

Emotion Detection for Afaan Oromo Using Deep Learning

Kabada Sori

Ambo University Institute of Technology Department of IT

Getachew Mamo(PhD)

Jima University School of Computing Department of IT

Abstract

Emotion detection in text has become more popular due to its various useful applications in a different area, such as tracking product perception, public opinion detection about political tendencies, stock market monitoring, text summarization, information extraction, recommendation system, question answering and etc. However, manually identifying emotion of a million of people and aggregating them towards a rapid and efficient decision is quite a challenging task due to the rapid growth of social media user. This study aimed to develop Afaan Oromo emotion detection model in order to tackle this challenge. This study adopts artificial neural network approach. We used python tools with Keras library. We conduct our experiments on five emotion class (anger(arii), love(jaalala), joy(gamachu), disgust(jibba), and sadness(gadda)) by collecting a total of 1005 emotional sentence of Afaan Oromo language that have been manually annotated. The sentence has been scraped from different official Facebook page such as Oromia Broadcasting Network (OBN) pages, Fana Broadcasting Corporation (FBC) Afaan Oromo page, and British Broadcasting Corporation (BBC) Afaan Oromo pages using Facepuger tools by creating Facebook API id. After collecting these data all preprocessing steps like tokenization, stop word removal and normalization have been undertaken. We used word embedding's for feature extraction of preprocessed data. Subsequently, we have applied three artificial neural network algorithms such as Feed forward neural network, long short-term Memory and Bidirectional long short-term memory for classification purpose of the vectorized sentence into their emotion class. We compared the three artificial neural network algorithms and found out that Bidirectional long short-term memory achieved the best performance. We have achieved an average accuracy of 66%, 78%, 83% using Feed Forward Neural Network, Long Short-Term Memory and Bidirectional Long Short-Term Memory respectively. Based on experimental result, the researcher concluded that increasing amount of dataset, tuning hyper parameters properly and trying by different algorithms can, in some case, improve the performance of the model.

Keywords: Emotion Identification, Afaan Oromo, Artificial Neural Network, Social Media

DOI: 10.7176/NMMC/92-01

Publication date: August 31st 2020

1. Introduction

Language is one of the methods used for human communication through either spoken or written, consisting of the use of words in a structured and conventional way. According to (Kaur & Saini, 2014) language is used not only for communication but also for imparting emotion associated with it. People express their emotion in different way, written text is one of a good source for expressing your views, ideas, emotions, and filling.

Emotion detection is the task of automatically identifying and extracting the emotions (such as happiness, sadness, fear, anger, surprise, and disgust)" expressed in a given text. Unlike Sentiment analysis (Gaiind et.al, 2019) which perform only whether the public reaction is positive, negative or neutral but fails to describe the exact feeling of the customers and the intensity of their reaction, emotion detection performs whether the public reaction is happiness, sadness, fear, anger, surprise, and disgust by making deeper study of document meaning.

Even though today the internet has become a key medium through which people express their emotions, feeling, and opinions. Every event, news around the world, is shared, discussed, posted and commented on social media, by millions of people. Capturing emotion, filling and opinion of people those commented on social media can be a source of valuable information, which can be used to study how different people react to different situation and events. But, manually recognizing the emotion of people from a huge amount of data is a major challenging task for the human being. According to (Miner et.al, 2016), the challenge of identifying emotion is further compounded by the difficulty in understanding context, natural language ambiguity and due to rapidly growing of the internet. However, with the growing prominence of messaging platforms like Facebook, Twitter, telegram, and WhatsApp as well as digital agents, it is essential that the machine is able to understand emotion in textual conversation.

Natural Language Processing (NLP) is a field of computer science, artificial intelligence (also called machine learning), and linguistics which is concerned with the interaction of computers and human beings, has been practiced in order to automatically identify the information contained in the text.

So, in this study, we used Artificial Neural Networks (ANNs) approach for identifying emotions in the textual

comment.

Artificial Neural Networks (ANNs) are software implementations of the neuronal structure of our brains. It attempts to simplify and mimic the behavior of our brain (Thomas, 2017). It is often used for analysis and modeling of data statistically, in which their role is supposed as an alternative to model nonlinear analysis techniques (Cheng & Titterington, 1994).

They can be trained in a supervised or unsupervised manner. The supervised Artificial Neural Network can be trained by providing corresponding input and output data samples, with the intention of getting the Artificial Neural Network to provide the desired output for a given input. It is typically used in solving many problems that may be couched in terms of predicting, or classification.

On another hand, unsupervised learning is another way of an Artificial Neural Network that can train in which an Artificial Neural Network able to comprehend the structure of the provided input data on its own (Thomas, 2017).

In general, an Artificial Neural Network has been applied in the domain of artificial intelligence problems like classification, clustering, regression medical diagnosis, financial market prediction, recognition of textual character, image and speech and etc. Due to this reason, many computer engineers and scientists understand the artificial neural network as parallel distributed computing. By this means, the Artificial Neural Network provides an alternative to the conventional algorithmic techniques that have dominated in machine intelligence (Gurney et.al, 1997).

2. Related Work

In this section we presented the revision of a number of emotion detection works by focusing on three languages: Indian, Indonesia and Arabic. While reviewing, we focused on the algorithm, emotion type, dataset used and performance of those works. Emotion detection is a new concept for Afaan Oromo and the language is resource scarce. Taking this problem into account and considering what we obtained from the revision of the related works, we propose a new solution to Afaan Oromo emotion detection problem. The proposed solution will be based on an Artificial Neural Network approach. The overall summary of each emotion detection work is presented in Table 1.

We have also reviewed on sentiment analysis and opinion mining work by focusing on Afaan Oromo and Amharic language as shown in Table 2 below.

