Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Impact of Normalization and Informal Opinionated Features on Amharic Sentiment Analysis.

By:
Abebaw Zewudu Gedamu

Advisor:
Dr.Eng. Getachew Alemu

A thesis submitted in partial fulfillment of the requirements for the degree of ***Master of Science*** in **Computer Engineering**

January 1, 2024

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Impact of Normalization and Informal Opinionated Features on Amharic Sentiment Analysis.

By: Abebaw Zewudu Gedamu

Dr.Bisrat Deribsa	______________________
Dean, SECE, AAiT	Signature
Dr. Eng. Getachew Alemu	______________________
Thesis Advisor	Signature
Dr. Fitsum Assamnew Andargie	______________________
External Examiner	Signature
Dr. Surafel Lemma	______________________
(Internal Examiner)	Signature

Declaration

I thus certify that this thesis is a presentation of my original work, completed after registering for a Master's degree at Addis Ababa University, and that it has not previously been included in a thesis submitted to this or any other institution for a degree, diploma, or other qualifications. I have reviewed the current research ethical rules at the university and take responsibility for carrying out the procedures in accordance with the university's senate committee. I have sought to identify any potential risks associated with this research, have gotten the necessary ethical and safety approvals, and have made every effort to conduct this research safely.

January 1, 2024

Acknowledgment

I'd like to express my heartfelt gratitude to my parents, who are the most important people in my life, as well as our research advisor, Dr. Eng. Getachew Alemu, for providing me with the wonderful opportunity to write this wonderful thesis on the topic Impact of Normalization and Informal Opinionated Features on Amharic Sentiment Analysis. Second, I'd want to express my gratitude to Dr. Surafel Lemma, one of my lecturers, for his significant assistance in finishing this thesis in such a short period of time.Without them, we would not be where we are now, and their contribution to my thesis is invaluable. I'd want to express my gratitude to my friends who, despite their hectic schedules, assisted me greatly in acquiring various information and mentoring me from time to time in the creation of this thesis. Finally, in all humility and gratitude, I would want to express my gratitude to all those who have assisted me in putting my thoughts into action.

Thanking you, Abebaw Zewudu

Abstract

Sentiment analysis is the computational study of people's ideas, attitudes, and feelings concerning an object via social media networks. To analyze the sentiment of these textual contents, previous study relied on formal lexicon and emoji with semantic and syntactic information as a feature. However, informal language is now being used to express opinions the majority of the time. It is challenging to create embedding features from unlabeled Amharic text files due to morphological difficulties of the informal and unstructured nature of Amharic informal texts. Despite the fact that normalization algorithms have been developed to convert informal language into its standard form, their impact on tasks such as sentiment analysis remains unknown.

To address the challenge of Amharic sentiment analysis, we apply state-of-the-art solutions to problems, such as utilizing normalization and embedding Amharic informal text contains opinionated with lowered word frequency parameters as automatic features on CNN-Bi-LSTM approaches. Using a combination of word and character n-gram embedding, potential information is generated as word vectors from unlabeled Amharic informal text files. In the studies, the maximum recall was 91.67 percent. When compared to state-of-the-art approaches using formal lexicon and emoji as a feature on Bi-LSTM, an average recall improvement of 2.8 was attained. According to the results, labeling with a mix of informal, formal lexicons, and emoji achieves 1.9 better accuracy than labeling with just formal lexicons and emoji.

KeyWords: Amharic, Sentiment analysis, Informal, CNN-Bi-LSTM.

Contents

List of Figures

List of Tables

Abbreviations

Bi-LSTM ……………………………………………Bi directional long short-term memoRY

BOW …………………………………………………Bag of words

CBOW ………………………………………………Continuous bag of words

CNN …………………………………………………Convolution neural network

DF ……………………………………………………Document frequency

LSTM ………………………………………………Long short-term memory

MNB ………………………………………………Multi Nominal Naive Bayes

NB …………………………………………………Naive Bayes

NLP …………………………………………………Natural language processing

SA …………………………………………………Sentiment analysis

SG …………………………………………………Skip gram

SK learn ………………………………………Scikit learn

SVM ………………………………………………Support vector machine

TF …………………………………………………. Term frequency

TFIDF.……………………………………………Term frequency by inverse document frequency

W2VEC……………………………………………Word2vec

WORD2VEC ……………………………………Word to vector

Chapter 1

Introduction

1.1 Background

Sentiment Analysis, also known as opinion mining, is the computational analysis of people's ideas, attitudes, and emotions with the goal of identifying whether a portion of a person's concept and opinion content is positive or negative [16]. The entity or objects can be used to represent individuals, events, or themes. Its purpose is to keep track of service providers and vendors in order to construct effective marketing and political campaigns. Even for people, evaluating whether a section of text represents a person's idea and opinion, particularly for things that use varied writing styles and informal forms of communication as positive or negative, has some challenges [23]. People nowadays use social media as a regular mode of communication since it allows them to send messages quickly and easily from a number of devices [18]. We noticed that there is a considerable volume of informal communications conducted by users on such networks in a range of natural languages from all over the world when we examined actual data from social media. People are using informal text to share their feelings regarding publicly stated worries, which is one of the most popular social media trends right now. As a result, an informal Amharic text was formed to portray people's diverse feelings in response to topics brought up by the internet-connected community. The term "informal language" refers to communication that does not adhere to standard grammar and spelling rules [5]. We need to use natural language processing to manipulate sentiment analysis during an unstructured and informal way (NLP). Because it is required for examining the role of a customer's sentiment, views, attitudes, and feelings regarding an object in an informal manner. Sentiment analysis is challenging, especially in languages like Amharic, which have few resources and a complex grammatical structure. A small amount of resources, such as training data for sentiment analysis, have an influence on the accuracy of the system. Amharic, like other low-resource and morphological rich languages, confronts similar challenges. When it comes to languages with a lot of resources, such as English, and Arabic, numerous studies have been conducted. Sentiment analysis for the Amharic language has received little attention [16]. Because the Amharic languages are related to other Semitic languages such as Arabic or Hebrew, they are morphological rich. The Amharic sentiment analysis problems were solves using different approaches. Rule-based classify reviews are based on how many positive and negative terms are present in the document [17]. Besides, machine learning performs with less human intervention [44]. The idea of this approach is to train a function capable of determining the text classification of an unseen document using a corpus of texts labeled by annotators. Word embedding was utilized as a feature in the past, and it allowed for the detection of both semantic and syntactic relationships between words [16]. There are various limits to these efforts. Instead of studying word representations, it will be difficult to explore the internal structure or morphology of informal terms [19]. The author [24] indicated that using informal lexicon as a feature for automatic recognition of non-standard terms and normalizing as part of the Preprocessing procedure could improve sentiment analysis. This study proposes a novel idea for sentiment analysis: normalizing and using unstructured morphology of Amharic informal via a concatenation of word embedding and character n-gram embedding. Additionally, consider the impact of employing informal language as a feature for automatic labeling in Amharic sentiment analysis.

1.2 Problem Statement

Due to social concerns, there is a strong motivation to improve the current sentiment analysis of Amharic languages. One of the most popular social media trends right now is for people to use informal text to express their feelings about publicly reported concerns [46]. The state-of-the-art study [16], on the other hand, uses formal lexicon and emoji as features, as well as neural word embedding as an automatic feature extractor model. There is a gap to apply it due to their informal morphological style and less grammatical worried practice on sentences. Because word embedding can only provide a certain amount of semantic information [51]. Due to the difficulty of informal text, sentiment classification may become more challenging, especially for a morphological complex language like Amharic. Furthermore, morphological analysis, particularly normalization, is one of the Preprocessing that have been developed to convert informal language into its standard form [16], their impact on tasks such as Amharic sentiment analysis remains unknown. According to the author [24], a system for automatic recognition of non-standard words and normalization as part of the Preprocessing step is essential. This could enable Amharic sentiment analysis work more effectively. However, no previous research has looked into the impact of normalizing and using Amharic informal words as a sentiment analysis feature. Finally, in this research, we'll look at an experiment in order to answer the following research questions.

RQ1. Is there any impact on Amharic sentiment analysis if the informal lexicon is used in automatic labeling?

RQ2. Can we improve Amharic sentiment analysis performance by embedding informal terms as features and normalizing them as a Preprocessing step?

1.3 Objective

The overall goal of this research is to explore how normalization and embedding of Amharic informal terms affects sentiment analysis in Amharic.

1.3.1 Specific objective

- Prepare a new publicly accessible data set or corpus from several social media sources.

- Creates a combination of word and character n-gram embedding as features for sentiment analysis classifier input.

- Comparing the effects of labeling text with simply a labeled formal lexicon against labeling material with both a labeled informal and formal lexicon .

- Compare the performance of several sets of classifiers Using a combination of word and character n-gram embedding as automatically generated features and normalization as Preprocessing step.

- Assess the impact of minimum word frequency parameter on several automatic feature extractors with an Amharic informal contains comment.

1.4 Research Design Methods

Literature review, data collecting, data Preprocessing, feature extraction, deep neural network model construction and deployment, and model evaluation were the five primary steps of research methods used in this study.

1.4.1 Literature Review

Literature study of theoretical foundations and existing feature extractor and classification model for sentiment analysis in both local and foreign languages.

1.4.2 Data Collection

For the studies, we used previously available datasets and automated web scraping from social media sites including Facebook, YouTube, and Amharic slang dictionary websites to gather annotated and unlabeled corpora.

1.4.3 Data Preprocessing

Various preparation operations, including as data cleaning, normalization, stop word removal, and Tokenization, are conducted to improve the system's performance.

1.4.4 Feature Extraction

After reviewing various papers and investigating existing feature extraction mechanisms, we chose a concatenation of word embedding and character n-gram as automatic feature extraction and appropriate selection for our thesis.

1.4.5 Model Development

After selecting the automatically generated features and the vector result as input, the Amharic sentiment analysis is built using various deep neural classifier networks.

1.4.6 Model Evaluation

Models created with automatically learned features are fed into the proposed models, and performance is measured using precision, recall, F-score, and accuracy metrics.

1.4.7 Limitation and Scope

The goal of this study is to create a sentiment analysis system for Amharic formal and informal text-based using a combination of word-level and character-level embedding. Additionally, CNN-BiLSTM is used as a classification algorithm. Our work does not take into account the following points, such as Amharic expressions like ***"ቅኔያዊ አነጋገር"***, and Amharic with an English accent.

1.5 Contribution

The foremost contributions of this research can be listed as below:

- The first Amharic informal text has been gathered and annotated, it now serves as a corpus for future studies.
- We have replaced the word embedding feature extractor that is required in the case of automatic design with a very crucial feature extractor that does a semantic and morphological information.
- We created Amharic sentiment analysis, which uses automatically produced a system to replace human jobs with machines that can directly process, analyze, and manipulate Amharic informal text data.

1.6 Organization of the Thesis

The thesis is broken down into six chapters. The first chapter deals with background about sentiment analysis and Amharic informal texts, and the gap from other researchers that we want to tackle. The second chapter tells us about the overall theoretical background of the algorithms we used in this thesis. The third chapter deals with different kinds of literature reviews with foreign and local sentiment analysis works. The fourth chapter deals with the proposed methodology of our model to tackle the problems that we identified on the statement using different feature extractor and classifier algorithms on nature of datasets. The fifth chapter focuses on the experimental result and performance analysis after the implementation of our proposed methodology. Finally, the sixth chapter focuses on the conclusion of our conclusive information about the results we have obtained, as well as recommendations for enhancing the state of art performance in Amharic sentiment analysis, which will be covered in the future or by the next interested researcher.

Chapter 2

Theoretical Background

2.1 Background

This chapter provides the fundamental theoretical foundations required to understand the concept of Amharic sentiment analysis and solve the challenge. It includes descriptions of Amharic sentiment classification techniques as well as a quick overview of neural word embedding as an autonomous feature extractor used in sentiment analysis.

2.2 Overview of Amharic Language

The Federal Democratic and Republic of Ethiopia uses Amharic as its official language. Following Arabic, amharic is the second most widely spoken semiotic language family [33]. Amharic is a difficult language to learn for a variety of reasons. In comparison to the English language, it has a fairly complex morphology [33, 43, 39]. Amharic is a highly inflectional and derivation language, making morphological study extremely difficult.

