromo language are not sufficiently implemented hate Speech detection.

To accomplish this work, we reviewed a literature in detail to understand the methods and techniques that used for developing hate speech detection. We prepared the Afaan Oromo hate speech dataset. The dataset contains 14,077 Afaan Oromo posts and comments out of this 11261 for training and 2816 for testing after passing different steps as stated in the experiment section. And then we designed and developed architecture for Afaan Oromo hate speech detection we created and trained a Convolutional neural network, Bidirectional long short-term memory and the hybrid model for the Afaan Oromo hate speech detection. Finally, we evaluated the developed prototype.

The prototype accepts a labeled Afaan Oromo hate speech dataset, which contains three class's strong hate, weak hate and neutral class. And then, the dataset is splitting into two. Specifically, training and testing data. 80% of the dataset is used for training and 20% for testing the three models. Then, the model trained with 80% of the data and tested on the 20% of the dataset. For testing dataset, the model predicts the class for Afaan Oromo input tweet. Additionally, 200-dimensional pre-trained word embedding was obtained by applying a CBOW word2vec model to a larger corpus. The effects of the pre-trained word embedding on deep learning models are discussed. The results show that, with pre-trained word embedding, all models performed better than the models without pre-trained word embedding.

The maximum results achieved by Convolutional neural network, Bidirectional long short-term memory and the hybrid methods are 98.15%, 97.98% and 96.91% accuracy respectively. This research indicated that Convolutional neural network model is more applicable to Afaan Oromo hate speech detection than Convolutional neural network, Bidirectional long short-term memory and the hybrid model. This experimental result shows that the performance of developed hate speech detection for Afaan Oromo is significantly good for first time of Afaan Oromo text.

6.2. Challenge of Hate Speech Detection for Afaan Oromo

During investigating Hate Speech Detection for Afaan Oromo, many challenges were encountered by the researchers. The primary challenge was preparing Hate Speech dataset of Afaan Oromo. Afaan Oromo has not had as such well resources that could have helped for exploring. For this research, we prepared Dataset manual from Facebook and Twitter. When we collect dataset of Afaan Oromo from different source, almost, the available text contains a different spelling mistake. In Afaan Oromo, if one character is added or reduced it can totally change the meaning of that word. This impression is challenging and to avoid this problem we corrected the spelling of some misspelled words manually for the reason that there is no Available Afaan Oromo spell correcting API to use.

6.3. Contribution of the study

- 14,077 label datasets were prepared and we make this dataset freely available online, in order to be used for further study by other researchers.
- Hate Speech Detection model was investigated with Convolutional neural network, Bidirectional long short-term memory and the hybrid of Convolutional neural network and Bidirectional long short-term memory for Afaan Oromo language.
- The Convolutional neural network approach result shows better accuracy for Afaan Oromo Hate Speech Detection than Bidirectional long short-term memory and hybrid approaches.

6.4. Recommendation

The study has shown that Afaan Oromo hate speech detection can be done automatically using Convolutional neural network and Recurrent Neural Network algorithm. Listed below are some of recommendations and future works:

- With regard to dataset, Afaan Oromo language didn't have dataset which is labeled categories. For this study, some amount of corpus is prepared manually but this corpus is insufficient and very small. So, in the future huge amount of dataset for Afaan Oromo hate speech has to be prepared and the system has to be trained on that to improve its performance even if the accuracy of our models is promising.
- Examining the different aspect of the class of hate, either hate speech with politics, ethnicity, religion, gender and socio-economy.

- The source of our dataset is mainly from Facebook and tweeter including other sources to improve the feature space for such under resourced language for computational purpose by adding alternative from other sources such as forums and from other social media is important.
- Our work does not include Afaan Oromo spelling stemmer, checker, tagger, parser and etc. and then, if it includes Afaan Oromo stemmer, spelling checker, tagger, parser, stemmer and etc. the model can be improved to understanding Afaan Oromo language.
- We endorse researchers to use different types of word embedding models such as Glove and FastText.
- Repeating this work for others Ethiopian languages and developing multilingual hate speech detection system.

REFERENCE

[1] SAS, Natural Language Processing (NLP) "what *is Natural Language Processing"* [Online] Available: https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html[Accessed: June, 28, 2020].

[2] "Hate Speech and Disinformation Prevention and Suppression Proclamation No/2019."

