2020-08

Multi Label Amharic Text Classification Using Convolutional Neural Network Approaches

Habtamu, Wubalem Dinku

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

FACULTY OF COMPUTING

Multi Label Amharic Text Classification Using Convolutional Neural Network Approaches

By

Habtamu Wubalem Dinku

Bahir Dar, Ethiopia

August, 2020

Multi Label Amharic Text Classification Using Convolutional Neural Network Approaches

Habtamu Wubalem Dinku

A Thesis Submitted to the School of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree

of

Masters of Science in the Information Technology program in the Faculty of Computing

Advisor: Gebeyehu Belay (Dr. of Eng.)

Bahir Dar, Ethiopia

August 14, 2020

Declaration

This is to certify that the thesis entitled "Multi label Amharic Text Classification using Convolutional Neural Network Approaches", submitted in partial fulfillment of the requirements for the degree of Master of Science in **Information Technology** under **Faculty of Computing** ,Bahir Dar Institute of Technology , is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Habtamu Wubalem Dinku		24/06/2020
Name of the candidate	signature	Date

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student **Habtamu Wubalem Dinku** Signature _____ Date **24/06/2020** As members of the board of examiners, we examined this thesis entitled "**Multi label Amharic Text Classification Using Convolutional Neural Network Approaches** " by *Habtamu Wubalem Dinku* We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "**Information Technology**".

Board of Examiners

Name of Advisor

Gebeyehu B (Dr.)

Signature

Date

14/08/2020

Name of External examiner

Yaregal Assabie (PhD)

Signature

Date

14/08/2020

Name of Internal Examiner

Abinew Ali(Asst. prof.)

Signature

Date

14/08/2020

Name of Chairperson

Esubalew Alemneh (PhD)

Signature

Date

14 Aug 2020

Name of Chair Holder

Derejaw L.

Signature

Date

Aug 17, 2020

Name of Faculty Dean

Belete Biazen

Signature

Date

11/12/2020 E.C

ACKNOWLEDGEMENT

ABSTRACT

Text classification is a technique which classifies textual information into a predefined set of categories. With the continuously increasing amount of online information, there is a pressing need to structure information. Automatic text classification is an inevitable solution in this regard. However, the present approaches are multi label text classification study aims to achieve news text classification and goal is to and out if it's possible to use a machine learning approach construct a classification system that can be used to extract features automatically.

The main objective of natural language processing is to make computers perform tasks that require the involvement of human.in order to solve labor force, cost and time devoted to do such tasks. These goals are achieved by implementing activities such as text classification, sentiment analysis, entity recognition and information retrieval. However, classification accuracy decreases and computational complexity increase as the number of categories increases specially in single label text classification. The aim of this study is to develop design and model scheme for multi label Amharic text classification using convolutional neural networks.

To achieve the objective, able to design effective model and to know the state of art, different literature was reviewed. Then designed multi label Amharic text classification using CNN consists preparing word embedding and by accepting the input vectors and token from embedding construct cnn model by setting parameters. The word embeddings are used individually and in various combinations through different channels of CNN a single dense layer with six outputs with a sigmoid activation functions and binary cross entropy loss functions to predict class labels. In order to handle such issues, we were used software Python used to pre-process and design the artifact model.

Finally, the multi label Amharic text classification model achieves an accuracy of 97.69%. The proposed model is pretrained embedding for CNN compared to other word2vec files that are not pre-trained embedding, testing for specific data and labels in typically six class datasets are used.

Keywords: -multi label text classification, CNN, word embedding

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AMMA	Amhara Mass Media Agency
ANN	Artificial Neural Network
BC	Binary Classification
BR	Binary Relevance
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
ELMO	Embedding for Language Modeling
FC	Flat Classification
HAN	Hierarchical Attention Network
LP	Label Power-set
MLTC	Multi-label Text Classification
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SG	Skip Gram
SLTC	Single-label Classification
SVM	Support Vector Machine
TC	Text Classification
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 Background

Now a days, technological advancement is rising rapidly and production of vast amounts of information in daily life as well. In particular, text databases are growing rapidly due to the increasing amount of information available in electronic forms such as e-publications, e-mail, blogs, social media and world wide web, So we need for a well-defined methodology to analyze and classify these voluminous data has drawn many communities' attention to this kind of data which is known as unstructured data. This phenomenon has made the importance of text classification begins to spring up.

Different researches were conducted to build an Amharic document classifier. However, the computational complexity increase as the number of categories increases. In parallel the classification accuracy, decrease with a higher rate as the number of categories. Therefore, the main aim of this study is to explore multi label Amharic text classification using deep learning approaches after create distributional word representation.