2.3 Challenges of Amharic Languages

One of the most challenging elements of processing natural languages is the morphological analysis of related words. [10]. The same uttered term might have multiple meanings in different languages, especially in Amharic. Other issues include a lack of capitalization, homophone variants, and a lack of punctuation. Abbreviations are not employed since there are no hard and fast rules when it comes to abbreviating Amharic nouns. For example, consider the phrase "ዶክተር" can be abbreviated as **ዶር**, **ዶ.ር**, and **ዶ/ር** [16]. The morphological analysis is presented briefly because it is the emphasis of this work. the focus of this study is on Amharic morphological analysis, is very briefly presented.

2.3.1 Amharic Morphological Analysis

The study of morphology in numerous domains is known as morphological analysis [43, 39, 10]. The internal organization of words or sub-words is the subject of morphological analysis (linguistics). The word obtained by eliminating letters at particular word locations w is the sub word information of a word w. The morphology of Semitic languages is extremely rich. Because Amharic is a language, single word might have several different forms. For example, the word **አደረሰ** can be inflected to **ያደርሳል**, **አስደረሰ**, **ያስደርሳል** and other forms can be inflected to and other forms.

To solve this problem different researchers apply Preprocessing tasks [43, 39]. The author [16], using various preprocessing, particularly stemming in Amharic, the morphology challenge is attempted to be removed. However, previous research has shown that due to the morphological complexity of the Amharic language, we cannot directly use morphological analyzers, particularly steamers, for Amharic language.

2.3.2 Homophone Variations

Various characters in the Amharic language are pronounced similarly yet have different symbolic meanings. This adds to the number of features without providing any benefits.There is no clear guideline as to when and where each character should be used [27, 16]. For instance, the word "**ጸሀይ**" ('sun') can be represented in Amharic as **ጸሀይ**, **ጸሐይ**, **ጸኀይ**,**ፀሀይ**, **ፀሐይ**. People use the characters **ሀ**, **ሐ** or **ኀ** and **ጸ** or **ፀ** interchangeably to write the same word.

2.4 Sentiment Analysis

Sentiment analysis or opinion Mining is the computational study of people's opinions, attitudes and emotions toward an entity [16, 17, 2, 8]. The entity can represent individuals, events or topics. Public reviews are uses to evaluate a certain object, i.e., person, product or location and might be found on different websites like Amazon. The opinions can be categorized into negative, positive or neutral [3].

2.4.1 Levels of Sentiment Analysis

Sentiment analysis can be valuable on different levels; we can see these in detail as follow.

Word level: is the analysis that determines the polarization of a word, i.e., if it is a positive, negative and neutral word [45].

Sentence level: is the analysis that determines the polarization of a sentence by extraction of people's opinion or emotion in at the sentence or phrase level [16]. It is a sequence of words, which aims at defining an opinion on a subject [45].

Document level: is the analysis that determines the polarization of an overall opinion of a document by assuming that each document is about a certain entity [16]. It is a more difficult level compared to the others, because, when the amount of words increases that distorts learning and complicates the prediction of polarity [45].

2.5 Sentiment Analysis Approaches

Sentiment analysis system approaches are classified into groups based on the methods employed. Some sentiment analysis systems are rule-based (linguistic approach), machine learning (statistical approach), and hybrid, whereas another Sentiment Analysis system is based on deep learning.

2.5.1 Rule Based Approaches

In this paper [17], used the rule-based model, which uses sentiment and a subjective lexicon of terms.By assigning the first polarity weight to all selected opinion terms, the model classifies positive and negative opinion terms as well as contextual valence sifters such as negations. Humans can understand rule-based systems, and they can be improved over time [15]. Although the term counting approach may be regarded as a valuable alternative for underdeveloped languages such as Amharic that face challenges in building a corpus, systems developed using this approach are difficult to scale [17]. They take a lot of time.

2.5.2 Machine Learning Approaches

Modeling the problem as a supervised learning problem is a popular approach. The knowledge behind this approach is to use a corpus of documents labeled by annotators to train a function capable of determining the text classification of an unseen document [44]. The machine-learning model can begin to make accurate predictions after it has been trained with enough training samples. Machine learning text classification is frequently far more accurate than human-crafted rule systems, especially for complex text classification. Machine learning approaches are classified into three types: supervised, unsupervised, and reinforcement learning. SVM, Random Forests, Decision CNN, logistic regression, and Nave Bayes are the most commonly used supervised machine learning algorithms.

Algorithms such as SVM and NB have become popular in Amharic sentiment analysis since the advent of machine learning.

Naive Bayes: The naves bayes works by compute the conditional probabilities of occurrence of two events. Based on the probabilities of occurrence of each individual event [42].This means that any vector that denotes a text will have to cover data about the likelihoods appearance of the words of the text. The naive bayes is an algorithm that can compute the likelihood of the text is belonging to their category [59].Its theorem arrange for a way of calculating the posterior probability, $P(c/x)$, from P(c), P(x), and P (x/c). Naive Bayes classifier adopts that outcome of the value of a predictor (x) on a given class (c) is self-determining the values of other predictors. This hypothesis is call as class conditional independence [50].

Support Vector Machine: When it comes to text classification, Support Vector Machines (SVM) is just one of many algorithms to choose from machine learning. SVM, like naive Bayes, does not require a large amount of training data to produce accurate results [22].It does, however, necessitate more computational resources than Naive Bayes. SVM, in a nutshell, is responsible for drawing a "line" or hyperplane that divides a space into two subspace. One subspace contains vectors that are members of a group, and another subspace contains vectors that are not members of that group [25].Those vectors are representations of your training texts, and a group is a tag you've assigned to your texts.

2.5.3 Hybrid Approach

Both rule-based and statistical approaches have advantages and disadvantages. As a result, rule-based and statistical approaches are frequently combined to benefit from their synergy effect, giving rise to the hybrid approach. For opinion analysis, a hybrid approach combines sentiment knowledge, machine learning, and a general linguistic framework [55].

2.6 Feature Extraction in Sentiment Analysis

In this section, we'll go over how features are extracted so that they can be used to perform classification later on. Since the representation step converts written human language into a form that a computer can understand. We categorize feature sets into the following groups. Bag of words, n-grams, TF-IDF, word embedding, character n-gram embedding, combination word embedding, and character n-gram embedding are a few examples.

2.6.1 Bag of Words

The bag-of-words model is a text representation of words that simplifies natural language processing and information retrieval. To extract features for each sentence, a distribution of different words in the sentence must be computed, i.e. how many times each word in the vocabulary appears in the sentence [2]. The text is represented in this typical feature extractor by a bag of its words. Pay no attention to grammar or even word order while maintaining multiplicity. The text is transformed into a bag-of-words where each entry corresponds to the number of occurrences of a specific term in the sentence. The feature matrix has a m * n dimension, where m is the number of sentences and n is the number of unique words in the corpus [53]. Bag-of-words have shortcomings when there are small changes in the terminology we are using as here we have sentences with similar meaning. In addition, the word sequence is ignore, and it is syntactic and semantic content [54]. Therefore, it be able to result in miss classification if the words are uses in different contexts.

2.6.2 N-gram

An extension of Bow is lead to the incoming of the bag of n-grams, which replaces the unit of interest in Bow from words to N connecting tokens [41].

A token is usually a word or a character in the text, giving rise to word n-gram and character n-gram models. Due to the connecting consideration, n-gram bags retain local spatial and order information [11]. An n-gram model forecasts the incidence of a word based on the occurrence of its N−1 previous word [11, 41]. For instance, a *bigram* model (N = 2) predicts the existence of a word given only its previous word (as N − 1 = 1 in this case). Similarly, a *trigram* model (N = 3) predicts the occurrence of a word based on its previous two words (as N − 1 = 2 in this case). The importance of terms in the classification known as by their weight [11]. The most widely use weighting techniques is term frequency, inverse document frequency and term frequency by inverse document frequency($TF-IDF$). Extract the n-gram features from the tweets and weight them according to their $TF-IDF$ values. The goal of using TF-IDF is to reduce the effect of less informative tokens that appear very frequently in the data corpus [49]. The disadvantages of N gram is it ignores syntactic and semantic content. Therefore, it can bring about miss classification when the words are practices in different contexts [3].

2.6.3 Neural Word Embedding

In recent years, deep learning methods have been used in text classification and automatic feature extraction. Word embedding is one of the methods used as an automatic feature extractor because it allows for the discovery of both semantic and syntactic relationships between words [16]. This allows for the capture of more refined attributes and contextual cues inherent in human language [2]. We use word2vec as a tool to represent distributed representations of words in a corpus when applying neural word embedding. There are two main learning sets in word2Vec: continuous bag-of-words and continuous skip-gram. Both the continuous skip-gram and skip gram, which consists of an an input layer, a projection layer, and an output layer to predict nearby words. let us see in the figure.

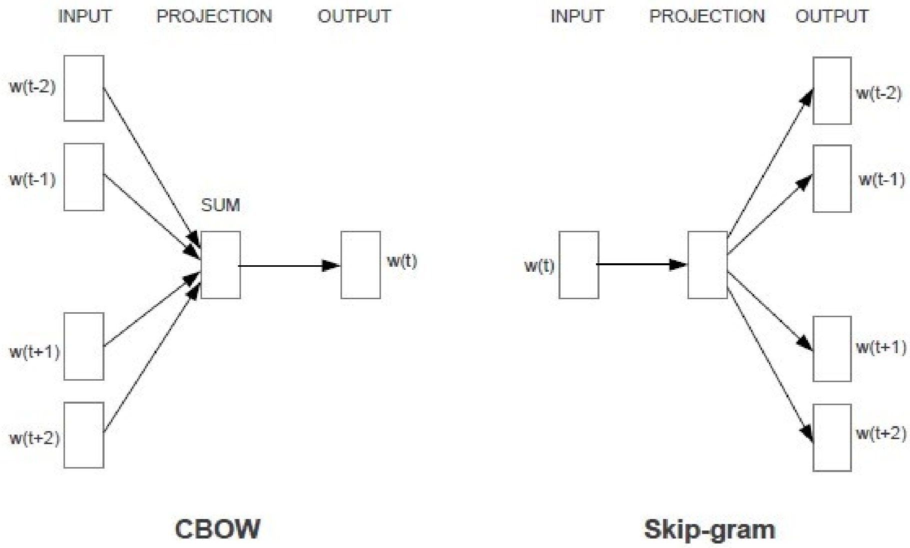

Figure 2.1: CBOW and the Skip-gram architecture (picture credit: ([9]).

Given a set of sentences then the model loops on the words of each sentence and either tries to use the current word w to predict its neighbors [34]. Continuous bows, predicts the present word grounded on the context, whereas skip gram predicts the neighboring words given the current word. The output is the probability distribution over all words in the vocabulary, which defines the likelihood of a word being selectness as the input word's context. The disadvantages of word embedding is it does not incorporating the morphological analysis in to word embedding considers the word itself only, not the n- gram char of the word or sub words. This makes the rare words due to misspelling and new word from corpus as word out of vocabulary (OOV), it leads to miss-classification of sentiment analysis

2.6.4 Neural Character N-Gram Embedding

Many natural language processing applications benefit from disseminated expression representations trained on large amounts of unlabeled sentiment.The morphology of words is ignored by well-known methods that learn word propagation representations [32]. Each individual word in the sentences is given its own vector representation. This is a significant constraint, especially for languages with vast vocabulary and numerous unusual terms, which we refer to as morphological complexity [9]. The Amharic language is one of the morphological complicated languages that may be encoded using neural fasttext embedding. There are two main learning sets of rules in neural fasttext embedding, which is an extension of neural word embedding. Even though the format is different, continuous bag-of-words and continuous skip-gram are similar to neural word embedding. Each word w is represented by an n-gram of characters. At the beginning and end of words, we use special boundary symbols and > to distinguish prefixes and suffixes from other character groupings. We additionally include the word w in the set of its n-grams in order to investigate a representation for each individual word (in addition to the n-grams). For example, Take the word “where” with n = 3, it will be represented by the character n-grams: <wh, whe, her, ere, re>.

$$G_{\omega}C(1 \cdots G)$$

the set of n-grams appearing in w [9]. We associate a vector representation zg to each n-gram g. We represent a word by the sum of the vector representations of its n-grams. We thus obtain the scoring function.

$$S(w,c) = \sum z_g^T vc$$

2.7 Deep Learning Classification Approaches

Deep learning, also referred to as hierarchical learning, is a type of machine learning algorithm that can learn layered model inputs [3]. It's a collection of algorithms and strategies based on how the human brain functions.The softness of the ways to apply certain extraction techniques manually and in combination with string labeling algorithms can be handled by deep neural network architectures [13]. In addition, text classification has be benefited from the deep learning architectures due to their potential to reach high accuracy with less need for engineered features [21, 3, 16]. The two main deep learning styles used in text grouping are convolutional neural networks (CNN) and recurrent neural networks (RNN) [58]. RNN is designees to process the sequential numeric data with hidden states that represent temporal information, whilst CNN learns the spatial information of an image [7].