[3] YouTube Community Guidelines [Online]. Available: https://www.youtube.com/yt/policyandsafety/communityguidelines.html

[4] The Twitter Rules [Online]. Available: https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

[5] Facebook Comment Policy [Online]. Available: https://www.facebook.com/help

[6] A. Sucia, M. Nasrun, and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network, Computer Engineering", [Online serial]. Available: https://www.researchgate.net/publication/333072313_Analysis_Text_of_Hate_Speech_Detection_Using_Recurrent_Neural_Network, [Accessed: June, 28, 2020].

[7] Felix Horne,"Tackling Hate Speech in Ethiopia", [Online],Available: https://www.hrw.org/news/2018/12/03/tackling-hate-speech-ethiopia.[Accessed: June, 28, 2020].

[8] Abebe Keno (2002) Case Systems in Oromo, MA Thesis. Ababa University, Ethiopia.

[9] Girma Debele (2014) Afaan Oromo News Text Summarizer, Master's thesis, Pohang University of Science and Technology, Pohang, Korea.

[10] Daniel Bekele(2011) Afaan Oromo Information Retrieval (CLIR): A Corpus Based Approach, M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

[11] Z. Mossie and J. Wang, ''social network hate speech Detection for Amharic language'', pp. 41–55, 2018.

[12] Q. Pham Huu, S. Nguyen Trung and H. Anh Pham, "Automated Hate Speech Detection on Vietnamese Social Networks", 2019.

[13] H. Faris, I. Aljarah, M. Habib and P. Castillo, "Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context", ETSIIT - CITIC, University of Granada, Spain.

[14] G. Bianchini, L. Ferri and T. Giorni,"Text analysis for hate speech detection in Italian messages on Twitter and Facebook",CEUR-ws.org/vol-2263 ,paper043.

[15] N.D.T. Ruwandika1 and A.R. Weerasinghe, "Identification of Hate Speech in Social Media "International Conference on Advances in ICT for Emerging Regions (ICTer): 273 – 278, 2018.

[16] S. Malmasi and M. Zampieri," Detecting Hate Speech in Social Media", Proceedings of Recent Advances in Natural Language Processing, pages 467–472, 2017.

[17] Z. Zhang, D. Robinson, and J. Tepper. 2018. Hate Speech Detection Using a Convolution-LSTM Based Deep Neural Network. In Proceedings of ACM the Web conference (WWW'2018). ACM, New York, NY, USA, Article 4, 10 pages. [Online], Available: https://doi.org/10.475/123_4, [Accessed: June, 28, 2020].

[18] Beekan G Erena, “Oromo Language (Afaan Oromoo)” Available: https://scholar.harvard.edu/erena/oromo-language-afaan-oromoo, [Accessed: August 15, 2020].

[19] "Afaan Oromo". University of Pennsylvania, School of African Studies, Available: https://en.wikipedia.org/wiki/University_of_Pennsylvania, [Accessed: August 15, 2020].

[20] R. Geetaachoo, FURTUU: Seerluga Afaan Oromoo(Oromo Grammar). 2009.

[21] B. Addunyaa, SEMMOO: Bu’uura Barnoota Afaaniifi Afoola Oromoo. 2014.

[22] W. Tesema and D. Tamirat, “Investigating Afaan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods”.

[23] D. Ince, “Acoustic coupler,” in *A Dictionary of the Internet*. Oxford University

[24] Wikibooks, Wikibooks,. .

[25] C. Mellish and G. Ritchie, ―The Grammatical Analysis of Sentences,‖ pp. 1–16.

[26] J. chen, “Neural Network”, Available: https://www.investopedia.com/terms/n/neuralnetwork.asp#:~:text=A%20neural%20network%20is%20a,organic%20or%20artificial%20in%20nature [Accessed: August 15, 2020].

[27] “What is Tokenization in NLP?, Available: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/ , [Accessed: August 15, 2020].

[28] “What is Word Embedding”, Available: https://www.mygreatlearning.com/blog/word-embedding/ , [Accessed: August 15, 2020].

[29] “How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras”, Available:https://machinelearningmastery.com/develop-bidirectional-lstm-sequenceclassificationpythonkeras/#:~:text=Bidirectional%20LSTMs%20are%20an%20extension,LSTMs%20on%20the%20input%20sequence, [Accessed: August 15, 2020]

[30] J. Brownlee, “Dropout Regularization in Deep Learning Models With Keras”, Available: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/ , [Accessed: September 3, 2020].

[31] H. Heidenreich, “Understanding Kera Dense Layers, Available: https://medium.com/@hunterheidenreich/understanding-keras-dense-layers-2abadff9b990, [Accessed: September 3, 2020].