Table 1 summary of Emotion Detection related works

Author	Title	method	Number of datasets	Emotion type	Result
(Vijay et.al, 2018)	Corpus Creation and Emotion Prediction for Hindi-English Code-Mixed Social Media Text	SVM with an RBF kernel classifier	2866 tweets	happiness, sadness and anger	58.2%
(Grover & Verma, 2016)	Punjabi Text Emotion Detection	Hybrid concept of keyword-based Approach and machine learning approach such as Support Vector Machine and Naïve Bayes	HC Corpora	happiness, fear, anger, sadness, disgust, and surprise	_____
(Effendi et.al, 2015)	A two-stage supervised approach to detect emotions on Indonesian Tweets.	machine learning approaches, such as maximum Entropy and SVM.	2203 data among this only 42.80% are emotional tweets	love, joy, anger, fear and sad	Maximum Entropy gets accuracy of 71.96 % and SVM gets 69.83%.
(Rabie & Sturm, 2014)	Emotion Detection in Arabic Social Media Content	SVM (known as SMO) and the Naïve Bayes and word-emotion lexicon	1776 datasets from Twitter, social media websites	_____	22.27% for word-emotion lexicon
(Abdullah et.al, 2018)	SEDAT: Sentiment and Emotion Detection in Arabic Text using CNN-LSTM Deep Learning.	CNN-LSTM Deep Learning.	SemEval-2018 Task1	Anger, Fear, Joy's, sadness	_____

Table 2 Summary of Afaan Oromo and Amharic Sentiment Analysis Related Work

Author	Title	Method	Domain,(data source)	Class	Number of dataset
(Tariku, 2017)	Sentiment Mining and Aspect Based Summarization of Opinionated Afaan Oromoo News Text	Rule based	ORTO news service	+ve and -ve	400 reviews
(Abate, 2019)	Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments	Unsupervised	OPDO official Facebook page, political bloggers page	+ve, -ve & Neutral	600 reviews
(Oljira, 2020)	Sentiment Analysis Of Afaan Oromo social media sentiment analysis using Machine learning approach	MNB, LSTM, CNN	ODP official Facebook page	+ve & -ve	1452
(Gebremeskel, 2010)	Sentiment Mining Model for Opinionated Amharic Texts	Rule and lexicon based	Movie review and news doc	+ve,-ve and neutral	303
(Mengistu, 2013)	Sentiment Analysis for Amharic opinionated text	NB, MNM, SVM	ERTA.com, fanaBC.com & diretube.com	+ve, -ve and neutral	576
(Abreham, 2014)	Opinion Mining From Amharic Entertainment Texts	MN, DT and ME	EBC, diretube.com and habesha.com	+ve and -ve	616
(Tilahun, 2014)	Linguistic Localization of Opinion Mining from Amharic Blogs	Rule and lexicon based	Hotel, university and hospital	+ve, -ve and neutral	484
(Philemon & Mulugeta, 2014)	A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts	NB	Social media, product marketing and news	-2, -1, 0, +1, +2	608
(Getachew & Alemu, 2019)	Deep Learning approach for Amaharic Sentiment Analysis	Deep learning	FanaBC official Facebook page	+ve, very +ve, extremely +ve, neutra, -ve, very -ve, & extremely -ve	1600 review

3. Methodology

In this section we introduce the steps we followed in order to conduct Afaan Oromo Emotion detection.

3.1. Data Collection

The quality of input data into neural network models may strongly influence the results of the data analysis. So, preparing data properly is an important and critical step in neural network modeling for complex data analysis and it has an immense impact on the success of a wide variety of complex data analysis.

Therefore, we have collected data from different official Facebook page such as Oromia Broadcasting Network (OBN) pages, Fana Broadcasting Corporation (FBC) Afaan Oromo page, and British Broadcasting Corporation (BBC) Afaan Oromo pages. Those pages are public and people can express their idea freely on those pages.

We collected 1800 sentence and the collected sentence were further edited manually by removing those sentences which does not have emotion at all. The source with their emotion class and number of collected emotional sentence after removing non emotional sentences were shown in Table 3.

Table 3 Proportion of emotional data set with respect to their sources

Source	Emotion class	Number of sentences
OBN	anger(arii)	67
	love(jaalala)	70
	joy(gamachu)	76
	disgust(jibba)	45
	sadness(gadda)	50
	Total	308
FBC	anger(arii)	72
	love(jaalala)	71
	joy(gamachu)	63
	disgust(jibba)	80
	sadness(gadda)	51
	Total	337
BBC	anger(arii)	62
	love(jaalala)	60
	joy(gamachu)	62
	disgust(jibba)	76
	sadness(gadda)	100
	Total	360
Grand Total		1005

As can be seen from Table 3, about 360 of emotional sentence were collected from BBC, 337 of emotional sentence were collected from FBC and 308 of them was collected form OBN. Thoroughly 201 emotional sentences for each five emotion class and totally 1005 emotional sentence were collected for experimental purpose as shown Figure 1. Why we used the data from various source is to show the possibility of developing Emotion identifier from different environment for Afaan Oromo.



Figure 1 Afaan oromo emotion dataset

3.2. Data Preprocessing

Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Hence, preprocessing is the preliminary step which transforms the data into a format that will be more easily and effectively processed (Kuhn & Johnson, 2013). Preprocessing must ensure that the source text be presented to NLP in a form usable for it. In this study, text preprocessing is a primary step to make our data sets compatible with the machine learning tool that was used in our study called python. In the preprocessing stage this study addresses tokenization, stop word removal, and normalization. All the processing tasks that we have used in this study were implemented using python software.

3.2.1. Tokenization

Tokenization is a module which splits up the text into a set of tokens usually words, based on the boundaries of a written text. Tokenization takes the input texts that is supplied from a user and tokenizes it in to a sequence of tokens. Token is the smallest unit that will be extracted from the input sentence before performing emotion detection task. In this study tokenization is needed for some purposes that are to select contexts, stop word removal and for further steps. For instance, if we have the sentence: **Lamma baay'ee hedduun si jaaladha; 'Lamma I love**

you very much' in our dataset. We have tokenized into set of words on the white space, like **Lamma**, **'baay'ee**, **'heddun'**, **'si'**, **'jaaladha'**.

3.2.2. Stop Word Removal

All words which found in the document (or corpus) are not equally important for emotion detection purpose. Some words are common in most documents. Therefore, removing words which are not used to determine emotion is very important. Such words may be removed using two methods. The first method is to remove high frequent terms by counting the number of occurrences (frequency). The second method is using stop word list of the language. Therefore, in this study the second method were used to apply stop word removal. Afaan Oromo stop words include pronouns, prepositions, conjunctions, articles, and particles. Those words need to be removed during preprocessing phase. Words like kana/'this', sun/'that', fi/'and', 'inni/'he', ana/'me', akka/'as', ishee/'she', isaan/'they', nuti/'we' and etc. are commonly appeared Afaan Oromo stop words. For the stop word removal process, there is no standardized stop words list prepared for Afaan Oromo text. Therefore, we manually collected Afaan Oromo stop words lists. Then, the lists of collected Afaan Oromo stop words where saved in one file. The algorithm that shows the removal of stop words is shown in the following Algorithm

1. Read the file containing list of stop word
2. For each word in the sentence
Check the word against the stop words list
If a word is in stop word list then
 Remove a word
Else return the remaining word
End for
End if
3. Close the file

3.2.3. Normalization

In this work, some characters of the same words are sometimes represented in uppercase or lowercase in the corpus as well as in the user input and hence we have normalized them into lowercase. The purpose of normalization in this case is to make similar the words in different cases in our corpus. This is the observation that many different strings of characters often convey essentially identical meanings. Given that to get at the meaning that underlies the words, it seems reasonable to normalize superficial variations by converting them to the same form. The most common types of normalization are case folding (converting all words to lower case). Case folding is easy in Afaan Oromo. For example: **BAREEDU KOO/'MY BEAUTIFUL'** similar to **bareedu koo/'my beautiful'**.