Text document classification (TC) is the process of assigning the text documents to one or more proper category based on their content by building model through a training data. Text classification (C. C. Aggarwal, 2012). is a process of assigning classes or marks to a public text document. This process is considered a supervised classification technique, since a set of predefined labeled documents is provided as training set. There have been various issues in proper text classification of texts. The size of datasets presents a big challenge, high dimensionality, large number of attributes which presents the problem of poor classifier performance. There are two basic approaches of document classification. According to (Alemu Kumilachew, 2017) These are concept (semantic) based and keyword based. For this study we follow concept using representing semantic document text a new concept of distributional representation of words as vectors.

Automatic text classification can be done in three different manners such as Supervised semi-supervised and unsupervised. In supervised manner, experts in the area can label or determine class of the data. Since we have labeled dataset, we use supervised learning scheme. Again, In the supervised learning tasks in machine learning algorithm can be categorized in to different groups according to the partition of instance and their target categories Such as single label classification, Hierarchical classification, Multi label text classification. (Kanj, 2017)

Single-label classification: Single-label (or traditional) classification is the task of automatically assigning a single class label to each input instance, where a classifier learns to associate to every un seen element its most probable one class or category. In general, single-label classification problems can be divided into two main groups: binary and multi-class problems. The binary classification problem is the simplest case, where the set of classes is restricted to two. In this context, we distinguish between the positive and the negative class, and we denote by positive instances, those having the positive class as a true label and vice versa.

Hierarchical classification: Hierarchical text classification is proposed by (Koller, 1997), as divide-and-conquer approach that utilizes the hierarchical topic structure to decompose the classification task into a set of simpler problems, one at each node in the classification hierarchy. These algorithms can be categorized into three groups: flat classification (A single huge classifier is trained which categorizes each new document as belonging to one of the possible predefined classes (Alemu Kumilachew, 2017). local classification (or Top-Down) and global classification (or Big Bang) approaches.

Multi label classification: -multi label classification is the task of assigning to an input instance multiple classes simultaneously from a set of disjoint classes; the classes are then no longer mutually exclusive. In this context, we often use the term "label" instead of "class". Each instance is associated with a set of relevant labels. The remaining labels are considered as irrelevant. (Younes. Z, 2011) For example, a patient with a high blood pressure is more likely to develop heart disease than another person, but less likely to

develop a muscular dystrophy. Multi-label learning has important applications in many real-world problems like text categorization, scene classification, video annotation and bioinformatics. Nowadays, multi-label classification approaches are increasingly required by real applications, such as gene classification and article annotation. A common method is to convert a multi-label problem into single label tasks without considering the relationship between labels (J. Fan, 2008).

The main objective behind this work is to use machine learning approaches to create models that allow an automatically created context dependent and rich representation of the classification of multi-label Amharic news documents based on category related details. Convolutional neural networks (CNNs) have dramatically improved the approaches to many active research problems. One of the key differentiators between CNNs and traditional machine learning approaches is the ability for CNNs to learn complex feature representation. We present to apply a CNNs to learn complex feature representation (Mark HUGHES, 2013). We proposed to apply a CNN-based approach to categorization of text fragments, at a sentence level, based on the emergent semantics extracted from a corpus of Amharic news text.

1.2 Statement of the problem

Now a day of rising ICT infrastructure is generating an exponential increase in digital content in Ethiopia. The steady growth in locally available data would increase the demand for software to simplify the extraction of relevant data. A lot of work had been done in English and Latin languages compare with low-resource language. Amharic is an Afro-Asiatic language family belonging to the Ethiopian Semitic group with its own unique alphabet.

For text classification, the problem is there is no pre-existing and reliable vocabulary to utilize in the sciences that guarantees important ideas are always expressed in the same terms. Moreover, underlying ideas in research are rarely easily expressed in simple "keyword" terms; they are concepts that require multi-word explanations that contain

multiple underlying concepts. Major challenge is to mine the scientific documents for the words that indicate the underlying concepts and then assign labels that make these concepts express explicitly. In general, there is not a standard keyword collection to be used for all text classification tasks. For example, some keywords appear in a computer science paper do not play an important role in the biology article. For a complex text structure, simply searching for a set list of keywords isn't enough.

Another problem facing various researchers in all the studies is the decrease in accuracy as the number of classes increases and each input instance is given a single class . (Surafel, 2003) (Worku, 2006 E.c) (Yohannes, 2007) (Alemu Kumilachew, 2017). This research proposed to uses one of the neural networks learning algorithm called Convolutional neural network (CNN) to study Automatic Multi label Amharic text news classification.

Research Question

RQ1: [*preprocessing for better representation and classification*]:

How text classification datasets are prepared for better word representation and evaluation of effects in Amharic text classification?