[32] Hatebase; Available from: https://hatebase.org/. [Accessed: September 3, 2020].

[33] “Supervised Machine Learning”, Available: https://www.datarobot.com/wiki/supervised-machine-learning/, [Accessed: September 5, 2020].

[34] “Unsupervised Machine Learning: What is, Algorithms, Example”, Available: https://www.guru99.com/unsupervised-machine-learning.html, [Accessed: September 5, 2020].

[35] M. Rouse, “Supervised learning”, Available:https://searchenterpriseai.techtarget.com/definition/supervised-learning, [Accessed: September 5, 2020].

[36] K. Ming Leung, “Naive Bayesian Classifier”, November 28, 2007.

[37] J. Teevan, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers", Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.
[38] "Text Classification Using Support Vector Machines (SVM)", Available https://monkeylearn.com/text-classification-support-vector-machines svm/#:~:text=From%20Texts%20to%20Vectors,encode%20any%20kind%20of%20data, [Accessed: September 4, 2020].
[39] "Using Convolutional Neural Networks for Sentence Classification", Available: https://missinglink.ai/guides/convolutional-neural-networks/using-convolutional-neural-networks-sentence-classification/ ,[Accessed: September 5, 2020].
[40] D. Britz, "Understanding Convolutional Neural Networks for NLP", Available: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/, [Accessed: September 5, 2020].
[41] "Deep Dive into Bidirectional LSTM", Available: https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/ , [Accessed: September 5, 2020].
[42] Denny Britz. Understanding convolutional neural networks for nlp. Available: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. [Accessed: October 16, 2020].
[43] Sh. Biere, Hate Speech Detection Using Natural Language Processing Techniques, 2018, [Accessed: October 16, 2020].
[44] Peiqiu Chen Dingjun Yu, HanliWang and ZhihuaWei. Mixed pooling for convolutional neural networks. 2014, [Accessed: September 5, 2020].
[45] Marc Moreno Lopez and Jugal Kalita. Deep learning for nlp. 2017, [Accessed: September 5, 2020].
[46] Packt Publishing. Sentence classification using cnns. Available: https://www.datasciencecentral.com/profiles/blogs/sentence-classification-using-cnns [Accessed: September 5, 2020].
[47] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
[48] J. Brownlee, "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning", Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ ,[Accessed: September 5, 2020].
[49] Binary crossentropy, Available: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy ,[Accessed: September 5, 2020].
[50] A. Choudhury, TensorFlow vs Keras: Which One Should You Choose, Available: https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/ ,[Accessed: September 5, 2020].
[51] "Binary crossentropy", Available: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy , [Accessed: September 17, 2020].
[52] J. Chen, What is a Neural Network?, Available: https://www.investopedia.com/terms/n/neuralnetwork.asp, [Accessed: October 17, 2020]

[53] "Anaconda and Jupyter Notebook Setup", Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119602927.app1#:~:text=Anaconda%20is%20a%20Python%20distribution,commonly%20used%20for%20data%20science.&text=Anaconda%20Navigator%20is%20a%20GUI,tools%20such%20as%20Jupyter%20Notebook. , [Accessed: October 27, 2020]
[54] "Hate speech and violence" , Available: https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/hate-speech-and-violence, [Accessed: October 27, 2020]
[55] How to use sparse categorical crossentropy in Keras?, Available: https://www.machinecurve.com/index.php/2019/10/06/how-to-use-sparse-categorical-crossentropy-in-keras/#sparse-categorical-crossentropy, [Accessed: October 27, 2020]
[56] J. Brownlee,What Are Word Embeddings for Text?, Available: https://machinelearningmastery.com/what-are-word-embeddings/,[Accessed: October 27, 2020]
[57] "Convolutional neural network",Available: https://en.wikipedia.org/wiki/Convolutional_neural_network//,[Accessed: October 27, 2020]
[58] J. Brownlee,"Best Practices for Text Classification with Deep Learning",Available: https://machinelearningmastery.com/best-practices-document-classification-deep-learning/,[Accessed: October 27, 2020]
[59] The brute force layer of a Machine Learning model, Available: https://iq.opengenus.org/fully-connected layer/#:~:text=Fully%20Connected%20layers%20in%20a,to%20form%20the%20final%20output. [Accessed: March 15, 2021].
[60] S. Kumar, "Overview of various Optimizers in Neural Networks", Available: https://towardsdatascience.com/overview-of-various-optimizers-in-neural-networks-17c1be2df6d5. [Accessed: March 15, 2021].
[61] F. Shaikh, "Comparison between Deep Learning & Machine Learning." [Accessed: March 15, 2021].
[62] S. Ambati," Deep learning: A brief guide for practical problem solvers", Available: https://www.infoworld.com/article/3003315/deep-learning-a-brief-guide-for-practical-problem-solvers.html. [Accessed: March 15, 2021].
[63] Nockleby JT. Hate Speech. Encyclopedia of the American Constitution. 2000; 3:1277–79. [Accessed: March 15, 2021].
[64] European commotion, "Code of Conduct–Illegal online hate speech Questions and answers",Available: https://ec.europa.eu/info/sites/info/files/code_of_conduct_hate_speech_en.pdf [Accessed: March 15, 2021].