3.3. Feature Extraction

Feature extraction is the process of increasing information density by retrieving key properties from a larger element of data. The representative of these large datasets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original dataset.

This process is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. It can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM) has an embedding layer between the input and LSTM and Bi-LSTM layer that used to create vector representation for incoming words as input. Likewise, Feed Forward Neural Network has its own embedding layer for word embedding purpose. The weights for the embedding layer can either be initialized with random values or more commonly, they are initialized with third-part word embedding's such as word2vec, GloVe or fasttext (or others) and these weights can optionally be fine-tuned during training. Nevertheless, in this research, we initialize the weight with a random value that is done by the embedding layer of Neural Network.

3.4. Splitting Training and Testing Dataset

In this subsection, we illustrate the splitting of training and testing datasets for our implementation. According to (Getachew & Alemu, 2019) division of training and testing dataset have an impact on the performance of deep learning approach. Artificial Neural Network need to learn more from training dataset in order generalize unseen dataset. One of the reasons for non-generalizing and over-fitting of Artificial Neural Network is having less training set. Hence, testing your model with high dataset can let you conclude whether your model is generalized or not. But most of the time division of training and testing dataset depend on how much dataset you have. As stated (Chamikara, 2014) if your dataset is small it's better splitting training and testing dataset into 90/10. In this

case we split our dataset into 90/10 which means 90% of data are used to train our model, while 10% are used to test the proposed Artificial Neural Network model since our dataset is small. This process can be done by using the scikit-learn library of python programming language. As stated above section we have 1005 emotional sentence, with this split 904 data can be used for training a model and others 101 were used for testing purposes.

3.5. Classification Algorithm

3.5.1. Feed Forward (Multilayer Perceptron) Neural Network

The multi-layer perceptron (MLP) model is also known as a simple feed forward Neural Network, which means the connections between the network neurons do not form a sequence or cycle rather it flows forward. In other word the signal between neurons moves only in one direction from input to the output neuron associated to them. All the nodes which exist in multi-layer perceptron (MLP) model are neurons with a non-linear activation function. Those functions then calculate an initial set of weights and forward to any number of hidden layers. The time it calculates and forward a set of input and weights to hidden layer was identified and governed by the parameters you pass to the algorithms. This means the algorithm of your choice for the activation and loss function, and the number of nodes that you allow the network to use. The architecture of the model is shown in Figure 2.

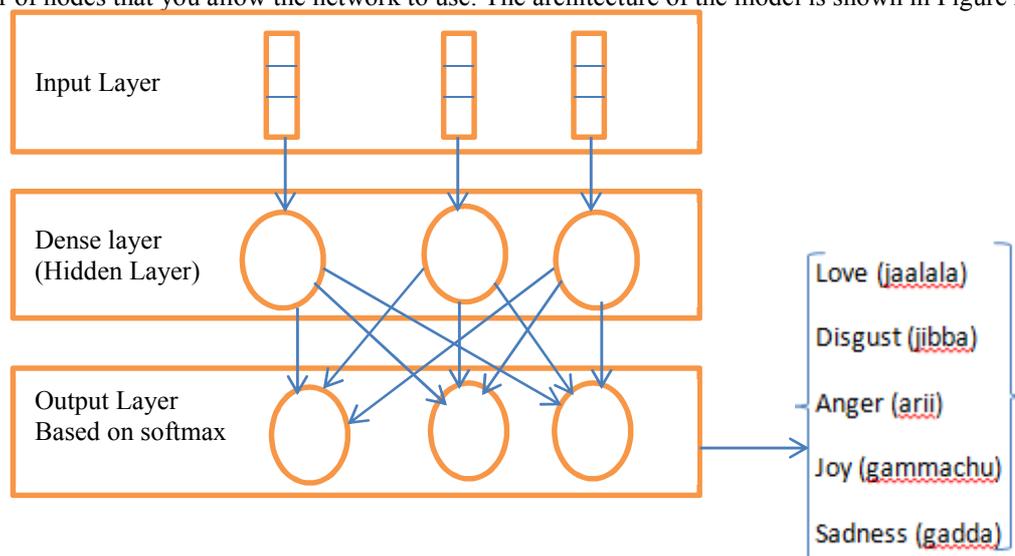


Figure 2 Feed forward (Multilayer perceptron) neural network architecture for Afaan Oromo Emotion Detection

Figure 2 Illustrates that the feed forward (Multilayer perceptron) is composed by the following layers: the input layer, hidden layer, and output layers.

The input layer is where we input or feed matrix of features and labels. Then afterward the activation function calculates an initial set of weights and forward to hidden layers (dense layers). The final result comes out in the output layer.

The dense layers in our model have two parameters: the output dimensionality and the activation function. The output dimensionality is defined as the embedding size and the activation was set to a Relu function.

While in the output layer we also used two parameters: the output dimensionality and the activation function. The output dimensionality represents the number of output dimensions of the model and is selected according to the specific task and the activation was set to a Softmax function.

In addition, we train the model with the Adaptive Moment Estimation (adam) due to the fact that works well in practice and out performs SGD (P.Kingma & Ba, 2015). Adaptive Moment Estimation (Adam) is a stochastic optimization algorithm and calculates adaptive learning rates for different parameters. So, it adapts its learning rate during training and optimizes the learning rate for every parameter. By using this approach, we need to specify two hyper parameters; those are batch size and initial learning rate. Batch size is used in the training step and initial learning rate used for updating the network weights. Then after we do a grid search on those parameters.