RQ2: *[number of class and news items increase what will be performance*]:

How to make effective and efficient when increasing number of labels and news items on multi label Amharic text classification performance using Convolutional neural network learning method?

RQ3*: [achieve multi label text classification Amharic text using CNN]:*

How to design a model for a multi label Amharic text classification using convolutional neural network learning methods?

1.3 Objectives of the study

1.3.1 General objectives

The major objective of this study is to develop model Multi label Amharic text classification using convolutional neural network.

1.3.2 Specific objectives

To realize the above-mentioned general objective, the study aims to carry out the following tasks.

- To develop and adopt tools of processing Amharic documents for classification purpose.
- To select suitable preprocessing techniques, feature extraction schemes, and classification algorithms for Multi label Amharic text classification.
- To construct a model for Multi label Amharic text classification.
- To examine the ways of reducing the problem of Multi label Amharic text classification by using the selected algorithms.

1.4 Scope and Limitation of the study

In order to properly classify the published documents or even a short text, it is essential to have automated categorization systems that can clearly label published articles. But there is no any such label Amharic document for the purpose of Multi label text classification. Currently, such labeling or categorization requires the time and effort of professional experts who have strong background and experience. Manually annotating articles is a time-consuming work. The scope of the present study is limited to design multi label text classification systems for Amharic news document. The data set composition of this study is text file formats; The system is trained and tested using limited amount of Amharic news text documents collected from Amhara Mass Media Agency.

This is due to lack of standard Amharic corpus prepared for text categorization purposes specially for Multi label Text Classification and we use datasets that consists of only a small

number (3 labels) maximum marks that appears in one sentence due to the nature of news text by decision of the knowledge expert. The CNN classifiers are covers Multi label text classification which covers also flat, multiclass classification it does not cover hierarchical classification. In addition, the classifier used in this study as input word embedding text representation technique to transform text to real valued vectors.

1.5 Significance of the study

The outcome of the study will have had a better success for text classification, especially for studying Multi label Amharic news text classification. This is also undeniably relevant because of the following points.

- The current study can be included in the media organization news text classification for Multi label Amharic text classification. Because in the near future everything will be digital so that it is easy to manage the amount of text data in which categorization is possible.
- This study has a great contribution when there will be an increase in text data from vast amount of information was produce and need to data access through the internet, social media and retrieval becomes more and more important.
- It can be used as a base for other researchers who are interested to work on Multi label Amharic text classification using neural architectures.
- Experimental evidence of the effectiveness of selected classifier algorithms for automatic classification of Amharic documents.

1.6 Organization of the Thesis

This thesis has five chapters. The remainder of the thesis presents what is done on this thesis work. Chapter two presents literature review and related works, while chapter three deals with the methodology we used in this research work, chapter four deals with result and discussion, finally, Chapter five presents the conclusion and recommendation.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter presents the processes in document classification, such as pre-processing, document representation, feature weighting and classification. With the rapid increase in recording processing, retrieving correct document is hard for the current retrieval system. Text categorization plays key roles in the handling and organizing of text data in order to strength this recovery framework task.

2.2 Text Classification

Text classification dates back to the early 60's, but got popular in the early 90's (SEBASTIANI, 2002) with the rapid growth of online data, the classification of text becomes one of NLP's tasks. It is important to manage and organize data that flows through social media, or through various media such as books, videos, etc. for efficient use of data, classification of news stories by author or subjects ,classification of help tickets by urgency, tagging of items by category, facilitation of storage search, and other related tasks are addressed using text classification.

We can find unstructured and unlabeled raw data in social networking sites and media, in the form of text, chat conversations, email messages, web pages. Currently, companies use text classification for structuring, automated marking and retrieval, combining documents and texts in a cost-effective way for automation processes and enhancement of decision making. These days, text classification is used by businesses for structuring, automatic labeling and extraction, balancing documents and text in a cost-effective way for automation processes and enhancement of decision-making. For example, given a text, a classifier can take the contents of the text t, analyze its content and then automatically assigns relevant categories.