APPENDICES

Appendix 1: Sample dataset

Strong hate	amaara Doofaan,kunoo kana kaa gorsi keessan.Afaan oromoof hangammi ilaalcha gadi aanaa akka qabdan agarsiisa.
Strong hate	warra akka kee kana warraa amaara sammu Furri kan dhigaa Oromoo dhugu.
Neutral	Sabnii Oromoo Abdii biyya Ethiopia tii kanaf akka Sabaa tii amba irratti wal xiiqeesu irraa mana kessatti wal gorfachu adda yoo godhane qofa tabba bahun danda'ama dhiima ummamu hunda yerro o dubbanu,yerro barresiinu fi ibsaa keniinu kallattii biilchiina?
Weak Hate	Luqatu qiga ilman oromoo akkanumati dhuga ilman hadha jala dhufte salama
Strong hate	gambeella jechuun osoo sammuu kee bananii ilaalanii Buqqee tortoraa raqa ajaa'aa xiraawaa udaan keessa guutee fooliin isaa reeffa 7 buleeru caalaa kan ajaa'udha.
Strong hate	tigree jechuun udaan raammoo keessaadha,Haati kee siin udaan waliin si hagde malee si hin dhalle
Neutral	Fincila Dargaggoota Tigraay jettaayi? Kan Abiy wanbadoota qibdeessee achitti hokkora kaase!
Strong hate	amaara ilma haadha salee. niftam
Neutral	Rabii beekaa dhigaa ilmaanaa orommo kan dhumaa jiran
Neutral	Media Nubia kana maaliif gara taayyee PLC tti hin jijjirtu
Neutral	Sirrii jette namni hoomacha (Arafaa) ykn Abiy qabate nyaatamuun isaa hin hafu yeroodhumaaf wixxifata malee.
Neutral	sitti fakkaate 4 kiloo keessaa Dr.Abbiyin gaggeesse malee dijitaalonni wayyannee warri waggaa 27 jara wajjiin turan isinuma cina jiru osoo akka caarraa wayyanneen kaate ishuma wajjiinin harka reebuu sis reebuu.
Strong hate	woyyaanee qofa ilaaluu osoo hin taane ishii eenyutu kuffise jedhii yaadi namni akkamitti waan ofii itti yaadee kuffiseetti hirkata amaara daalacha dhama'aa woyii
Weak Hate	Kunis kufun isaa hin hafu seena isa gaafa inni kufu nu gaafadhu yaa dhuufu
Neutral	Nagaan bultaniin sila afaan keetii hin ba'u ganamaan kaatee isa abjootaa bulte hagda waa'ee du'aa akkamitt akka dhala namaa haqanqaaltee ajjeeftu.
Strong hate	Osoo Angoof bochuu yoo dutee hoo amaara Attii dhugaa qabdaa badaa jaldeesaa mukaa keessaa Oromoon wayyaanee jalaa issiin basee jenee
Strong hate	amaara Fokkataa akka kee irra waan gaariin hin eegamu woyyaanen kufte tigraay situ bulchaa jiraare bada goorillaa
Neutral	kan fulduraa jiru hoomocha;homa gochu akka hin dandeenye fi ashkaroota akka tataan Yohannis bohayalehun dubbateera.