3.5.2. Long-Short Term Memory (LSTM)

Long-short term memory is an improved type of recurrent neural network. This recurrent neural network is a special type of deep learning designed to solve the sequence problem by learning short term temporal dependence and contextual information from the input data. But this recurrent neural network is suffered by temporal dependence if the sequence of text is large; it became hard to recapping information from earlier time step to the later ones. So Recurrent Neural Network can forget important information from the beginning if you are predicting from the large text. Recapping, RNN is suffered by exploding and vanishing gradient problem during the training of the model. LSTM was a solution to this short term-memory of recurrent neural networks.

It improves the learning ability of the long-time sequence of data (Hochreiter & Schmidhuber, 1997). The structure of the LSTM is described in Figure 3. LSTM has a different activation function layer called gate i.e. forget get, input gate, an output gate, which can regulate the flow of information. There is also a called cell state that runs through the whole-time sequence and it's controlled by gates of an LSTM.

The first step of LSTM as equation (1), is identifying information that is not required and will be thrown away from the cell state. This decision is made by sigmoid layer called forget gate. In this case, LSTM considers the input data from the present time x_t and the output for the previous time step h_{t-1} and is then multiplied by the weight w then after bias is added. Later on, the result is sent to the sigmoid function to obtain how much information to be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots\dots\dots (1)$$

The second step as equation (2) and (3), is to decide what new information we are going to store in the cell state. Those whole processes comprise of following steps. A sigmoid layer called the input gate layer decides which value will be updated. Next, a tanh layer creates a vector of new candidate values that could be added to the state. After that, combine these two parts to obtain the result of the input gate layer.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots\dots\dots (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \dots\dots\dots (3)$$

In step three as equation (4), it will update the old cell state, C_{t-1} into the new cell state, C_t . first it multiplies the old state C_{t-1} by f_t , forgetting the things we decide to forget earlier. Then it adds $i_t * \tilde{C}_t$. This is the candidate, scaled by how much we decide to update each state value. The new information to be stored in the cell state and the updated cell state C_t is obtained.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots\dots\dots (4)$$

In the final step, the output gate runs sigmoid function to decide which parts of the cell state we are going to output as shown in equation (5).

Then it put the cell state through tanh (push the value to be between -1 and 1) and multiply it by the output of sigmoid gate so that we only output the parts we decide to, as shown in the equation (6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \dots\dots\dots (5)$$

$$h_t = o_t * \tanh(C_t) \dots\dots\dots (6)$$

This all equation of LSTM component is used to compute the output of the hidden layer of h_t .

Since our work is many to one, which means entering a sequence of the sentence containing emotion as input and identifying a single class of emotion as output, it only uses the final hidden state to produce the one output y_k . Each input will be a vector representing a word from the text. The output y_k will be a vector containing five numbers representing joy, love, sadness, anger and disgust. We apply softmax to turn those values into probability and ultimately decide between joy, love, sadness, anger, and disgust.

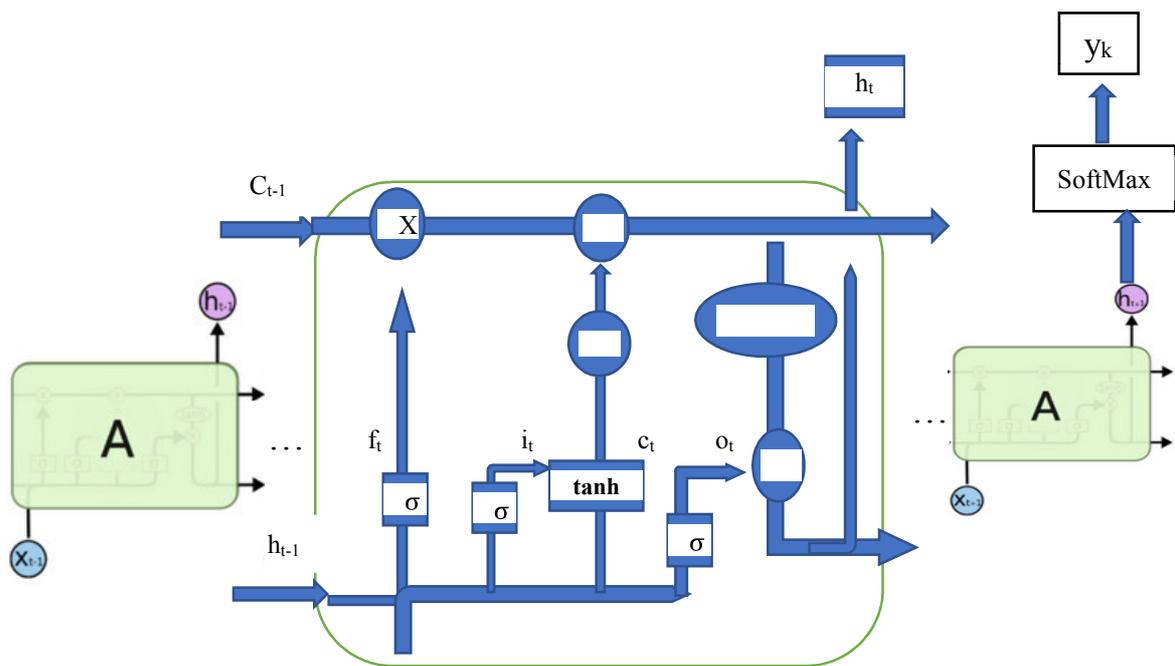


Figure 3 LSTM structure for Afaan Oromo Emotion Detection

In this study for the Afaan Oromo emotion detection model, the LSTM structure with two LSTM layers is applied. We input textual feature representation of the sentence into the model for training and then the emotion output Y_k for the input feature is determined.

Additionally, to get more accuracy different LSTM network configuration was performed with different parameters i.e. embedding dimension, dropout, Memory unit, epochs, Learning rate, and Batch size.

3.5.3. Bidirectional LSTMs

The bidirectional Long Short Term Memory model was presented with the aim of refining the performance of standard Recurrent Neural Networks and its extension Long Short Term Memory. Recurrent Neural Network and Long Short Term Memory models not have access to future information on a current state. To solve this limitation, the bidirectional Long Short Term Memory model connects two LSTM layers that go side by side in opposite directions to the same output, so that the output layer can access information from past and future states. The architecture of Bi-LSTM is presented on Figure 4.