Classification of text is one of the important tasks for the NLP in areas such as interpretation of emotions, recognition of intent and intelligent responses. The aim of classifying text is to classify documents (such as reviews, opinions, news, comments, tweets, replies, emails, etc.) into specific categories. It includes assigning predefined tags to free-text documents (Zhang, 2015). The tags or categories can vary from two SLTC to n MLTC.
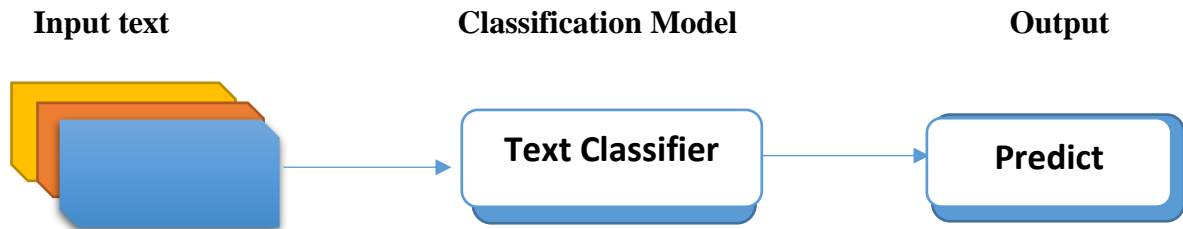
Figure 2. 1 Text classification Overflow Diagram

General definition of classification

The general text classification problem can formally be defined as the process of predicting a new category assignment function $f : D \times C \rightarrow \{0,1\}$, where D is the set of all possible data and C is the set of predefined categories. The value of F (d, c) is 1 if the text or document or data d belongs to the category c and 0 otherwise (Feldman & Sanger, 2007). The predicting function f: $D \times C \rightarrow \{0, 1\}$ is a classifier, that produces results as "close" as possible to the actual category assignment function F.

The advent of deep learning in the research area had a considerable impact on the development of NLP helps computers to extract meaning from natural languages (Farzindar A, 2015). It is a multidisciplinary area where able to designed and built to evaluate, interpret, understand and generate useful information from the natural languages used by people. Although the field has been around for a long time as trend topics for research and study, significant advances and remarkable achievements have been observed in NLP in recent years. Therefore, this study follows the machine learning approach.

2.3 Types of document classification

There are three types of document categorizations. Here below we show such categorization types.

I. Flat and Hierarchical classification

Generally, document classification can be divided into two categories, flat and hierarchical classification. The flat document classification does not consider sub-titles exist in the main document. It is very difficult to search through categories if the number of documents in that category is massive, whereas, hierarchical classification is created to solve some of the problem of flat classification. (Alemu Kumilachew, 2017) Hierarchical classification constructs, the relation between documents that are kept by dividing the main document into a different flat classification of sub-categories. Hierarchical classification solves some of problems of flat classification. However, it is computationally expensive and data set preparation takes too much time (Alemu Kumilachew, 2017).

II. Single-label and Multi-label classification

While single-label document classification assigns only one category to a document in the collection. Whereas, multi-label classification assigns zero or more than one category to a document in the corpus. We use multi label classification approach.

III. Hard and Soft classification

Hard type of classification is a binary decision like true or false. A document belongs to a particular group if it clear follows one selection criteria whereas, by taking score ranges soft classification assign category to a document text class. (Yufeng Liu, 2011)

2.4 Multilabel versus Single-label classification

The single-label text categorization problem assigns only one predefined category to each "unseen" natural language text document and often defined as non-overlapping (Antonie, 2002). In this label for a given integer k each element of C must be assigned to exactly k (or < k, or >k) elements of D. For instance, this happens when the category needs to be evenly populated (Berger H. , 2004). So, when there are two or more categories in category it assigns an object to exactly one category. Because of the existence of text overlapping each other in category spaces it is difficult to categorize each document under a single label.

However, it is impossible to categorize each document under a single label because of the nature of the text overlapping each other in the category spaces. For example, the economics field often overlaps (relates) with the political science field. This fact forces the different constraints on single-label text categorization task. Whereas the multi-label case is general case in which any number of categories from 0 to M (M is at least one) may be assigned to the same document (Popa, 2007). In general, multilabel classification assigns a text, data, or sample to one or more than one, or no label at all.

The categorization of multilabel text assigns more than one predefined category to an "unseen" document and is named as overlapping text categorization tasks because it is the task of assigning an object to one or more categories simultaneously. According to (Addis, 2010), the single-label is more general than the multi-label categorization. Because the algorithm for single-label can be used for multi-label by transforming a problem of multi-label with categories $\{c1, c2…, C_m\}$ into m independent problems of single-label classification with categories $\{ci\}$, for i = 1, 2…... m. This can be done if the categories are stochastically independent of each other. However, the converse is not true in general. If there is algorithm for performing the multi-label, it is not always true that it can use for single-label categorization. Nevertheless, if the text, data, or sample or document belongs to exactly one class or category, the classification is known as multiclass. Here each sample belongs to exactly one group, as the groups or marks are exclusive to each other. It assigns

one and only one label to each sample. For evaluation technique the first type, i.e. multilabel classification, is chosen in this work.