Appendix 2: Summary of model for Convolutional Neural Network

```
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_13 (Embedding)     (None, 200, 100)          500000
_________________________________________________________________
conv1d_13 (Conv1D)           (None, 196, 128)          64128
_________________________________________________________________
global_max_pooling1d_13 (Glo (None, 128)               0
_________________________________________________________________
dense_13 (Dense)             (None, 4)                 516
=================================================================
Total params: 564,644
Trainable params: 564,644
Non-trainable params: 0
_________________________________________________________________
```

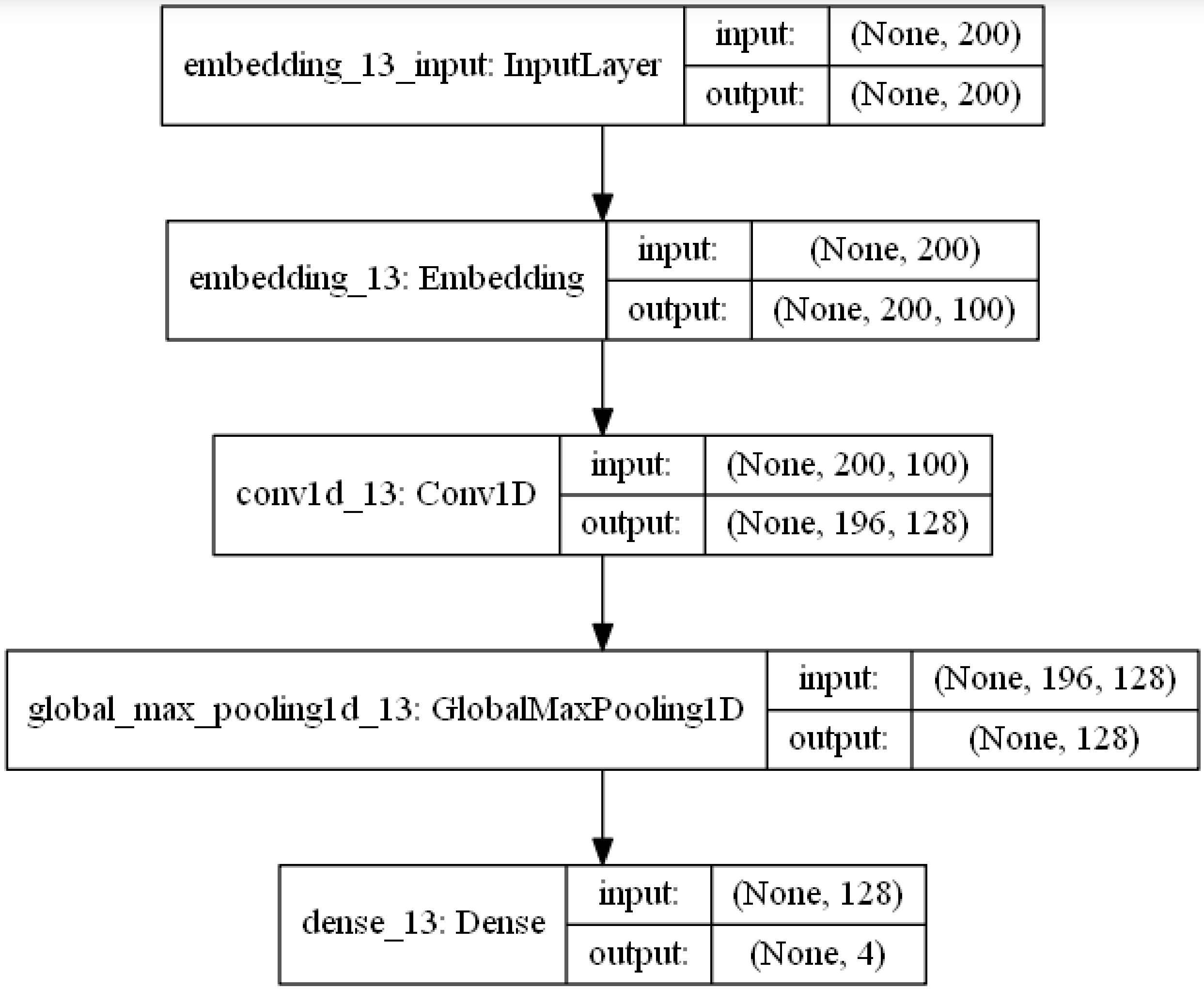
Appendix 3: Summary of model for BLSTM Neural Network