2.7.1 Convolution Neural Networks

Convolutional neural networks (CNNs) form the pillar of multiple up-to-date computer vision systems. Nowadays, beyond object detection and image classification, there is positive effect on semantic segmentation and sentiment analysis tackled successfully by CNN [7, 1, 45].

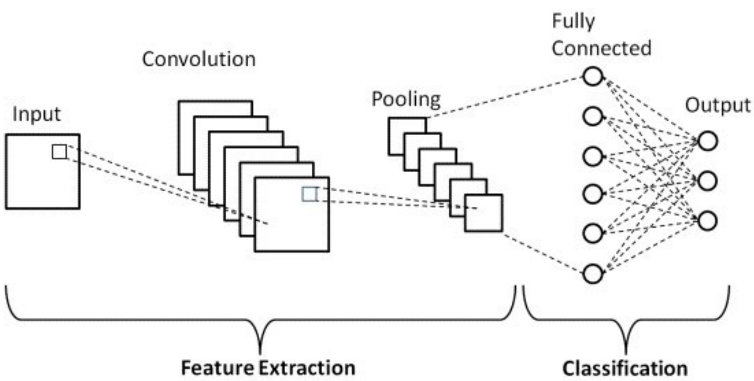

Figure 2.2: Symbolic representation of CNN (picture credit ([7]))

2.7.2 Recurrent Neural Networks

One of the benefits of RNNs is that the perception may be able to link previous data to the current task [7]. Recurrent Neural Networks use short-term memory when the character sequence on the sentences is too long.

Data from earlier time steps will be difficult to convey to those further ahead. The notations b and W represent the weight matrix for each time step. Moreover, t is the time and x, h and y, on the other hand, represent the input layer, hidden layer and output layer vectors.

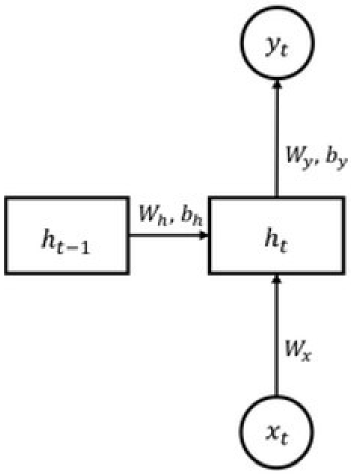

Figure 2.3: Symbolic representation of RNN (picture credit ([7])).

2.7.3 Long Short-Term Memory Networks (LSTM)

LSTMs are a sort of recurrent neural network that maintains text's long-term reliance [16, 1, 7, 13]. The cell state and its many gates are at the heart of LSTM. The cell state serves as a passageway for relative information to travel down the sequence chain. Gates are internal systems that can control the flow of information.

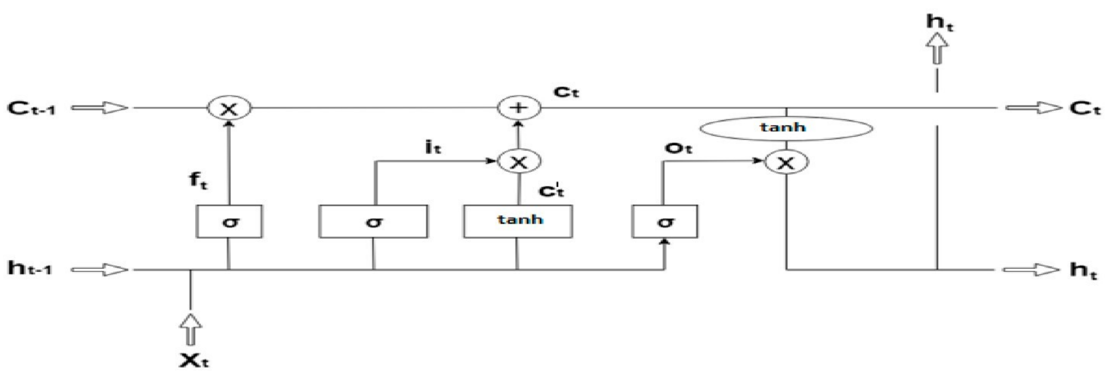

Figure 2.4: Symbolic representation of LSTM. (Picture credit ([7]))

As presented in Figure below, there are three different gates ft, It, ot. Where ft is forget gate, it is input gate and ot is the output gates. The top horizontal line in the repeating module of LSTM is its core part, which is call as a cell state.

2.7.4 Gated Recurrent Units (GRU)

The GRU is a newer generation of recurrent neural networks that is similar to the LSTM. GRU has escaped the cell state and is using the hidden state to transfer data. It also only has two gates, rt and Zt, which serve as the reset and update gates, respectively. Because GRUs have fewer tensor operations, they can be trained more quickly than LSTMs. Researchers and engineers typically test both to see which one works best for their application. In previous Amharic hate speech detection researcher [38] got good performance on GRU with word embedding.

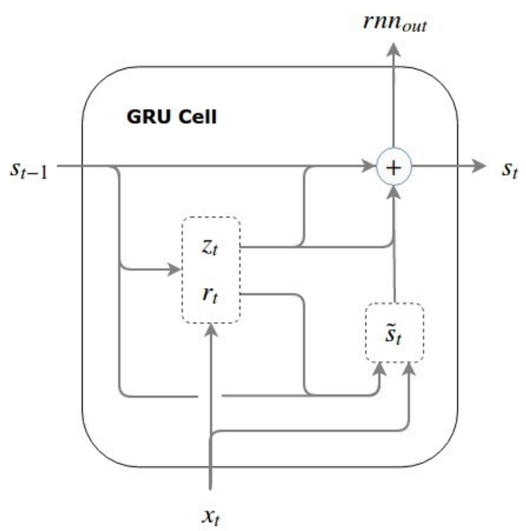

Figure 2.5: Symbolic representation of GRU (picture credit ([7])).

2.7.5 Bi-LSTM

LSTM and GRU allow the LSTM memory cell to store and access data over time. However, there are still limitations. On the one hand, single-directional LSTMs and GRUs do not take into account contextual information from future tokens. Bidirectional LSTM and GRU with word embedding perform well when dealing with sequential data that includes previous and future tokens as well as their context [13].

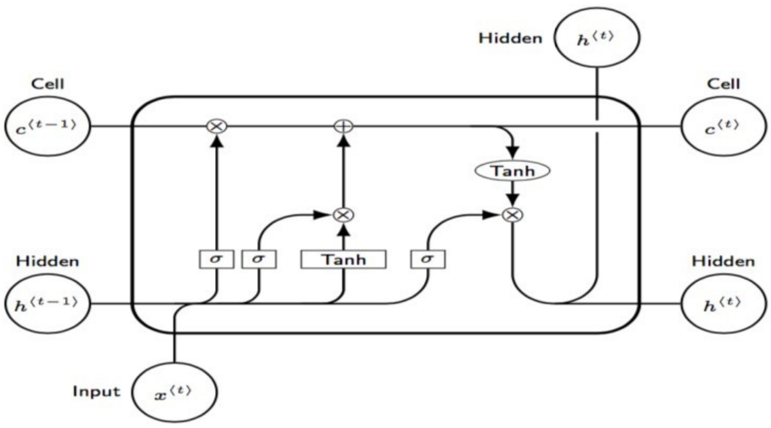

Figure 2.6: Bi-LSTM architecture (picture credit: ([13]))

2.7.6 CNN with Bi-LSTM

For each participation sentence, the Bidirectional LSTM generates a sequence of annotations (h1, h2,... hTx). Overall, the embedding of words h1, h2,..., etc., used in their work is primarily the concatenation of forward and backward hidden states entered into the coder. Only the last state of the encoder LSTM was used as the context vector in the simple encoder and decoder model.

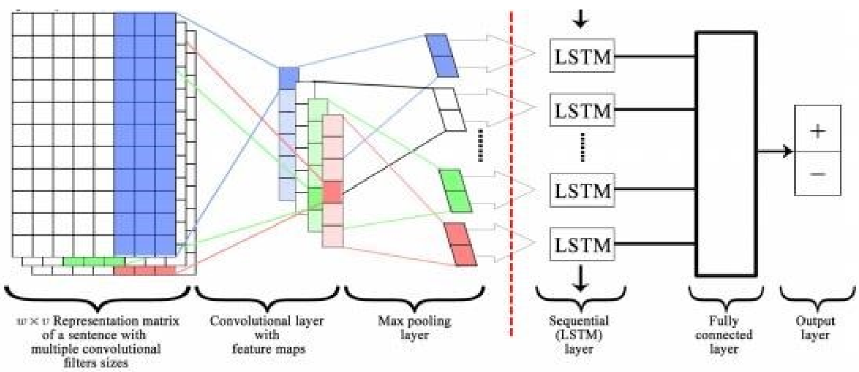

Figure 2.7: Combined CNN-LSTM Model architecture (picture credit: ([1]).

However, while creating the context vector by taking a weighted sum of the hidden states, it should place emphasis on embedding all of the words in the input.

Chapter 3

Literature Review

3.1 Background

This section reviews relevant literature to further understand the concept and investigate the difficult study, as well as provides a brief description of approaches and features extractors used in Amharic as well as foreign-language based sentiment analysis.

3.2 Sentiment Analysis in Foreign Languages

The author [21] worked on sentiment analysis of Arabic Tweets using feature selection via word embedding and deep learning. To predict the sentimentality of Arabic tweets, the author used 3,315 tweets from Twitter combined with convolution neural network (CNN) and long short-term memory (LSTM) models. The author model achieves an F1-score of 64.46 by utilizing these features.

Aswathi S and Lakshmi K. [47] worked on an improved approach for movie review analysis utilizing deep learning techniques and neural embedding with convolution layer acting as a feature extractor When compared to other traditional models, the author approach performed well on the IMDB movie review dataset, with an accuracy of 79.

Guixian Xu et al. [57] TF-IDF weighted word vectors are input into Bi-LSTM to capture context information effectively, and the comment vectors are better represented. The author introduces the Bi-LSTM structure, which can more effectively capture contextual semantic information. The proposed system by the authors yields an F1 score of around 92 percent.

Kai Zhou and Fei Long. [31] sentiment analysis of Chinese product reviews text using a convolutional neural network (CNN) in conjunction with a bidirectional long-short term memory network (Bi-LSTM). The author uses e-commerce product reviews as a dataset, and each of the 21,105 reviews is manually classified into two categories: positive emotion and negative emotion. The overall accuracy of 91.05 increased by 0.49 compared to the state of the art.

Mehmet U and Ilhan A. [48] a novel hybrid deep learning model for sentiment classification was developed. The author combined various neural word embeddings (word2Vec, fasttext, and character-level embedding) as features with various deep learning classification methods (LSTM, GRU, Bi-LSTM, CNN). It is critical to address the issues associated with OOV; we developed a character-level embedding model that can learn the embedding of character n-grams or parts of sub-words. The obtained results are F1 85.7 and accuracy 91.3. According to the author, this method is best suited for languages that are difficult to analyze morphologically, such as Turkish, Arabic, and Lithuanian. Furthermore, the proposed hybrid model can be improved by including attention mechanisms[48].

Karrabi, Mohammad, et al. [26] Sentiment Analysis of Informal Persian Texts Using Informal Word Embedding and an Attention-Based LSTM Network. The results show an increase in accuracy from 76.4 to 80.3 and an increase in f-score from 67.8 to 73.4. The results show that using word vectors from an informal text corpus is significantly more accurate than using formal word vectors. Furthermore, the results show that the attention model has a positive effect on Persian sentiment analysis. The author's approach is not simple, because instead of learning word representations, the author uses only the neural character embedding feature of the embedding layer as the context vector.