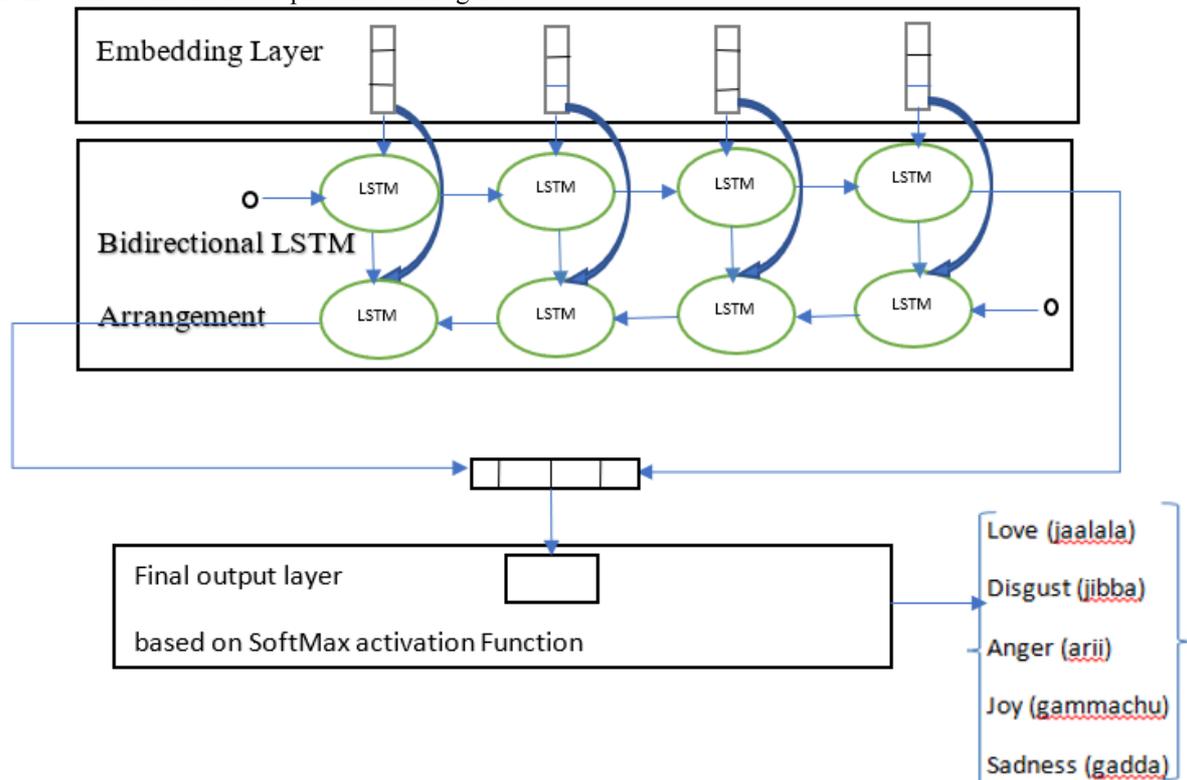


Figure 4 Bi-LSTM architecture for Afaan Oromo Emotion Detection

Figure 4 illustrate that the bidirectional Long Short Term Memory model is composed by one embedding layer, one LSTM layers with a bidirectional arrangement and at the end an output layer.

Each of these Bi-LSTM layers has their own parameters. Like LSTM model different LSTM network configuration was performed with different parameters i.e. embedding dimension, dropout, Memory unit, epochs, Learning rate, and Batch size to get more accuracy.

3.6. Evaluation Method

As depicted in the earlier subsection our dataset is divided into training and testing data according to 90/10 rule. In other words, 90% of the dataset is for the training set and 10% of data is for the test set. This testing dataset is used for evaluation of the effectiveness and efficiency of Artificial Neural Network, our proposed algorithm. In this study the testing dataset can be evaluated using the confusion matrix which provides the summary of prediction results on the classification problem. This confusion matrix classified into two classes' i.e. positive and negative class which is later classified as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

- TP is the sentence that is actually positive and will be predicted as positive. In another word the sentence that is actually emotion X and will be predicted as emotion X
- TN is the sentence that is actually negative and will be predicted as negative. In this case the sentence that is actually not emotion X and will be predicted as not emotion X
- FP is the sentence that is actually negative but predicted as positive. This means that the sentence that is actually not emotion X but predicted as emotion X

- FN is the sentence that is actually positive but predicted as negative. In another words the sentence that is actually emotion X but predicted as not emotion X.

From those kinds of the confusion matrix, the average value of accuracy, precision, recall, and F-measure can be obtained, which are planned to evaluate our experiment.

The accuracy of our algorithm can be calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy reflects how close a value is to its true value. Nevertheless, there are problems with accuracy. In other word 99% accuracy doesn't mean it's almost perfect rather than it depends upon the problem.

To obtain the value of Precision we divide the total number of the correctly predicted positive sentence by the total number of predicted positive sentence like the following formula:

$$precision = \frac{TP}{TP + FP}$$

High precision specifies a sentence labeled as True Positive is indeed a positive and small number of False Positive. Another method is recall which can be defined as the ratio of the total number of correctly classified True Positive sentence divided to the total number positive sentence

$$recall = \frac{TP}{TP + FN}$$

High recall shows the class is correctly classified with a small number of False Negative.

High recall, low precision means that most of the Positive sentences are correctly classified with low False Negative but there are a lot of False Positives. Were as low recall, high precision shows we miss a lot of Positive sentence with high False-negative but those predict as positive are indeed positive with low False Positive. So that high recall and precision are better, but achieving it at the same time is impossible. F-measure is proposed as a solution to balance precision and recall. In this study, we calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean and it's always nearer to the smaller value of Precision or Recall.

F-measure is calculated as:

$$F - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{Precision} + \text{recall}}$$

3.7. Tools

In this study, we have used python programming language due to its simplification of complex data sets and make sense of a large collection of data.

Moreover, one of the most powerful and easiest Python libraries for developing and evaluating artificial neural network models, is Keras which wraps the efficient numerical computation libraries TensorFlow were applied in this study. The benefit of these libraries is mainly that you can get started with neural networks in an easy and fun way.

In addition to Keras libraries, scikit-learn was applied in this study to split data into training and testing datasets. We also applied pandas, NumPy and matplotlib to do data analysis, provide fast mathematical computation on arrays and matrices, and visualize data in a pictorial or graphical representation respectively.

However, out of all the other programming languages, Python provides better support to deep learning algorithms.