2.5 Techniques of Multi label text classification

In the classification of multi-label text, different approaches are used to classify the text given to a specific category. For SC problems most conventional learning algorithms are developed. Many methods in the literature therefore turn the multi-label problem into multiple single-label issues, so that the current single-label algorithms can be used. There are quite a few different approaches and here we'll give you an overview of them. A clear distinction can be made between algorithm-dependent approaches and algorithm-independent approaches to problem transformation.

Algorithm independent problems transform the problem to multiple single-label problems, which can be further divided into instance-based and label-based methods. When the transformation is based on instances, i.e. simply eliminating all instances with multiple labels, it is termed instance-based. Label-based methods transform the multi-label problem in to one or more single-label issues problems by only looking at the class labels (Freitas, 2009). A hierarchical structure representing the approaches, here on the figure 2.2
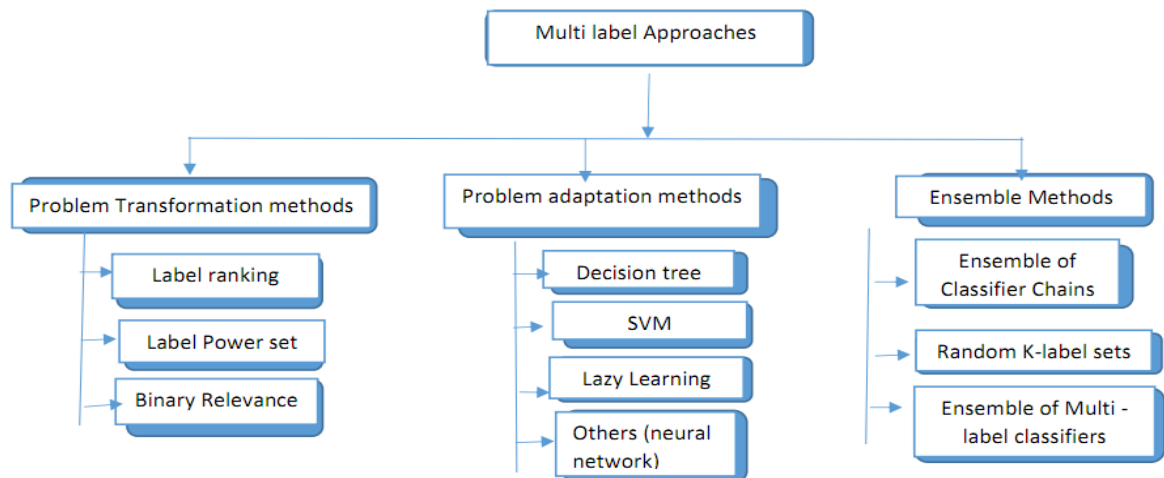
***Figure 2. 2 Multi label Learning approaches (adopted from** (Kanj, 2017)*)*

Most approaches have to choose between at one hand the computational complexity and on the other hand taking into account the correlation between class labels. The simplest method is probably instance elimination, which simply ignores all multi-label instances. Another simple method called simplification, randomly selects one label from multiple labels and uses that to train on. The two previously mentioned approaches however do not give good results. We proposed to use two algorithm independent approaches (problem adaption methods) solve multi label problem by the selected algorithms and adaption the problem. So, we are selected neural network architectures using CNN method.

2.6 Amharic Language writing Systems

Amharic (አማርኛ- Amarəña) is a sematic language that is spoken mainly in Ethiopia. It is the second largest Semitic language next to Arabic. Amharic is the working language of the federal government of Ethiopia. Amharic is spoken by about 30 million people as a first or second language, making it the second most spoken Semitic language in the world (after Arabic), probably the second largest language in Ethiopia (after Oromo, a Cushitic language), and possibly one of the five largest languages on the African continent.

(Gambäck B. &., 2010) it has its own alphabet ፊደል (Fidel). Fidel is a syllabary writing system where the consonants and vowels co-exist within each graphic symbol. Amharic language has basic 33 characters, each character has seven forms depending on the vowel is used to pronounce it. There are problems associated with Amharic writing that challenges NLP in Amharic texts. Some of the major challenges of Amharic document processing are described below.

2.6.1 Problem of Amharic writing system

There are a number of problems associated with Amharic writing system which are challenging natural language processing of Amharic documents; which are dealt below. Redundancy of some characters: sometimes more than one character is used for similar sound in Amharic (Zelalem, 2001). Though the various forms have their own meaning in

Ge'ez, there is no clear-cut rule that shows its purpose and use in Amharic according to (Bender, 1976)

Table 2. 1. Illustrates the different forms of Amharic characters with similar sound.