The paper [6] has examined effect of noise elimination and stemming in sentiment analysis such as data with tokenization, stop word removal only, addition of normalization only, and addition of stemming using (Othman) and Fatimah Algorithm. From the results, all datasets obtain small improvements after applying different types of pre-processing [6]. SVM classifier with data tokenization, stop word removal, normalization and stemming using Fatimah performs better than Othman algorithm. It also achieves the best performance for the malay documents classification.

The author [36] investigated that the contribution of text normalization techniques to sentiment analysis on informal texts. Therefore, in this paper we explore the results of applying lexical normalization applied to a sentiment analysis classification task on Web 2.0 texts, shows more than a 2.6 improvement over average F1 for the most informal data. However, not all the texts present the same level of informality and may not require additional preprocessing steps [36]. After the experimental results, this paper suggested that text normalization consistently helps to improve classification systems on the most informal genres and can be useful on some of the less informal texts.

The paper [24] showed that applying data processing approaches such as deleting stop words removal, normalization, stemming and lemmatization has a positive impact on unstructured and informal tweeter datasets. For instance, recurrent neural network applying stemming or lemmatizing, removing stop words and punctuation's improved the overall classification accuracy from 75.42 to 82.357. However, RNN without any transformation results (83 accuracy and 84 F score). This paper suggested Preprocessing process could improve by introducing novel algorithms and system for automatic recognition of non-standard word.

3.3 Sentiment Analysis in Amharic language

The study of sentiment analysis has grown in popularity in recent years, with the vast majority of studies focusing on foreign language content. However, only a few studies have been conducted in Amharic.

Selama.G. [17] developed a sentiment analysis model for the opinionated Amharic language using rule-based approaches. By assigning the first polarity weight to all selected opinion terms, the model classifies positive and negative opinion terms as well as contextual valence shifters such as negations. By employing manually crafted rules and lexicon, the author employed an Amharic blog feature level sentiment analysis model. To classify the opinionated text of 303 movie and newspaper reviews, a lexical of Amharic sentiment terms was used to identify and assign the initial polarity value to the detected opinion terms.

Wondwossen P and Wondwossen M. [44] worked on the nave bayes machine learning algorithm for classification, as well as uni gram, bi gram, and hybrid variants for feature extraction. The author gathered 608 posts from the websites of facebook, twitter, dire tube, and Ethiopian reporter and manually annotated the data by assigning polarity and opinion intensity scale values. To improve the system's performance, the author performed various preprocessing tasks such as data cleaning, data normalization, and stop word removal. With little training data, the author achieved promising performance accuracy of 43.6, 44.3, and 39.5 for uni gram, bi gram, and hybrid language models.

Musa S. [52] worked on a hybrid sentiment classification system for amharic book reviews, categorizing them as positive or negative. With 1500 features, the author used nave bayes machine learning algorithm and lexical approach, as well as the mutual information gain feature selection method.

The experiments are carried out with the help of 600 Amharic book reviews collected from various sources such as Facebook, personal blogs, and manually collected from individual book readers. The author has the highest accuracy of 93.33.

Yeshiwas G and Abebe A. [3] also worked on a deep learning approach for amharic sentiment analysis using count vectorizer and TF-IDF vector. The author used 600 documents, each of which contains a text file as well as Emoji. The manual labeled comments from facebook postings using language professionals in seven labeling classes: positive, very positive, extremely positive, neutral, negative, very negative, and extremely negative. The author received an accuracy score of 83.

Turegn F. [16] investigated the effect of preprocessing on LSTM-based sentiment analysis in Amharic with word embedding. Prior to data annotation, the stemming operation was performed; the morphological operation used makes automatic labeling difficult. Stemming is implemented after data annotation in this paper because the labeled lexicons are not stemmed. To demonstrate the effect of stemming, the authors used LSTM and MNB test models with 1077 comments. The results show that using stemming reduces the accuracy of the model. When using long short-term memory-based sentiment analysis, the model's accuracy decreases by 6.43 and by 0.43 when using bi gram multinomial naive.

Preprocessing can have a positive or negative impact depending on the type of datasets used, the language it is written in, and the preprocessing method used algorithm [16, 36]. However, this paper does not analyze normalization; rather, it uses it to remove difficulty at the classification stages. The author achieved state-of-the-art performance with 90.7 accuracy. Instead of learning word representations, the author takes a more difficult approach that ignores word internal structure and morphology, and only the last state of the encoder LSTM.

3.4 Summary

In section, we are attempting to describe the state of the art research efforts and point out what are they missing form doing in the table below.

Table 3.1: Sentiment analysis for the Amharic language: a summary of research

Author	Title	Features	Model	Domain	Dataset	Limitation
Selama [17]	Sentiment Mining Model For Amharic Text	lexicon	Rule Based	Movie reviews	303	Time consuming for generating rules
wondwosen [44]	Machine Learning Approach to Multi-Scale SA	N-gram	NB	Product marketing and news	608	Did not handle context features
Tadesse [2]	Sentence level Amharic SA	lexicon	machine learning	Movie comments	480	Indirect opinions cannot be classified
Mekonne [52]	hybrid approach of Amharic SA for kana tv	rule	Dictionary	Kana tv	1022	not considered the concept of sentence
Yeshiwas [3]	Deep learning approach for Amharic SA	TFIDF	ANN	Politics	1600	didn't capture context-based and sequential feature
Turegn.F [16]	effect of stemming in Amharic sentiment analysis	Word em-beddi ng	Deep learning	Movie review	9138	Didn't consider Morphology of the word

Chapter 4

Proposed Methodology

4.1 Background

We will design and explain the proposed methodology structure for Amharic informal text based sentiment analysis in this chapter. The structure consists of five fundamental processes. These are data collection, preprocessing, feature extraction techniques, model training, and model testing.

4.2 Overall structure of proposed methodology

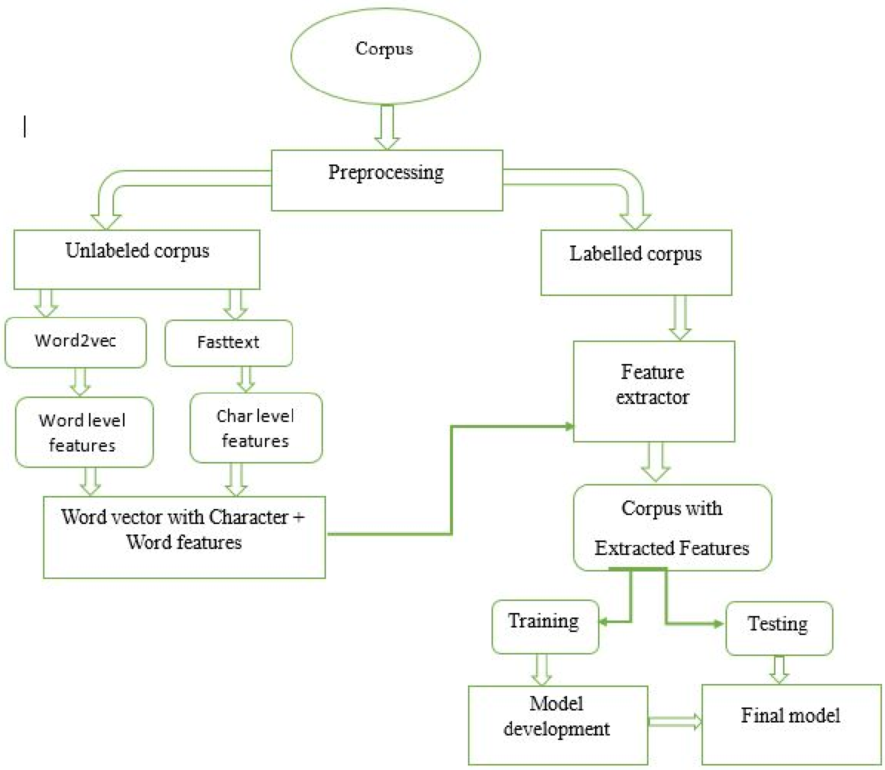

Figure 4.1: Overall structure of Proposed Methodology for Amharic Sentiment analysis.

4.3 Preprocessing

The corpus that has been collected and annotated should first be converted to a representation that will allow the data to be used more effectively and increase the sentiment analysis's performance. Because the data is gathered from social media, the text is in an unstructured format. Therefore, the text must be processed and represented in a concise and identifiable format or structure. We made use of the following data cleaning, tokenization, normalization, and stop-word removal are all part of the preprocessing steps. However, data preprocessing is beneficial for sentiment analysis, but it has less of an impact on Amharic informal text. As a result, we must use two types of data processing for automatically labeled data and manually labeled data. Because manually labeled data necessitates more thorough preprocessing than informal text of actual data that has not been labeled manually.

4.3.1 Data Cleaning

We gathered the data from facebook social media, which is very unstructured, so we started by removing non-standard words like an empty space, null value non-word characters, HTML, URLs, and non-Amharic texts and numbers.

Input: text in a dataset

Output: clean text

Begin:

I. Read the text in the dataset.

II. While (not end of the text in a dataset):

If the text contains special char [,’! @#∧*]

then Remove special char

If the text contains symbol [<>‹›« »=:-`~_/]

then Replace symbol and add space

If a text contains geez number [፩ ፪ ፫ ፬ ፭ ፮ ፯ ፰ ፱ ፲ ፳ ፴ ፵ ፶ ፷ ፸ ፹ ፺ ፻] t

then Remove geez number

If a text contains Amharic Punctuation ='[፧ ። ፡ ፣ , • ፨] then Remove Amharic Punctuation

If text contain English word and number = [a-z A-Z] [0-9]

then Remove English word and number

If text contain number = [0-9] then

Remove number

If a text contains extra white space then

Trim the text

III. Return clean text.

End:

4.3.2 Tokenization

Tokenization is the process of dividing a string into pieces, such as words, keywords, phrases, representations, and other elements known as tokens. It can be a single word, a phrase, or even an entire sentence. Some characters, such as punctuation marks, are removed during tokenization. Tokens are used as input for various applications, such as parsing and text mining. In this step, input texts are tokenism into a stream of lettering using white spaces, which aids in the conversion to a list of words.

4.3.3 Normalization

Normalization is one of the data preprocessing steps performed to cleaning or removing irrelevant data from a huge collection of extracted data to make it more clear and consistent [36]. The writing style used in social media usually contains informal elements that can lower the performance of natural language processing applications. For this reason, text normalization techniques have drawn a lot of attention recently when dealing with informal content. Normalization is one of the techniques used to improve the performance of natural language processing applications in general and sentiment analysis in specific [30].

Some researches use normalization as one of the preprocessing steps while others argue that normalization does not have demonstrative impact on text classification tasks. Many research are done on Normalization to show whether it positively or negatively affect NLP applications. In Amharic writing system there are characters with the similar sound have different symbols so called homophone characters. For example, the character **‘ሠ’** and **‘ሰ’** can used interchangeably as **“ሠው”** and **“ሰው”** to mean “man”. These different characters must considering as similar because they do not change the meaning.

Inconsistency of terms may cause increasing representation of words that causes huge data size processing and regard as the word like a different term. In this work, such a kind of inconsistency in writing words will be handles by changing characters of similar sound to their common standard. In addition, there is a requisite for normalization of words to handle the inconsistencies nature of informal word contained sentences. For instance, **አረረረረረረረረረ**, **አረአረአረአረአአረአረ**, and **አረረረረረረረረረ** with **አረ**. Understanding it is challenging since many informal expressions with numerous spelling errors, abbreviation and slang are used.

Therefore, a crucial task is to preprocessing corpus to reduce text's ambiguities. It helps to reduce the tweets' representation space and to increase the similarity between two similar texts written in two different ways [29]. In addition, converting all abbreviations, slang to their meaning (example) **“ደቅ”** to **“ቁጭ”**). To get the meaning of abbreviations, slang, a dictionary manually constructed from Amharic slang dictionary and other corpus. In this thesis we used 200 words and 35 characters to normalized 3120 number of sentences from 5000 informal contains sentences.