4. Result and Discussion

As we have discussed above, Artificial Neural Network has been implemented for this work. Based on the compiled dataset, we have trained and tested the algorithms. In this chapter, we have summarized their results and shown them.

4.1. Result of the Model

In the following section, we present the results achieved for Afaan Oromo Emotion Detection based on experiment performed. To give an overview of the results, Table 4 presents the best performance of the proposed three classification algorithms. It displays the test accuracy, average precision, average recall and average F1-score for each algorithm. We compute the performance of those three-classification algorithms on all 101-test dataset.

Table 4 the performance result of FFNN, LSTM, and Bi-LSTM

Model	Average precision	Average recall	Average F1-score	accuracy
FNN	0.65	0.65	0.64	66%
LSTM	0.78	0.77	0.77	78%
Bidirectional-LSTM	0.85	0.82	0.82	83%

The results of the ANNs will be presented in the next sections in more detail. Nevertheless, we will briefly summarize the outcomes here. The FFNN classifies the emotion sentence with an accuracy of 66%. The LSTM

accomplishes an accuracy of 78 % and Bi-LSTM classifies 83% correctly with the same test dataset as FFNN and LSTM. Generally, Bi-LSTM is the most accurate Model on our dataset.

4.1.1. Feed Forward Neural Network

The FFNN predicts our emotion dataset with an accuracy of 66%. The optimal hyper parameters for the FFNN as depicted in Table 6 are one hidden layers, 5 training batch size at 9 epoch, categorical cross entropy loss function and adam Optimizer with default learning rate of 0.001. The result is shown in the following Table 5.

Table 5 performance evaluation results using FFNN algorithm for Afaan Oromo Emotion detection

Emotion class	FP	FN	TP	TN	Precision	Recall	F1 score	Accuracy
anger(arii)	10	4	9	78	0.47	0.69	0.56	0.66
love(jaalala)	4	10	8	79	0.66	0.44	0.53	
joy(gamachu)	7	10	15	69	0.68	0.6	0.63	
disgust(jibba)	4	3	21	73	0.84	0.87	0.85	
sadness(gadda)	9	7	14	71	0.60	0.66	0.63	
Average result					0.65	0.65	0.64	

The results shown in the above Table 5 are the performances achieved using FFNN model. In addition to testing accuracy the proposed FFNN model achieved 65% for recall and precision, and 64% for F1-score in average. This can be interpreted as; the model has performance of 65% to classify emotion X from the dataset and out of classified emotion 65% of them were correctly classified as emotion X in the dataset.

Table 6 FFNN configuration for Afaan Oromo Emotion Detection

Hyper parameters	Values
Word length	200
Vocabulary size	4000
Input shape	73
Number of hidden layers	1
Batch size	5
Epochs	9
Number of nodes for each Layer	64
Optimizer and loss function	Adam with Default Learning rate (0.001) and categorical cross entropy

Likewise, the accuracy and loss results of the FFNN model with hyper parameters of the training and validation dataset can be achieved as shown in Figure 5.

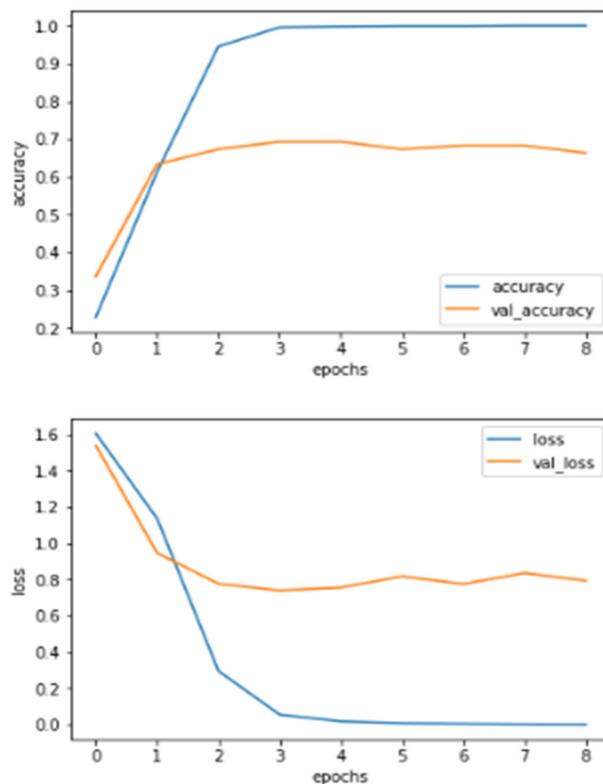


Figure 5 Result of Feed Forward Neural Network model accuracy and loss for the training and validation with epoch.

4.1.2. Long Short-Term Memory

After necessary preprocessing and feature extraction we feed it into LSTM model in order to train and evaluate the proposed LSTM model. So, that on the selected LSTM model experiment was conducted using Afaan Oromo Emotion Datasets. The result of experiment is shown in the following Table 7.

Table 7 performance evaluation results using long short-term memory algorithm for Afaan Oromo Emotion detection

Emotion class	FP	FN	TP	TN	Precision	Recall	F1_score	Accuracy
anger(arii)	2	4	9	86	0.81	0.69	0.75	0.78
love(jaalala)	7	3	15	76	0.68	0.83	0.75	
joy(gamachu)	5	8	17	71	0.77	0.68	0.72	
disgust(jibba)	3	2	22	74	0.88	0.91	0.89	
sadness(gadda)	5	5	16	75	0.76	0.76	0.76	
Average result					0.78	0.77	0.77	

The results shown in the above Table 7 are the performances achieved using long short-term memory model. With the same dataset the proposed long short-term memory algorithm accomplished testing accuracy of 78% by trying different hyper parameters. Subsequently, we get the following LSTM network configurations for Afaan Oromo Emotion Identification as shown in Table 8.

Table 8 LSTM network configuration for Afaan Oromo Emotion Detection

Hyper parameters	Values
Word length	200
Vocabulary size	4000
Embedding dimension	64
Number of memory units	64 for each LSTM layers
Dropout layer	0.5
recurrent dropout	0.2
epochs	6
Batch size	5
Optimizer and loss function	Adam with default learning rate (0.001) and categorical cross entropy

Moreover, the proposed long short-term memory model achieved 77% for recall, 78% for precision, and 77% for F1-score in average. This can be interpreted as; the model has performance of 77% to classify emotion X from the dataset and out of classified emotion 78% of them were correctly classified as emotion X in the dataset. Furthermore, the LSTM is measured by the average accuracy and loss of training and validation dataset for each epoch. Figure 6 shows the accuracy and loss results of the LSTM model with best hyper parameters on each epoch of the training and validation stages.