Character	Other form/s of the character
ሀ (hä)	ሐ and ኀ
ሠ (sä)	ሰ
አ (ä)	ዐ
ጸ (tsä)	ፀ

The problem of the same sound with various characters is not only observed with core characters, but also exhibited in the same order of characters. For example, ሀ and ኃ, ነ and ኃ; አ and ኣ; etc ((Tewodros, 2003)). The use of various forms of characters for the same sound poses a problem in the process of feature preparation for the classifier learning since the same word is represented in different forms. For example, the word 'ጸሀይ' ('sun') can be represented in Amharic as ጸሀይ, ጸሐይ, ጸኀይ, ፀሀይ, ፀሐይ, ፀኀይ, etc. Amharic characters with different forms of the same sound Character other form/s of the character.

2.6.2 Spelling variation of the same word

One can imagine how the meaning of the original word is diverted to different contexts. Spelling variation of the same word: the same word is written in various forms (Tewodros, 2003). For example, the word 'ሰምቷአል' ('he hears') can be written in Amharic as ሰምቷአል, ሰምቷል, ሰምትዋል, etc. Spelling variation may happen also in the case of translating foreign word to Amharic. For instance, the word 'ቴሌቪዥን' ('television') can be written as ቴሌብዢን, ቴሌብዥን, ቴሌቪዥን, etc.

2.7 Steps in automatic text Classification

Automatic text classification involves pre-processing, word representation, dimension reduction, feature weighting and classification phases. (SEBASTIANI, 2002) We describe the use of each phase in this study for the purpose of Amharic text classification.

2.7.1 Pre-processing

The first stage in document classification is document pre-processing. Document pre-processing helps us to transform a raw document to the form that can be used by the classifier. It is highly required in document classification, since not every term or features in the document is useful for classification.

2.7.2 Tokenization

Tokenization is the process of identification of all the individual words of a document. In Amharic, punctuation marks and spaces are used to indicate the beginning and ending of tokens. Tokens can be defined as the last chopping pieces" of document unit. Tokenization split documents into its last pieces (tokens); at the same time, it eliminates characters like punctuation marks. Terms in Amharic documents are identified by Amharic word separators such as single space, netela serez (፣), hullet netb (፡), dirb serez (፤), arat netb (።), carriage return, line feed, tab, etc.

2.7.3 Normalization

In the Amharic writing system, there are different characters with the same sound, which are called homophones. For instance, ሰ and ሠ; ሀ, ሐ, and ኀ; ጸ and ፀ so on are Amharic alphabet consonants having the same meaning and sound. To avoid the unnecessary representation of a given word in different forms normalization is required. Normalization helps to identify keywords in a document.

2.7.4 Stop-words removal

Words in the document do not have equal weight in the classification process. Some are used to fill the grammatical structure of a sentence or does not refer any object or concept. The list of non-content baring terms must be collected to reduce the space and time complexity of text processing. The common words in English text like, a, an, the, who, be, and other common words that bring less weighted in the document preprocessing are known as stop-words. Like any other languages, Amharic has its own stop-words like ብለዋል, ተከናውኗል, ነበረ, አስታውሰዋል, ነበር, ነው, ናቸው,ነገርግን, ሁኔታ, ሆኖም, and so on. (Worku, 2006 E.c) Since this study focused on news document classification, news related stop-word list like አስታወቀ, ተናግረዋል, ብለዋል, እስካሁን , አሳሰበ , አስረድተዋል etc. are taken in part. In this study, we used two mechanisms to remove stop-words: remove words that found in stop-word list prepared for this study and remove those terms, which have relevance score less than the defined thresholding with each scoring metric. Teshome Kassie describes about stop words that since stop and function words have no discriminating power for information retrieval, they must be removed and they list the stop words. (Kassie, 2009)

2.8 Deep Learning

Deep Learning is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered features. The two main deep learning architectures used in text classification are CNN, RNN and HAN (Mark HUGHES, 2013) are found very effective in the work of text classification. Deep learning models have gained popularity in computer vision (Krizhevsky, 2012) and speech (Graves, 2013) in recent years. Within NLP, much of the work with deep learning methods has involved learning word vector representations through neural language models (Bengio, 2003) (Mikolov T. Y., 2013) and performing composition over the learned word vectors for classification (Collobert J. W., 2011).

CNN is type of a deep learning, feed forward ANNs that uses a variation of multilayer perceptron designed to require minimal preprocessing. ANNs are nowadays vastly used not only in text processing but also in image processing applications. This network exploits the spatial structure of data to learn about it so that useful output can be obtained. (Oludare Isaac Abiodun, 2017) CNN models, originally invented for computer vision, utilize layers in word vectors to extract local features. In NLP, the features as an input usually take the form of word vectors.