4.3.4 Stop Word Removal

In classification, not all words in a sentence have the same weight. Some words are used to complete a sentence's grammatical structure. As a result, they do not denote an object or concept. The common words in the text carry less weight. Stop words are used in Preprocessing. To reduce the space and time complexity of processing, a list of non-content bearing terms should be compiled. We manually prepared 130 words of stop word lists from our data set for this thesis. We find terms that have no effect on the meaning of the sentence and terms that are irrelevant to representing the sentence. For example, Amharic words such as , ኋላ, ሁኔታ, ሆነ, ሆኑ, ሆኖም, ሁል, ላይ, ሌላ, ሌሎች, መሆኑ, ማለት, ማለቱ, መካከል, የሚገኙ, የሚገኝ, ማድረግ, ማን.

4.4 Feature Extraction

The most important task in text classification is the selection and design of the feature extractor after preprocessed data. In the Amharic language, various feature extraction techniques were used, including handcrafted based, char-n-gram, tf-idf, and word embedding. Various features have been implemented and tested in our system as part of this research. In order to select the most convenient features for Amharic informal text sentiment classification.

4.4.1 Neural Word Embedding

The method presented in word2vec is one of the most well-known methods of word embedding algorithms for preserving the meaning of words in vector representations [26]. We used the word2vec pre trained word embedding to learn individual words with their embedding vector from our collected data. Train the sentiment using pre-trained word embedding with initializing weight as input to the lstm keras embedding layer. This is a more gradual approach. However, it adapts the model to a specific (manually labeled) training dataset. The embedding layer is pre-programmed with random weights and learns an embedding of all the words in the training dataset. It can be used as part of a deep learning model in which the embedding is learned alongside the model.

Many languages have successfully trained neural word embedding [33]. While training word-level models with such informal text can lead to words being captured that have the same meanings, these models cannot capture all words that can be encountered in the real world due to out-of-vocabulary (OOV) words [4]. As a result, there is a problem with word recognition of nearby words, which leads to the creation of various vectors [26]. This research focuses on developing embedding models for Amharic (Ethiopia), a Semitic language that is both morphological rich and written as an alpha syllabify (abugida) rather than an alphabet[33].

4.4.2 Neural Character N-Gram Embedding

Along with the original target word, neural character n-gram embedding added a simple sub-word model on top of the skip gram. That have their own vectoral representations that are added together to form a word. for example, for the word **አይመለስም** with n = 3, the set of vectors would include <**ለስም**, **መለስ**, **ይመለ**, **አይመ**>).

In a word similarity task, this sub word model improved embedding for the morphological rich languages of Arabic, German, and Russian [33]. Other research on morphological rich languages has used sub-word models to automate tasks such as the handling of informal words. The fundamental distinction between Vectors derived from formal texts cannot be used in informal contexts due to the distinction between formal and informal texts. Social networking services, for example, do not produce desirable accuracy. [26]. To overcome this drawback, the author [26] provide a large integrated text corpus of several different sources of informal comments and we utilize the neural character n-gram embedding as the word-embedding algorithm. On the way to learn, morphological features establish in each word, we used a character n-grams model using fasttext a tools [56]. Character n-grams differ from word embedding in that they can learn vectors of character n-grams or parts of sub-words. This feature enables the model on the road to capture words with similar meanings but different morphological term establishments. It includes free pre-trained word vectors designed for 294 languages and trained on wikipedia datasets. Amharic is also one of the 294 languages that have been chosen. However, because wikipedia and social media use different text types, these word vectors are ineffective for our task. As a result, we developed our own model with 100 dimensions for training. To implement the fasttext, we used the genism Library 8.

4.4.3 Combination of Neural Word and Character N- Gram Embedding as A features.

The features we propose are a combination of Neural Word and Character N-Gram Embedding. The word embedding tools we used for the word representation models have been applied successfully to a variety of natural language processing tasks, including sentiment analysis [4]. These models, however, do not always work well in certain social media contexts. It is important to note that a wide range of linguistic domains are involved. This is primarily due to the fact that social media users use a variety of informality in their communications. While training word-level models with such informal text can result in words with similar meanings being captured, these models cannot capture all words encountered in the real world due to out-of-vocabulary (OOV) words.

One of the main limitations of this word level model is its inability to identify words. Character-level embedding, on the other hand, can deal with this issue effectively due to their ability to learn. The author [4] used both character- and word-level models to determine the best ways to represent Arabic words in tweets.

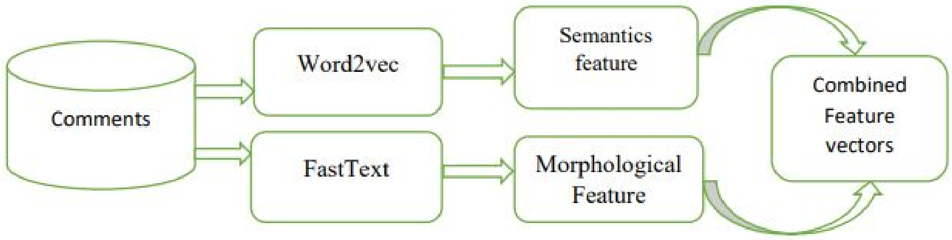

Figure 4.2: Combined feature extraction flow diagram

Concatenated inputs are combined to create an embedding matrix, which is a list of all words and their corresponding embedding, using two well-known pre-trained tools: fasttext for character n-grams and word2vec for word-level embedding. They are then used to initialize the keras LSTM embedding layer's weights.

The weights of the keras LSTM embedding layer were updated during training to fine-tune the model development. If not, the keras lstm embedding layer learns an embedding for each word in the training datasets.

4.5 Deep neural network model development

The steps we took to create deep neural networks, LSTM, and CNN-LSTM, an improved version of the recurrent neural network. It is capable of tackling a variety of problems and providing robust solutions, such as the vanishing gradient problem.

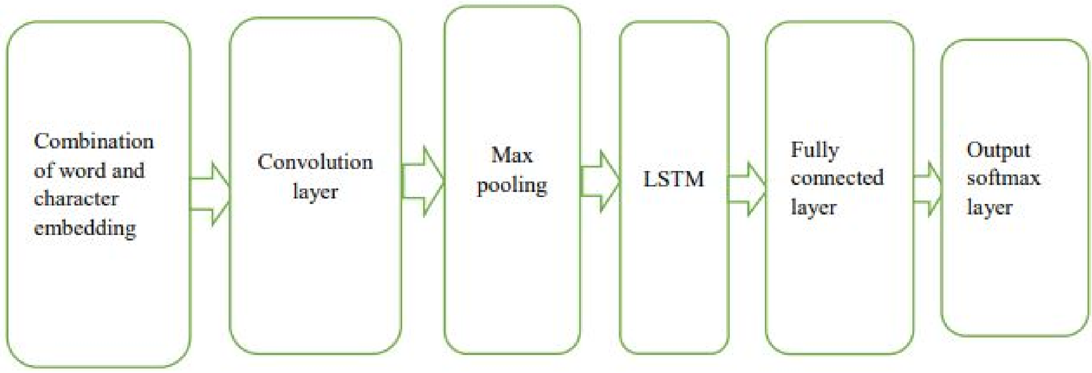

Figure 4.3: CNN-LSTM model with Word and character n-gram embedding

The LSTM's central structure is made up of four layers that interact with one another. The Forget Gate, Input Gate, Modulation Gate, and Output Gate are the four layers. We combined all of the word embedding models (see section 4.4.3) to improve performance, and these were then used as prepared weights in the keras embedding layer.

4.6 Model evaluation

The learning ability of different deep neural learning models trained on the Amharic informal containing text datasets was investigated and evaluated. The datasets gathered from various social media platforms would be fed into the proposed models, as well as various performance evaluation metrics appropriate for models.

TP (True Positive): The number of occurrences that are positive and are correctly predicted as positive.

FP (False Positive): The number of instances that are negative but are incorrectly predicted as positive.

FN (False Negative): The number of instances that are positive but are incorrectly predicted as negative.

TN (True Negative): The number instances that are negative and truly predicted as negative.

4.6.1 Precision

It is the percentage of the number of items labeled as a particular desired class, true positives (TP) to a total number of items labeled as that class.

$$p = \frac{Tp}{Tp+Fp}$$

4.6.2 Recall

It is the total number of true positives distributed by total number of items that are knowns to belong to that class.

$$p = \frac{Tp}{Tp+FN}$$

4.6.3 F-score

It is the weighted average of Precision and Recall. Therefore, this mark taking in cooperation with false positives and negatives into account.

$$fscore = \frac{2xRxp}{R+p}$$

F score is a better and well-known measure to use if we need a balance between precision and Recall.

4.6.4 Accuracy

The additional performance measure of a model is accuracy. Accuracy measures how much accurately the model learns to classify the data.

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

Chapter 5

Experimental Analysis

5.1 Background

This chapter presents the experimental results of the proposed solution for sentiment analysis in Amharic informal words, as implemented using the proposed methodology discussed in chapter four. This study employs an investigation approach to answer research questions using informal opinions such as feature extraction and deep neural learning classification algorithms. We carried out a variety of experiments, which we divided into two categories. The first experiment in the group demonstrated the impact of informal opinion as a feature for automatic labeling. The second experiment's results show the effect of normalization on Amharic informal words and their combinations.

5.2 Data Collection

In this study, social media platforms such as Facebook and YouTube are used to create textual-based training, validation, and testing datasets. We collected Amharic informal and formal textual data from social media using face pager and the youTube API of web scribing tools, which previous researchers and most Amharic informal text users had publicly posted. Face pager is a facebook crawler that uses the graph API to select the content of facebook comments and posted opinions. Because there aren't enough published corpora for Amharic text, a new corpus is required. We are the first researchers to conduct sentiment analysis on Amharic informal.

5.2.1 Dateset Annotation

Annotation is a method of categorizing collected data opinions into different polarity classes [40]. The annotators perform manual categorization based on their Amharic language skills and the annotation requirements. The annotation was completed in accordance with the researcher's standards.

5.2.2 Data Annotation Guidelines

The framework or guidance instructions provided by the researcher to the annotator to facilitate the annotation process are referred to as "annotation guidelines" [12]. The guidelines instruct us on how to categorize annotators' opinions into three categories: positive, negative, and neutral. The instruction focuses on the classification of positive, negative, and neutral sentiments and is organized by reviewing various sentiment analyses. For high-quality annotations, clear and simple instructions are essential [35].

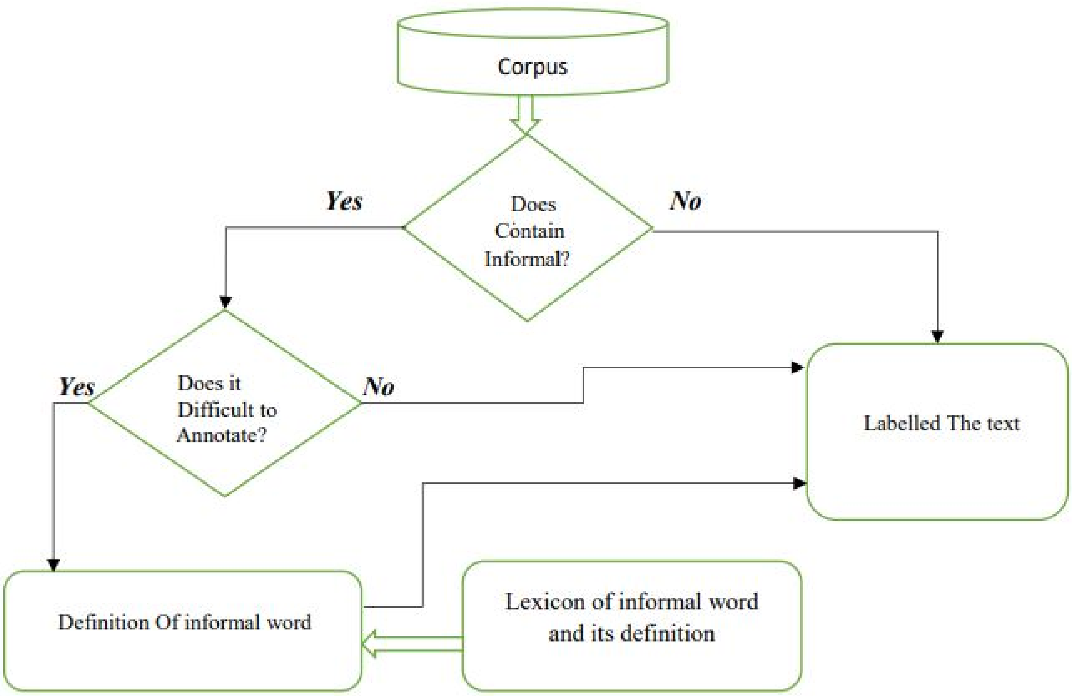

Figure 5.1: The overall structure of annotation guidelines

5.3 Datasets Description

We collected posts and comments from facebook and youtube using the content retrieval tools face pager and youtube online comment scraper to create the datasets. Furthermore, we created a publicly available datasets of informal words containing comments written in Amharic using manually and artificially generated data from previous researchers.The total manual and automatic labeled datasets contains 10,653 comments written in a mix of formal and informal text, as well as a large corpus. The datasets is described in detail in Table 5.3. The data includes the total number of sentences and words, as well as the maximum comment length.