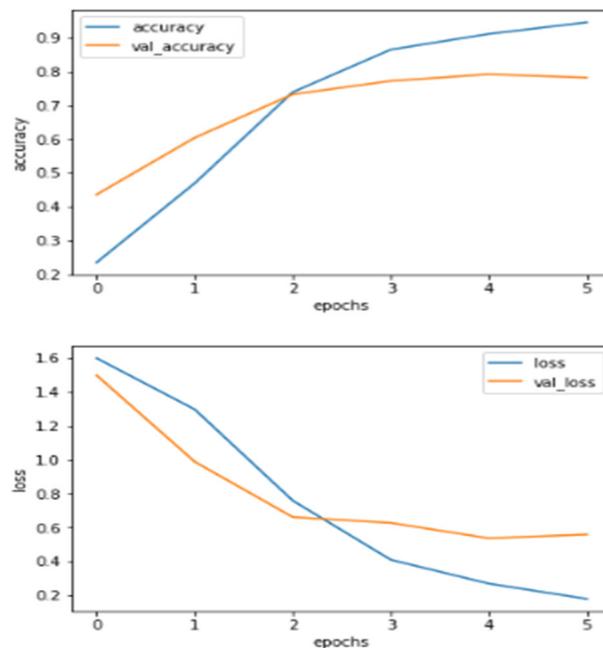


Figure 6 Result of Long Short-Term Memory model accuracy and loss for the training and validation with epoch.

4.1.3. Bidirectional-LSTM

Based on experiment conducted on Afaan Oromo Emotion Dataset Bi-LSTM model achieved the best promising result as shown in the following Table 9.

Table 9 performance evaluation results using Bidirectional long short-term memory algorithm for Afaan Oromo Emotion detection

Emotion class	FP	FN	TP	TN	Precision	Recall	F1_score	Accuracy
anger(arii)	0	4	9	88	1	0.69	0.81	0.83
love(jaalala)	5	1	17	78	0.77	0.94	0.85	
joy(gamachu)	3	9	16	73	0.84	0.64	0.72	
disgust(jibba)	5	1	23	72	0.82	0.95	0.88	
sadness(gadda)	4	2	19	76	0.82	0.90	0.86	
Average result					0.85	0.82	0.82	

The results shown in the above Table 9 are the performances achieved using Bi-LSTM model. We get the testing accuracy of 83% using the proposed Bidirectional long short-term memory model. In addition to testing accuracy the proposed Bi-LSTM model achieved 82% for recall, 85% for precision, and 82% for F1-score in average. This can be interpreted as; the model has performance of 82% to classify emotion X from the dataset and out of classified emotion 85% of them were correctly classified as emotion X in the dataset. In addition to precision, accuracy, and f1 score, the Bi-LSTM is measured by the average accuracy and loss of training and validation dataset for each epoch. Figure 7 shows the accuracy and loss results of the Bi-LSTM model with best hyper parameters on each epoch of the training and validation stages.

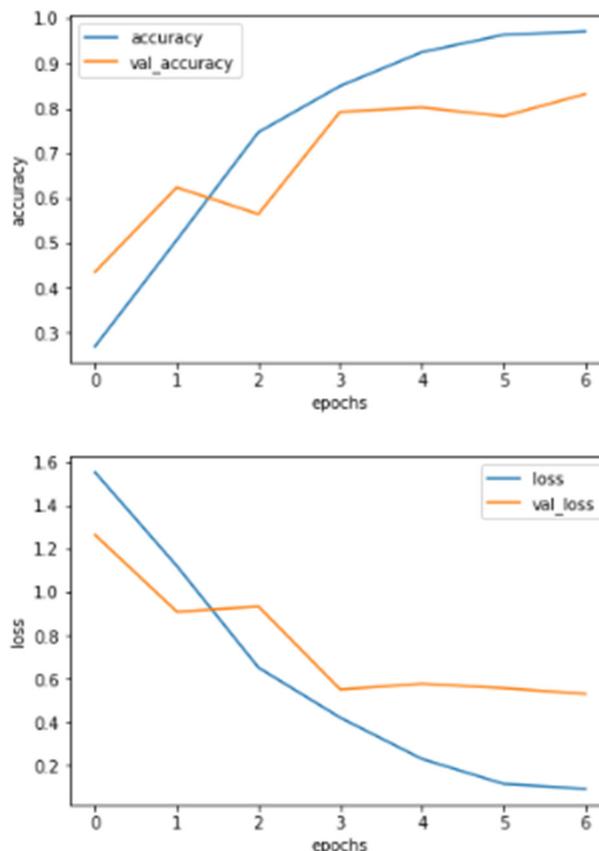


Figure 7 Result of Bidirectional Long Short-Term Memory model accuracy and loss for the training and validation with epoch.

4.2. Discussion of Experimental Results

4.2.1. Comparison of FFNN, LSTM and Bi-LSTM

As the result shown in the above Table 3 Bi-LSTM model outperforms LSTM and FFNN model. It seems that Bi-LSTMs are able to capture the underlying context better by traversing inputs data twice, from inward (left to right) and then from backward (right to left). On other word, the output layer of Bi-LSTM access information from past and future states. Even though LSTM models are capable of memorizing sequences of data, it has limitations in memorizing long sequences of data. Through a LSTM model, since the input data are passed only once from

forward (left to right), a certain number of input items can be fed into the training model. On the other hand, in a Bi-LSTM model, the training network needs to train not only the input data from forward (left to right), but also from backward (right to left).

Similarly, LSTM model outperforms FFNN model as illustrated in the above Table 4. Unlike FFNN, which are not generally able to handle sequential inputs and all their inputs (and outputs) must be independent of each other's, LSTM model were trained to learn sequence data from input and output using memory blocks connected into layers.