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 200)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 200, 64)           320000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 200, 200)          132000
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 200)               0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               20100
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 404
=================================================================
Total params: 472,504
Trainable params: 472,504
Non-trainable params: 0
_________________________________________________________________
```

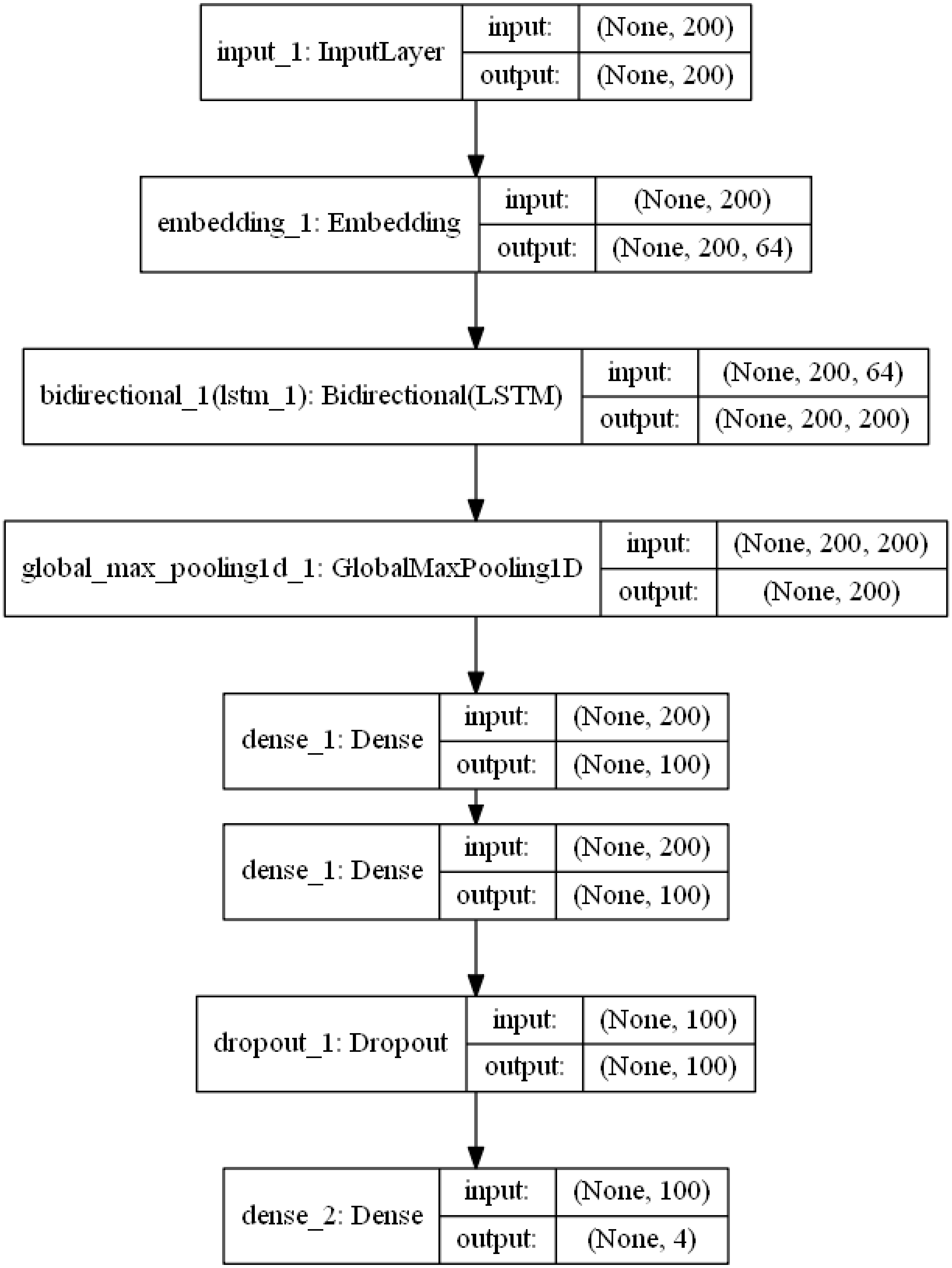
Appendix 4: Summary of model for CNN-BLSTM Neural Network

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_9 (Embedding)      (None, 200, 64)           320000
_________________________________________________________________
dropout_9 (Dropout)          (None, 200, 64)           0
_________________________________________________________________
conv1d_9 (Conv1D)            (None, 196, 64)           20544
_________________________________________________________________
max_pooling1d_9 (MaxPooling1 (None, 49, 64)            0
_________________________________________________________________
bidirectional_9 (Bidirection (None, 140)               75600
_________________________________________________________________
dense_9 (Dense)              (None, 4)                 564
_________________________________________________________________
activation_9 (Activation)    (None, 4)                 0
=================================================================
Total params: 416,708
Trainable params: 416,708
Non-trainable params: 0
_________________________________________________________________
```

Appendix 5: Convolutional Neural Network Model



Appendix 6: BLSTM Neural Network Model



Appendix 7: CNN-BLSTM Neural Network Model