Table 5.1: Datasets Description

Total number of corpus	Type of data	Level of sentiment
10,000	Automatically labeled	sentences
2,000	Manually labeled	sentences
21,230	unlabeled	sentences

5.4 Implement Environment

When you first start learning a new programming language, you spend a lot of time and effort getting to know the syntax, code style, and built-in tooling. This is true for Python as well as any other language. Once you've gained enough familiarity with python to feel comfortable with its intricacies. To implement the proposed solution, this study made use of an efficient Python development environment, including tools, packages, and an experimental setup.

5.4.1 Development Tools and packages

In this section, we'll go over the tools, as well as their descriptions.

Table 5.2: Tools and its description

Tools	Description
Python	Powerful programming language easy to learn and develop a machine learning application.
Anaconda Navigator	Allows us to launch development applications and easily manage conda packages, environments.
Jupyter notebooks	Open-source network application that permits us to produce and share documents that hold live code, equations, conceptions, and narrative text.
Pandas	High-performance, easy-to-use data structures, and data analysis tools.
Keras	A library for developing deep neural learning networks in python
Tensor Flow	Open-source platform that has flexible tools and resources for machine learning. This study uses it Tensor Flow as a backend.
NLTK	NLTK is a platform that used for constructing deep neural learning programs to work with natural language data. This study uses it for tokenization.
Scikit-learn	A set of python modules uses it for feature extraction, training, and testing model
NumPy	Array processing for number, strings, and objects. This study uses it for handling converting the text to numeric
Genism	Python library for topic modeling document indexing and similarity retrieval with large corpora. This study uses it for the constricting word2vec and fasttext model.
Matplotlib	Publication quality figures in python. This study uses it for data and results visualization.
Flask	Micro framework for making web services in python. This study uses it for implementing the prototype for selected models.
Notepad++	Several languages are supported by this source code editor and Notepad replacement.
Texstudio	TeXstudio is a LaTeX document creation environment with an integrated writing environment. Our goal is to make LaTeX writing as simple and enjoyable as possible.

5.4.2 Experimental Setup

This study uses one personal computer for deployed the developmental tools and packages, which discussed in the above section. Table 5.3 shows the hardware and software description of the hard ware and software requirement used in all computer to do an effective experiment.

Table 5.3: Experimental setup

Model	HP Pro Book 6470b
Processor	Intel® Core™ i5-4310M @2.70GHz,2601MHz, 2 Cores,8 GB of physical memory
Hard disk	1terabyte
Memory	4GB
Operating System	Windows 10 Pro, 64 bits

5.4.3 Model Parameters setting

In addition to hardware and software tools and their specifications, we must also configure the deep learning network parameters for developing sentiment analysis models. The model parameters define or represent a model in deep learning [20].However, training a model entails selecting the optimal parameters that will be used by the learning algorithm to learn the optimal parameters that correctly map the input features to the labels or targets. The CNN-LSTM network was configured with the following parameters in this experiment: One LSTM layer with 8 neurons, 10 epoch, 14 batch, 8 vocabulary size (embedding size), 'adam' optimizer, 7 seed, 90 train size, and 10 test size.

Dropout number is used to select an appropriate epoch. Because detecting the occurrence of over fitting in the neural network during training is critical,Over fitting occurs when a model learns training datasets so well that it generates noise about the training data to be memorized. As a result, it is unable to predict an output for an input that has never been seen before [16].

Over fitting is a term used to describe a model that models the training data too well. A model that cannot model the training data or generalize to new data is said to be under fit. An under fit machine learning model is unsuitable and will be obvious due to poor performance on training data. under fitting occurs more frequently on small datasets than on large datasets. Even though our datasets are larger than the state of the art, they are small in comparison to what deep neural networks require, so techniques such as optimizing the number of epochs and dropout may be required.

Figure 5.2: Accuracy and Loss for train-vs-validation

According to the graph, if the model is trained beyond epoch 5, the model's training accuracy increases while its validation accuracy decreases. As a result, we can say that the model is over fitted, and our epoch number will be set to five to observe the effect of over fitting.

5.5 Result and Discussions

5.5.1 Effects on utilization of the Amharic informal lexicon as a features in Amharic sentiment analysis.

The amount of variation in the datasets used in sentiment analysis can affect the results [16]. We conducted an experiment in which we tracked the volume of datasets before and after the use and normalization of Amharic informal words. Previously, in order to answer the first research question, we investigated the impact of using Amharic informal opinion as features for automatic labeling on Amharic sentiment analysis. We ran an exploratory experiment to compare the sentiment classifier performance of a labeled informal lexicon with formal lexicons and emojis on the datasets. The graph depicts how using a labeled informal lexicon for comment labeling affects data distribution.

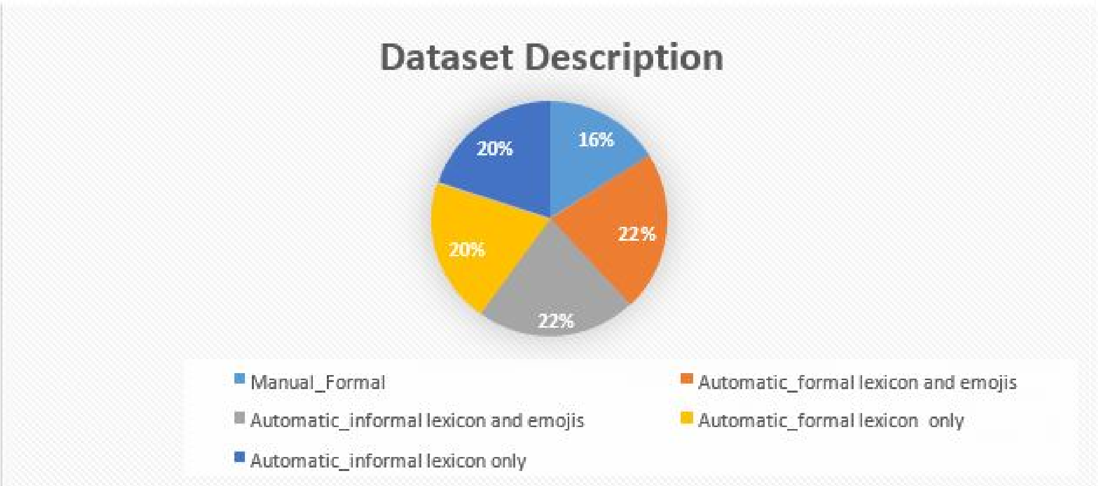

Figure 5.3: Distributions of manual labeled and automatically labeled comments

From the total number of collected comments, 22 percent use informal language and emojis, while the remaining 20 percent do not. This demonstrates that a significant number of comments express their opinions using informal language or a combination of informal and emoji. Informal and

emoji expressions are regarded as reliable indicators of emotion. They are used to either generate more training data automatically or to improve sentiment classification. The use of labeled Emojis for labeling user-generated data affects the number of positive, negative, and neutral comments [16].
A use of informal and Emoji improve the impressibility of a text message. For instance, the word **”የዛሬው ፈዋ ጅንስ ነው”** difficult to labeled as positive or negative but "**አረረረረረረ የዛሬው ፈዋ ጅንስ ነው”** is classified as negative because it expresses negative feeling towards the food (**ፈዋ**). In the same way the texts like,**ምን አባቴ ላድርግህ** are difficult to determine their sentiment class,and the comment **“ምን አባቴ ላድርግህ ኡፍፍፍፍፍፍ”** cannot labeled as either positive or negative. However, **ምን አባቴ ላድርግህ ኡፍፍፍፍፍፍፍ** can labelled as positive. Comments that contain only informal like**” ኡፍፍፍፍፍፍፍ”.** and cannot be labeled unless the labeling system combine list of positive and negative informal lexicon and emoji with labeled formal lexicon words.
The results show that when we use both labeled formal lexicons, emoji, and informal lexicons, we can label a total of 13,010 comments. However, when only labeled formal lexicons are used, 5000 comments are labeled. The number of labeled comments using the formal lexicon approach is 5500 less than the combined formal, informal, and emoji-based approaches. Unless we include a labeled informal lexicon in our comment labeling system, the remaining 8,010 comments will be unlabeled. This demonstrates that informal lexicons play an important role in automatic labeling systems.
Recent studies have focused not only on the unequal distribution of classes, but also on the difficulty in incorporating the informality nature of the workforce [16]. The first method is up sampling, which transforms the data by either creating new data for a minority datasets or replacing it. The other method is to remove some data from the majority datasets, which is known as down sampling.

However, this frequently results in the removal of important samples or the introduction of insignificant new objects [16]. The final approach, which is gaining popularity, is a hybrid level that combines the benefits of the first two approaches [16].

The datasets used in this study have an unbalanced distribution of positive, negative, and neutral classes. The paper chose random oversampling of the minority class from 13,010 to 15,228 to overcome datasets imbalance. Under majority sampling, class is not the best option because further removing from a small datasets is risky [16]. The purpose of this study is to look into the impact of using a labeled informal lexicon for comment labeling. The table below shows the sentiment analysis performance results obtained with and without the usage of the informal lexicon for comment labeling.

Table 5.4: Performance of Bi-LSTM with informal lexicon as a feature on 10 and 90

	Performance metrics							
	Informal opinionative features				Formal opinionative features			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
neutral	88.36	87.42	87.42	88.36	87.42	88.37	88.32	87.42
positive	82.72	86.48	84.6		88.4	80	83.66	
negative	90.24	88.36	89.3		87.5	90.24	89.4	
Macro averag	87.1	87.4	87.42		87.34	86.48	87.3	

Table 5.5: Performance of CNN-Bi-LSTM using informal lexicon as a feature on 10 and 90

	Performance metrics							
	Informal opinionative features				Formal opinionative features			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
neutral	93.12	84.5	88.34	89.28	82.7	94.08	87.36	87.36
positive	90.25	85.6	87.32		92.16	84	85.45	
negative	85.5	94.1	89.28		93.2	87.4	90.3	
Macro averag	89.3	88.32	88.35		89.25	87.4	87.38	

We can see from the table above that the Bi-LSTM-based approach is a classification-based method because the forward-backward encapsulate contextual information during various Amharic-based-target learning stages. In general, we conducted additional experiments with two other deep learning methods in terms of specific training and testing dataset proportionality. We compare our Bi-LSTM results to those of other well-known deep learning models, such as the convolutional neural network (CNN) and the Long Short-Term Memory (LSTM) network. CNNs should have convolving filters over each input layer to generate the best features, and it has been proven that CNN is a powerful tool for selecting feats.

Table 5.6: Performance of Bi-LSTM using informal lexicon as a feature on 20 and 80

	Performance metrics							
	Informal opinionative features				Formal opinionative features			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
neutral	82	87.4	84.7	86.17	83.7	87.5	85.6	85.7
positive	87.5	80	83.67		84.6	82	82.8	
negative	87.8	85.56	86.5		87.45	85.6	86.5	
Macro averag	85.6	84.6	84.3		84.7	84.6	84.8	

Table 5.7: Performance of CNN-Bi-LSTM using informal lexicon as a feature on 20 and 80

	Performance metrics							
	Informal opinionative features				Formal opinionative features			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
neutral	82.6	90.3	86.4	87.4	92.2	78	84.6	85.5
positive	87.4	80.1	84.5		81.7	86.5	84.48	
negative	90.3	85.44	87.4		82.6	91.2	86.5	
Macro averag	86.7	85.5	86.1		85.5	85.44	85.5	

In general, the experiments conducted to answer RQ1 show that the use of informal opinionated as a feature in both Bi-LSTM and CNN-Bi-LSTM based automatic text classification approaches is also gives convincing superior performance result. For Bi-LSTM, the use of informal lexicon with emoji's and formal lexicons have improved the accuracy and precision values of the automatic labeling while for CNN with Bi-LSTM values of the precision measures are equal with emoji's and formal lexicon.