The input to a CNN, given a tokenized text $T = \{t_1, t_2 \ldots t_N\}$, is a text matrix A where the ith row is the word vector representations of the ith token in $\mathbf{T}$. The matrix A can be denoted as $A \in R^{N \times d}$ where d is the dimensionality of the word vectors. CNNs use convolutional layers which are like a sliding window over a matrix. CNNs are many layers of convolutions with non-linear activation functions. In CNNs the output is computed over the input layer from local connections and then each layer applies different kernels, usually thousands of filters, to then combine their results.

According to (Kim, 2014) CNNs perform remarkably well for classification by using different tuning of hyper parameters. CNNs utilized the distributed representation of words after converting the tokens comprising each sentence into a vector which forms a matrix to be an input. However, these models require setting hyper parameters and regularization of parameters (Zhang Y, 2015). The other family of deep learning methods is the Recurrent Neural Network. RNN is one of deep neural network approaches where connections form a recurrent node (or a directed graph) along a sequence. They are networks with loops that aids in persistence of information. RNNs improved the traditional neural networks which considers all inputs as independent to each other by gaining memory and capturing information in arbitrary long sequences and predicting the previous and next sequences in the networks. They are deep in temporal dimension and used in time sequence modeling.

The role of RNNs in text classification is recurrently and sequentially process words in a sentence and map a dense and low-dimensional representation of words into a low-dimensional vector. one of the features of RNNs is their capability to improve time

complexity and analyze texts word by word there by preserving the context of texts. This ability arises from their way of capturing the statistics of a long text. In this perspective RNNs has fall short of balancing the role of both earlier words and recent words. This issue can be overcome by introducing long short-term memory (LSTM) model.

Word Vector Representation

Vector Representation of Words", one of convenient method of representing words as vectors, also known as word embeddings. Most of the models are work with high-dimensional vectors of datasets ranging from pixel values in raw images to power densities in audio tracks. Now, all the useful information that your text is in this raw data whereas, in the case of working with natural language processing, words are treated as symbols, For example, the word *"ከተማ"* can be represented as a vector of 4 dimensions *[0.006821954 0.012614403 -0.0027638178 -0.0019159443]* and *"ወረዳ"* has a vector of *[0.0068338476 -0.0021512192 -0.0076785153 -0.01227795*] from the give document. This allotment is random and provides no information if the data is interrelated. Therefore, representing words as individual identities may lead to sparsity in the dataset, which increases the need for more data for your model.

In linguistics and language theory, particularly in the works of (Harris, Distributional Structure. WORD, 1954) and in the 1950s, the concept of word embedding and representations has its origins use hand-crafted features, for example, object representations are used to measure semantic similarity. The semantic differentials technique was used by scholars in the early 1950s to measure the meaning of concepts.

Methods for using contextual characteristics were later developed in various thematic areas of study in the 1990s. Latent Semantic Analysis (LSA) was the most known. LSA is a method used to examine the relationships between documents and words within them, in NLP in general and in distributional semantics in particular, by having a collection of concepts relevant to documents and words. This technique is taken for granted. This uses a technique called Singular Value Decomposition (SVD) to reduce the number of rows in the

matrix while remaining intact with the linguistic features. The linguistic features such as the sense of contextual use of words are extracted and interpreted by statistical computations applied to a text corpus. It helps predict continuing term representations.

At about the same time, in the sense of population genetics, the Latent Dirichlet Allocation (LDA) was introduced by (Pritchard, 2000) This scheme was rediscovered in 2003 by LDA as a generative probabilistic model of a collection of documents and terms and/or phrases in the sense of machine learning (Blei, 2003) .LDA can be used to collect any discrete data, according to the authors. Self-organizing maps and simple recurrent networks are other well-known models based on neural networks contextual representations. The former, developed by uses (Kohonen, 1982), unsupervised, competitive learning to generate low-dimensional high dimensional data representation while at the same time maintain intact similarity relationships between data objects. Although the concept behind word embedding was found in the early works of (Harris, Distributional Structure WORD, 1954) the introduction of automatically generated contextual features an deep learning methods for NLP gives word embedding the opportunity to become the most common research areas in the early 2010s (Mandelbaum, 2016) since then, new technologies and different embedding models have been developed.

Later (Collobert R. &., 2008) shows the power of pre-trained word vectors as a tool for downstream tasks ranging from structural linguistic features like POS tagging to language-based meanings and logic, such as word-sense disambiguation. A single CNN architecture was also introduced by the authors that defies older systems. However, it was (Mikolov T, 2013) who made the eventual popularization of word embeddings after they released word2vec. Following the release of the word2vec toolkit, word embeddings became the latest in natural language processing. This sparked a huge amount of interest in the topic. In (Jeffrey Pennington, 2014) released Glove, another model for unsupervised learning of word representations, which brought word embeddings to the mainstream NLP tasks. This model develops a co-occurrence matrix using the global statistics of word-word co-occurrence.