Also, in this experiment one of the most interesting observations is about the changes of loss values in each Epoch of proposed model. Figure 5, 6 and 7 shows the changes for loss for FFNN, LSTM and Bi-LSTM. As Figure 5 illustrate the training model still keeps continue learning from the data and tuning the parameters. The loss value starts at 1.6030 and then quickly falls into 1.1349 at the second epochs and significantly decreases into 0.0017 at 9th epochs. This indicates that the FFNN model keeps training its parameters and needs to fetch additional data to tune its parameters. Likewise, as figure 6 demonstrate the loss value starts at 1.5982 for epoch 1 and the quickly falls into 1.2961 at the second epoch and significantly decrease into 0.1786 at 6th epochs. This trend of change indicates that the LSTM model still keeps continue learning from the data and tuning the parameters. LSTM also need more data to train and tune its parameters. On the other hand, the trend of change for Bi-LSTM exhibit a similar behavior as observed for the FFNN and LSTM. As Figure 7 shows the loss value starts at 1.5678 and then quickly falls into 1.0874 at the second epochs and significantly decreases into 0.0396 at 7th epochs. This indicates that the Bi-LSTM model keeps training its parameters and needs more data to train and fine tune its parameters.

5. Conclusion and Recommendation

5.1. Conclusion

The study is used as groundwork for emotion detection model development for Afaan Oromo language. In this study we reviewed different approaches, related work of emotion detection and we have comprehended the concept, feature and characteristics of emotion detection.

Since there is no dataset for natural language processing of the language, sample dataset is designed and developed for the study.

Additionally, Feed forward neural network, Long-short term memory, and bidirectional Long-short term memory algorithms and 1005 labeled emotional sentence were used for emotion detection of Afaan Oromo Language.

In the study performance of algorithms is evaluated based on performed experiments. The first algorithm that was explored in the study is Feed forward neural network approach and it achieves 66% of accuracy. Another algorithm that was used in the study is long-short term memory and bidirectional long-short term memory; and they achieve 78% and 83% of accuracy respectively. Therefore, Bi-LSTM model outperforms LSTM and FFNN model because of the output layer of Bi-LSTM receive information from forward and backward direction.

5.2. Recommendation

There are further works that should be considered by future researchers in the area of emotion identification.

- The accuracy of artificial neural network depends on standardized and sufficient amount of dataset. Lack of standardized annotated dataset is one of the limitations in this study. So, it would be interesting to prepare standardized annotated dataset for the further study
- This study performs with feed forward neural network, Long Short-Term Memory and Bidirectional Long Short-Term Memory. Therefore, researchers and experts can repeat the same study with the same method and approach with huge dataset
- There are also other approaches that are different from artificial neural network which can be applied for similar problem. So, experts and researchers can develop the same title for the language in rule based, machine learning and hybrid approach to compare the result with the performance of this study.
- Additionally, Facebook contain huge number of emoticons and emoji that carry joy, sadness, angry, disgust, love, and etc.; hence, it would be interesting to add emoticons and emoji and check the performance of the model.
- Moreover, it would be interesting to add fear, surprise, and more class and check the performance of the classifier. However, in this case, the training and testing datasets have to include fear, surprise, and more class samples to feed the model and evaluate it.
- The same approach and method can be repeated for developing emotion identification for other local languages, like Amharic, Tigrigna, Wolaita, etc.

Reference

Abate, J. (2019). Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments, Haramaya University,

- 2019.
- Abdullah et.al, S. S. (2018). SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning. *2018 17th IEEE International Conference on Machine Learning and Applications*. IEEE.
- Abreham, G. (2014). OPINION MINING FROM AMHARIC ENTERTAINMENT TEXTS. *OPINION MINING FROM AMHARIC ENTERTAINMENT TEXTS," Addis Abeba Universty, Addis Abeba, 2014*. Addis Abeba, Ethiopia: Addis Abeba Universty.
- Chamikara, M. (2014). Can someone recommend what is the best percent of divided the training data and testing data in neural network 75:25 or 80:20 or 90:10 ? Retrieved from https://www.researchgate.net/post/can_someone_recommend_what_is_the_best_percent_of_divided_the_training_data_and_testing_data_in_neural_network_7525_or_8020_or_9010/54840346d2fd643d268b45e9/citation/download.
- Cheng, B., & Titterington, D. (1994). Neural networks: a review from a statistical perspective. *Statistical Science* 9, 2–54.
- Effendi et.al, A. F. (2015). A Two-Stage Emotion Detection on Indonesian Tweets. *ICACISIS*, 143-146.
- Gaind et.al, V. S. (2019). Emotion Detection and Analysis on Social Media. *Global Journal of Engineering Science and Researches (ICRTCET-18) (2019)* 78-89.
- Gebremeskel, S. (2010). Sentiment Mining Model for Opinionated Amharic Texts. *Unpublished Masters Thesis*. Addis Ababa, Addis Ababa, Ethiopia: Addis Ababa University.
- Getachew, Y., & Alemu, A. (2019). Deep Learning approach for Amaharic Sentiment Analysis University of Gondar.
- Gurney et.al. (1997). *An introduction to neural networks*. London and New York: UCL Press.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, vol. 9, no. 8, pp. 1735-1780.
- Kaur, J., & Saini, J. R. (2014). Emotion Detection and Sentiment Analysis in Text Corpus: A Differential Study with Informal and Formal Writing Styles. *International Journal of Computer Applications (0975 – 8887) Volume 101– No.9*.
- Kuhn, M., & Johnson, K. (2013). *Data Pre-processing*. In: *Applied Predictive Modeling*. Springer, New York, NY.
- Mengistu, B. (2013). *Sentiment Analysis for Amharic opinionated text*. Addis Ababa: Ababa university.
- Miner et.al, A. M. (2016). Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health. *JAMA internal medicine* Vol. 176, 619–625.
- Oljira, M. (2020). *SENTIMENT ANALYSIS OF Afaan Oromo social media sentiment analysis using Machine learning approach*.
- P.Kingma, D., & Ba, J. (2015, Dec.). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representation*. San Diego.
- Philemon, W., & Mulugeta, W. (2014). A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts. *HiLCoE Journal of Computer Science and Technology*, No. 2, Vol. 2, 87.
- Tariku, W. (2017). *Sentiment Mining and Aspect Based Summarization of Opinionated Afaan Oromoo News Text*. Debre Berham: Debre Berhan University.
- Thomas, D. (2017). *An introduction to neural networks for beginners*.
- Tilahun, T. (2014). Linguistic Localization of Opinion Mining from Amharic Blogs. *International Journal of Information Technology & Computer Sciences Perspectives* © Pezzottaite Journals., 895.
- Vijay et.al, D. B. (2018). Corpus Creation and Emotion Prediction for Hindi-English Code-Mixed Social Media Text. *Proceedings of NAACL-HLT 2018: Student Research Workshop Association for Computational Linguistics*, (pp. 128–135). New Orleans, Louisiana.