This convincing improvement could be due to nature of the algorithms used for labeling and the proportions of training and testing datasets.

Therefore, this increment or improvement on accuracy due to utilizing informal opinionated features to handle the morphological importance of Amharic informal text.

5.5.2 Effects on Normalization and Embedding of the Amharic Informal Words as a Feature in Sentiment Analysis.

In this section, we investigate the effect of normalizing Amharic informal words on the performance of Amharic sentiment analysis. The table shows examples of the structured or normalized word and the formal form of the informal words while using dictionary-based normalization. To demonstrate the effect of normalizing, we conducted sentiment analysis using normalized and non-normalized datasets without varying the polarity of class on non-normalized datasets.

Normalizing means changes informal words to their formal form that results in a reduced number of features or vocabulary size. Sometimes the formal form of informal words may lost the morphological nature and semantical meaning of the word in the sentences. For instance, from the sentences **“የዛሬዉ ስራ መላው አይነፋም”** the word” **መላ”** can normalized with **“ብር”** Nevertheless, after normalization it gives **“የዛሬዉ ስራ ብርው አይነፋም”** it gives complex understanding on semantically features based only. Changing all informal forms of words to a formal form results in weak dependence between in the semantically and morphological features.

Because, the word **ብር** and **ብርው** are different features or vector representation on the morphological relationship between this two words. In addition, the informal words [**ቀቀቀቀወዉዉዉ**, **ቀቀቀቀቀቀቀወ**, **ቀውውውው**] can normalized to **“ቀው”** but it is not possible to write all the possible combination of morphological related word especially in complex languages like Amharic. Due to it takes too much time and it needs Amharic language qualified human labor force. Therefore, we need the mechanism to use the benefit of both normalization and incorporate the direct morphology of Amharic informal words to utilize the feature of Amharic informal words in sentiment analysis [24].

We used the fasttext char n-gram embedding automatic feature extractor works on sliding on variable n-gram character of a word to hold the morphological information into embedding space [26]. Later the following table's shows that the result record on impact of normalizing and embedding informal words in sentiment analysis on different classification algorithm's methods.

Table 5.8: Performance of Bi-LSTM before and after normalized Amharic informal word

	Performance metrics							
	Before normalized				After normalized			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
No Embed	85	84	83	84	86	85	86	85
fastText	88	87	87	89	86.63	86.3	87	88.28
word2vec	87	86	85	85	88.39	88.2	87.25	88.38
word2vec and fasttext	89	86	88	87	89.63	86.7	89.26	88.3

Table 5.9: Performance of CNN with Bi-LSTM before and after normalized informal word

	Performance metrics							
	Before normalized				After normalized			
	Precision	Recall	Fscore	Accuracy	Precision	Recall	Fscore	Accuracy
No Embed	85	86	87	85	86	87	88	86
fastText	87	90	89	87.81	87.89	89.28	88.38	89
word2vec	87	87	87	87.88	89.13	89.9	90.35	89
word2vec and fasttext	88	87	87	88.05	90	89	90	90

Tables 5.7 and 5.8 show that after normalizing, the accuracy of sentiment analysis on Bi-LSTM and CNN with BiLSTM is higher than before normalizing by an average of 3.38, 1.83 on Bi-LSTM with word2vec, word2vec, and fasttext feature extractor models, respectively, and an additional enhancement of accuracy of 1.19, 1.12 on CNN with BiLSTM for Word2vec and fasttext feature extractor models. Normalizing yields the same F-score for fasttext automatic feature extractor implementation as before normalizing on Bi-LSTM. However, Bi-LSTM precision and accuracy were reduced from 88 to 86.63 and 89 to 88.28, respectively. Furthermore, CNN with Bi-LSTM recall and F score on fasttext.

The most recent papers not only address the gap between two automatic feature extractor mechanisms by combining these two different types of word embedding, but also the difficulty of incorporating the informality nature of the data due to its significance into a supervised learning framework for the task of sentiment analysis of Amharic informal text. As a result, the learning algorithm will have to create words from the vocabulary in order to create the word vector representation. By incorporating the informality of the data, there are two ways to word out the vocabulary. The first approach is to use a large amount of data to train an automatic feature extractor [37].

The first method is insufficient for languages with limited resources, such as Amharic. Because of the morphological nature of different data types, the language type of datasets used in sentiment analysis has an effect on the result. We chose the second method because it is best suited to low-resource languages with complex morphology. Amharic is a morphological complex language as well. However, in highly resourced and morphological simpler languages like English, the decreasing frequency of the word may reduce performance [28].

Due to the curse of dimensional of highly infrequent comments and posts, which can sometimes prevent the classifier from correctly learning how to assign a sentiment label to unseen text [28]. Because of the curse of dimensional, infrequent comments and posts can sometimes prevent the classifier from correctly learning how to assign a sentiment label to unseen text. As a result, after normalizing Amharic informal words in sentiment analysis, we want to compute the performance metrics of our model in different embedding as a feature using the fewest possible frequency of words parameters.

Because of its flexible nature of production, such as shorthand, elongated, and slang, informal text may appear less frequently. As a result, reducing the frequency of word occurrences will assist us in producing less frequently informal text. For example, short comments, which are one type of less frequently informal text [37]. It increase the word vocabulary size by minimizing the word out of vocabulary problems. We observed that when we increased minimum word frequency from total unstructured corpus, the number of informal words in the data may increases, which causes decrease in classification accuracy especially for word embedding based automatic feature extractor. This might be relates with withdraw of the precise short and less frequent opinions by the feature extraction methods [37].

As a result, we can say that the less frequently occurring word may contain informal text and will arise from vocabulary. While decreasing the minimum count of the word on word embedding may improve the word's out of vocabulary problems or vocabulary size, the overall correctness of the model may decrease, particularly on word2vec based automatic feature extractors. Because semantic word embedding similarity measures necessitate a higher minimum word count to determine the semantic relationship between the words,[14]. As a result, we need to conduct an experiment to determine the optimal number of minimum word frequency parameters with varying word embedding.

Table 5.10: The effect of the number of minimum word frequency parameters on the analysis of Amharic sentiment.

word frequency	null word vector	vocabulary size	Accuracy		
			word2vec	fasttext	word2vec and fasttext
1	129	26,953	86.8	90	89.33
2	830	25,237	87.65	89	89.98
3	13141	9,087	90	88	90.17
4	13661	8,650	91	86.65	91.65
5	16567	4,565	88	87.4	89.77

From the above table result we can see, adding the feature word frequency by decreasing minimum count of the word has negative effect on word embedding based similarity on Amharic informal text contains. Moreover, adding the feature word frequency by decreasing minimum count of the word has positive effect on character n-gram features based similarity on

Amharic informal text contains. This reduction is because the word embedding based feature model cannot capture complete attributes well with small amount of training vocabulary data size during training. Minum count is one of a parameters used in embedding layer for filter out the word which is mostly occurs or not occurs, which is an important mechanism for handling informality analysis. It increase the syntactical information of the word embedding but minimize the semantically information of the word embedding because semantically information needs repetitive pattern to get the meaning of similar context of different words in the sentences [14].

A word2vec is a network of artificial neurons of two layers that have the ability to learn how to represent each word with a real number vector with its semantic features [45]. As we have seen from the above table, Word2vec and the combination of word2vec and fasttext based automatic feature extractor model increased the accuracy by increasing the minimum count of the word frequency parameter. This indicate a reduction of training and testing vocabulary size by 3.11 reduces the accuracy on combination of word2vec and fasttext from 91.65 to 89.33.

Therefore, this increment or improvement on accuracy due to normalized informal text used to handle the morphological complexity of Amharic informal text in addition to handling it with character n-gram embedding. Even if normalization of informal word only gives comparable performance but it is too hard and less possible to do normalization for large size data. Nevertheless, improving the word out of vocabulary problems by decreasing minimum count of the word on the corpus or increased the word vocabulary size only is not guaranteed to enhance the performance on Amharic informal text based sentiment analysis. Because, the above table shows that the similarity of these words mostly based on semantics for the word embedding and morphology for the char-n gram embedding.

Therefore, effective resources and tools are needs to better understand and treat these various morphological forms when targeting sentiment analysis on informal text containing Amharic sentence. Because as we stated in the introduction, the form of informal contains Amharic sentence used varies widely, which leads to the rising of number of null word embedding. Our main understanding is that while word-level embedding with decreasing minimum count of the word on embedding space seems to give less importance to the Amharic sentiment analysis with informal text contains and char-level embedding with decreasing minimum count of the word on embedding space are more importance to the Amharic sentiment analysis with informal text contains. To sum up, the experiments conducted to answer RQ2 show that normalized informal text and embedding informal words by combination of word2vec and fasttext with decreased the minimum number of word frequency has a positive effect on sentiment analysis o Amharic informal text.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

In this study, we generated neural embedding and informal lexicon as a feature in Amharic languages based text to represent the morphology and semantics for each word in a given task. We aimed to highlight the effect of normalization as a preprocessing step on the construction of an enhanced sentiment analysis based on a semantics, morphology, and informal lexicon were all used in the study as features. Utilizing informal opinionated lexicon and neural embedding of informal text contains as a feature for automatic labeling has a good overall impact on the performance of Amharic sentiment analysis by incorporating them into a supervised learning framework for a range of affect-sensitive tasks. The neural embedding we used is char n-gram Embedding that processes the representation of the text at character level, which is more suitable for the classification of a informal text processing typical of Amharic informal text containing document, unlike the traditional formal text only containing documents. We present the result of a mixture of convolution and bidirectional recurrent neural networks model provides good results,since it benefits from the CNN's ability on the way to extract features. Whereas, BILSTM is characteristic in the direction of study long-term both past and before directional dependencies of the text. In conclusion, the experiments conducted show that normalizing in the Preprocessing and embedding the Amharic informal words using combination of word embedding and character n-gram embedding with decreased minimum word frequency parameter has a positive impact on Amharic sentiment analysis.

6.2 Recommendation

There are a number of limitation in our researches around public opinion involvement in sentiment analysis. That following challenges have also been from our findings would benefit for further research directions.

1. In depth examination of how combining text with corresponding pictures, audio, and video become beneficial to public opinion and how to influence the analyzing performance of public sentiment would be very helpful.

2. Huge amount of labeled data and new unlabeled datasets are required for more challenging tasks, especially for less resourced and morphological complex languages with memory efficient models are required.

3. Popular deep learning techniques such as deep reinforcement learning and generative adversarial networks can be evaluated to solve some challenging tasks.

4. Evaluate the effect of different usage of a similar homophone amharic words can be benefical to solve amharic morpholoical complexity tasks.

5. Other hybrid and ensembles approaches improve performance must be developed.

References

[1] Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. A combined cnn and lstm model for arabic sentiment analysis. In International cross-domain conference for machine learning and knowledge extraction, pages 179--191. Springer, 2018.

[2] Abebe Alemu. Sentiment mining for discovering hidden knowledge in opinionated amharic text: the case of movie review . University of Gondar, Gondar, M. sc Thesis department of computer science, 2012.

[3] Yeshiwas Getachew Msc Abebe Alemu. Deep learning approach for amharic sentiment analysis university of gondar. 2018.

[4] Abdullah I Alharbi and Mark Lee. Combining character and word embeddings for the detection of offensive language in arabic. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 91--96, 2020.

[5] Salihah AlOtaibi and Muhammad Badruddin Khan. Sentiment analysis challenges of informal arabic. IJACSA) International Journal of Advanced Computer Science and Applications, 8(2), 2017.

[6] Shereena M Arif and Mazlina Mustapha. The effect of noise elimination and stemming in sentiment analysis for malay documents. In Proceedings of the International Conference on Computing, Mathematics and Statistics (iCMS 2015), pages 93--102. Springer, 2017.

[7] Sang-Soo Baek, Jongcheol Pyo, and Jong Ahn Chun. Prediction of water level and water quality using a cnn-lstm combined deep learning approach. Water, 12(12):3399, 2020.

[8] Mengistu Belete. Sentiment analysis for amharic opinionated text. Mscthesis, Addis Ababa university, Addis Ababa, Ethiopia, 2013.

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135--146, 2017.

[10] Nick Cercone. Morphological analysis and lexicon design for natural-language processing. Computers and the Humanities, pages 235--258, 1977.

[11] Himadri Tanaya Chidananda, Debashis Das, and Santwana Sagnika. Sentiment analysis using n-gram technique. In Progress in Computing, Analytics and Networking, pages 359--367. Springer, 2018.

[12] Stefanie Dipper, Michael Götze, and Stavros Skopeteas. Towards user-adaptive annotation guidelines. In Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora, pages 23--30, 2004.

[13] Hang Thi-Thuy Do, Huy Duc Huynh, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen, and Anh Gia-Tuan Nguyen. Hate speech detection on vietnamese social media text using the bidirectional-lstm model. arXiv preprint arXiv:1911.03648, 2019.

[14] Ábel Elekes, Martin Schäler, and Klemens Böhm. On the various semantics of similarity in word embedding models. In 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), pages 1--10. IEEE, 2017.

[15] Yuliadi Erdani. Acquisition of human expert knowledge for rule-based knowledge-based systems using ternary grid. PhD thesis, Universitätsbibliothek Duisburg, 2005.

[16] Turegn Fikre. Effect of preprocessing on long short term memory based sentiment analysis for amharic language. http://etd.aau.edu.et/handle/123456789/22638, 2020.

[17] Selama Gebremeskel. Sentiment mining model for opinionated amharic texts. Unpublished Masters Thesis and Department of Computer Science and Addis Ababa University and Addis Ababa, 2010.

[18] Ralph Grishman. Computational linguistics: an introduction. Cambridge University Press, 1986.

[19] Karol Grzegorczyk. Vector representations of text data in deep learning. arXiv preprint arXiv:1901.01695, 2019.

[20] Yasasvi Hari, Rohit Singh, Kizito Nyuytiymbiy, and David Butera. Consenting to internet of things across different social settings. arXiv preprint arXiv:2102.09499, 2021.

[21] Maha Heikal, Marwan Torki, and Nagwa El-Makky. Sentiment analysis of arabic tweets using deep learning. Procedia Computer Science, 142: 114--122, 2018.

[22] Jin Huang, Jingjing Lu, and Charles X Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy. In Third IEEE International Conference on Data Mining, pages 553--556. IEEE, 2003.

[23] Nitin Indurkhya and Fred J Damerau. Handbook of natural language processing. Chapman and Hall/CRC, 2010.

[24] Muhammad Javed and Shahid Kamal. Normalization of unstructured and informal text in sentiment analysis. (IJACSA) International Journal of Advanced Computer Science and Applications, 9(10), 2018.

[25] Samed Jukic, Muzafer Saracevic, Abdulhamit Subasi, and Jasmin Kevric. Comparison of ensemble machine learning methods for automated classification of focal and non-focal epileptic eeg signals. Mathematics, 8(9):1481, 2020.

[26] Mohammad Karrabi, Leila Oskooie, Mohsen Bakhtiar, Mohammad Farahani, and Reza Monsefi. Sentiment analysis of informal persian texts using embedding informal words and attention-based lstm network. In 2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS), pages 143--147. IEEE, 2020.

[27] Worku Kelemework. Automatic amharic text news classification: Aneural networks approach. Ethiopian Journal of Science and Technology, 6(2):127--137, 2013.

[28] Jordan Kodner. Computational models of morphological learning. In Oxford Research Encyclopedia of Linguistics. 2022.

[29] Huong Thanh Le and Nhan Trong Tran. A sentiment analyzer for informal text in social media. Journal of Science and Technology, 131: 006--012, 2018.

[30] Alexander Ligthart, Cagatay Catal, and Bedir Tekinerdogan. Systematic reviews in sentiment analysis: a tertiary study. Artificial Intelligence Review, pages 1--57, 2021.

[31] Fei Long, Kai Zhou, and Weihua Ou. Sentiment analysis of text based on bidirectional lstm with multi-head attention. IEEE Access, 7:141960--141969, 2019.

[32] Daniel Loureiro and Alípio Jorge. Language modelling makes sense:

Propagating representations through wordnet for full-coverage word sense disambiguation. arXiv preprint arXiv:1906.10007, 2019.

[33] Amanuel Mersha and Stephen Wu. Morphology-rich alphasyllabary embeddings. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 2590--2595, 2020.

[34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[35] Saif Mohammad. A practical guide to sentiment annotation: Challenges and solutions. In Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, pages 174--179, 2016.

[36] Alejandro Mosquera, Yoan Gutiérrez, and Paloma Moreda. On evaluating the contribution of text normalisation techniques to sentiment analysis on informal web 2.0 texts. Procesamiento del Lenguaje Natural, 58:29--36, 2017.

[37] Zewdie Mossie and Jenq-Haur Wang. Social network hate speech detection for amharic language. Computer Science & Information Technology, pages 41--55, 2018.

[38] Zewdie Mossie and Jenq-Haur Wang. Vulnerable community identification using hate speech detection on social media. Information Processing & Management, 57(3):102087, 2020.

[39] Wondwossen Mulugeta and Michael Gasser. Learning morphological rules for amharic verbs using inductive logic programming. Language Technology for Normalisation of Less-Resourced Languages, 7, 2012.

[40] Samira Noferesti and Mehrnoush Shamsfard. Using linked data for polarity classification of patients' experiences. Journal of biomedical informatics, 57:6--19, 2015.

[41] Kennedy Odhiambo Ogada. N-grams for Text Classification Using Supervised Machine Learning. PhD thesis, Jomo Kenyatta University of Agriculture and Technology, 2016.

[42] Ugochukwu E Orji, Modesta E Ezema, and Jonathan C Agbo. Mining twitter data for business intelligence using naive bayes algorithm for sentiment analysis. International Journal of Progressive Sciences and Technologies, 27(2):412--419, 2021.

[43] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Asa: A framework for arabic sentiment analysis. Journal of Information Science, 46(4):544--559, 2020.

[44] Wondwossen Philemon and Wondwossen Mulugeta. A machine learning approach to multi-scale sentiment analysis of amharic online posts. HiLCoE Journal of Computer Science and Technology, 2(2):8, 2014.

[45] Anwar Ur Rehman, Ahmad Kamran Malik, Basit Raza, and Waqar Ali. A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools and Applications, 78(18):26597--26613, 2019.

[46] Fakhira Riaz. Language and literacy on and off-line: A case study with multilingual teenagers in pakistan. The Journal of Humanities and Social Sciences, 26(2):13, 2018.

[47] Aswathi Sajeevan and KS Lakshmi. An enhanced approach for movie review analysis using deep learning techniques. In 2019 International

Conference on Communication and Electronics Systems (ICCES), pages 1788--1794. IEEE, 2019.

[48] Mehmet Umut Salur and Ilhan Aydin. A novel hybrid deep learning model for sentiment classification. IEEE Access, 8:58080--58093, 2020.

[49] KS Sampada, Lalit Adithya, and NP Kavya. Performance analysis of multidimensional indexing in keyword search. In Proceedings of International Conference on Recent Advancement on Computer and Communication, pages 171--184. Springer, 2018.

[50] Ross D Shachter. Probabilistic inference and influence diagrams. Operations research, 36(4):589--604, 1988.

[51] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. arXiv preprint arXiv:1805.09843, 2018.

[52] Musa Shikur. A Hybrid Sentiment Classification for Amharic Book Reviews. PhD thesis, St. Mary's University, 2019.

[53] S Soumya and KV Pramod. Sentiment analysis of malayalam tweets using machine learning techniques. ICT Express, 6(4):300--305, 2020.

[54] Chrysoula K Themeli. Hate speech detection using different text representations in online user comments. no. October, 2018, 2018.

[55] Tulu Tilahun. Linguistic localization of opinion mining from amharic blogs. International Journal of Information Technology & Computer Sciences Perspectives, 3(1):890, 2014.

[56] Csaba Veres and Paul Kapustin. Enhancing subword embeddings with open n-grams. In International Conference on Applications of Natural Language to Information Systems, pages 3--15. Springer, 2020.

[57] Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. Sentiment analysis of comment texts based on bilstm. Ieee Access, 7: 51522--51532, 2019.

[58] Huang Yi, Sun Shiyu, Duan Xiusheng, and Chen Zhigang. A study on deep neural networks framework. In 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pages 1519--1522. IEEE, 2016.

[59] Lingling Yuan. An improved naive bayes text classification algorithm in chinese information processing. Science, pages 267--269, 2010.

Appendix

Appendices A. Programming Code

A1. Read the datasets/import the file

$importpandasaspd$

$importnumpyasnpimportmatplotlib.pyplotasplt$

$importseabornassnsdf = pd.DataFrame()$

$df = pd.read_csv('2020.csv', encoding =' utf - 8')$

$df.isnull().values.any()$

A2. Train and test split

$X = df['text']Y = df['lables']$

$fromsklearn.model_selection$

$importtrain_test_splitx_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state =$
42)

A3. Combination of word embedding and character embedding on embedding matrix.

$fromtqdmimporttqdmfromnltk.corpus$

$importstopwordsfromnltk.tokenize$

$importRegexpTokenizer$

$importos, re, csv, math, codecsprint('loadwordembedding...')$

$embedding_index = tokenizer = RegexpTokenizer(r'w+')$

$coefs = np.asarray(values[1 :], dtype =' float32')embedding_index[word] = coefs$

$f.close()$

$g = codecs.open('C : /Users/TenagneComputer/Desktop/AZ/normalized_fasttext_1.vec', encoding ='$
$utf - 8')forlineintqdm(g) :$

$word = values[0]coefs = np.asarray(values[1 :], dtype =' float32')embedding_index[word] =$
$coefsg.close()print('foundswordvectors'len(embedding_index))$

$MAX_NB_WORDS = 100000word_index = Tokenizer_obj.word_indexwords_not_found = []nb_words =$

$min(MAX_NB_WORDS, len(word_index)+1)$

$embedding_matrix = np.zeros((nb_words, embed_dim))forword, iinword_index.items():$

$if(i >= nb_words):$

$continue$

$embedding_vector = embedding_index.get(word)$

$if(embedding_vectorisnotNone)and$

$len(embedding_vector) > 0:$

$embedding_matrix[i] = embedding_vector$

$else:$

$words_not_found.append(word)print('Numberofnullwordembeddingsfoundd'np.sum(np.sum(embeddi$

$1) == 0))$

A4. Deep neural network modeling with CNN-LSTM

$fromkeras.layersimportEmbedding, Input$

$fromkerasimportModelfromkeras.models$

$importSequentialfromkeras.models$

$importSequentialfromkeras.layers$

$importDense, Dropoutfromkeras.layers$

$importEmbeddingfromkeras.layers$

$importLSTMfromkeras.layers$

$importEmbeddingfromkeras.layers.convolutional$

$importConv1Dfromkeras.layers$

$importLSTM, Bidirectionalfromkeras.layers$

$importEmbeddingfromkeras.layers.convolutional$

$importConv1Dmax_len = 34model = Sequential()$

$input_length = max_length, weights = [embedding_matrix], trainable = True))model.add(Embedding(nb_wo$

$max_length, weights = [embedding_matrix], trainable = True))model.add(Dropout(0.2))$

A5. Add a Convolution1D and LSTM

$model.add(Conv1D(filters, kernel_size, padding = "valid", activation = "relu", strides =$

$1))model.add(Bidirectional(LSTM(units = lstm_o p_s ize)))$

A6. Add fully connected layer

$model.add(Dense(hidden_d ims))$

A7. Apply 30 layer dropout

$model.add(Dropout(0.3))$

$model.add(Activation(''relu''))$

A8. Output layer then sigmoid

$model.add(Dense(3))$

$model.add(Activation(''softmax''))$

$model.compile(loss =' sparse_c ategorical_c rossentropy', optimizer =' adam', metrics = ['accuracy'])$