Glove uses the strengths of word2vec skip gram model for word analogy tasks, and matrix factoring methods for global statistical knowledge (Bengio, 2003) was first person to agree with term word embeddings. As per terminology, word embedding has different names like distributional semantic model, distributed representation, semantic vector space and so on. We have used popular term word embedding. Word embedding is the task of converting words, strings, or characters into machine-readable formats specifically vectors. It is a means of representing a word as a low dimensional vector which preserves the contextual property of words (Mikolov S. K., 2013)

A word that embeds words into a vector space as the name indicates. Each word is associated with a vector in a way that preserves the relationship between words. Relationship between words is expressed by the vector relationships. "Similar words are associated with similar vectors in this vector representation." (Collobert R. &., 2008) (Mikolov T. Y., 2013) .The vectors associated with words are called word embeddings, also known as word vectors (Basirat, 2018) Words are converted into real numbers. Therefore, word embedding can be described as vector representation of a word.

(Berger M. J., 2019) Multi-label text labeling has been extended to a variety of tasks, including document indexing, recommendation of tags and classification of feelings. Researchers explore how word embedding and word order can be used to enhance the learning of multi-labels. Furthermore, they are investigating how both a convolutionary neural network (CNN) in Amharic text classification to solve multi label problems.

Types of word embedding

Word embeddings are classified into two broad categories: -
   a) Count (frequency) based embedding and; b) Prediction based embedding. (Singh, 2017) These types are discussed as follows.

   a) frequency-based embedding

In frequency-based embedding, various vectorization methods are employed, amongst the methods the widely known are count vector-IDF(TF-IDF) vector and co-occurrence vector. In count vector method, the number of times each word appears in each document is assessed. For example, suppose there are D documents, T number of different words from all documents (called vocabulary). Then the size of the count vector matrix will be D×T. (Singh, 2017)

Frequency-based approach faces two problems in the first place. First, the size of the vocabulary and the dimension (after multiplication) would be very huge for bigger corpus. For big data, with millions of documents, hundreds of millions of unique words can be extracted. Therefore, the matrix would be very sparse and inefficient for computation. Second, there is no straightforward way of counting every word, whether using the frequency method or simply using the presence of terms. In this second point, if word frequency is considered, the least important words such as stop words, punctuation marks, etc. are the most commonly used in real life corpus. That's another issue. The solution is TF-IDF vectorization for this case.

TF-IDF stands for Term-frequency - Inverse Document Frequency. Term frequency (TF) shows the number of times a term or a word occurs or just frequency of occurrence (Salton, 1988) in a document. Term frequency often used in Text Mining, NLP and Information Retrieval tells you how frequently a term occurs in a document. In the context natural language, terms correspond to words or phrases. Since every document is different in length, it is possible that a term would appear more often in longer documents than shorter ones. Thus, term frequency is often divided by the total number of terms in the document as a way of normalization. IDF only reduces the occurrence of the most common terms in a document and increases the weight of rare terms. IDF takes the richness of a word into account by evaluating how abundant a word is or whether it is a common word or not. It is the quotient logarithm of the ratio of the total number of documents to the number of documents, as shown by the word t

$$idf(t,d) = log \frac{N}{|\{d \in D : t \in d\}|} \tag{2.2}$$

Where N is number of documents in the corpus N = |D|

| {d ∈ D : t ∈ d} | → number of documents where term t is part of (t*f* (t, d) ≠ 0).

If term t is not part of the corpus, the denominator will be adjusted to 1 + | {d ∈ D : t ∈ d} | to avoid division by zero. Then

$$tf - idf(t, d, D) = tf(t, d). idf(t, D) \qquad \text{(2.3)}$$

The other method worthy to discuss is the co-occurrence matrix. This captures the extent words occur together so that relationships between words are also captured. This is done simply by counting how words occur or found together in a corpus the other approach worth mentioning is the matrix of co-occurrence. It captures the amount of terms that occur together to capture all relationships between and words are captured. This is achieved by counting how words occur or are found together.

b) Prediction based embedding

Frequency-based methods have been used for many natural language practices, such as sentiment analysis and text classification. Nevertheless, after the introduction of word2vec (Mikolov T. K., 2010) to the NLP group, frequency-based approaches have been shown to have disadvantages, the frequency-based methods have been shown to have limitations. Such approaches contribute to the development of architectures on neural networks. Implementation for a concept of neural network architectures, vector computing. Such model architectures, according to authors, are very effective in maximizing computational output by optimizing the model's hidden layer, and boost efficiency was modeled with a simple layer of projection instead of hidden layer. (Singh, 2017)

One of the architectures proposed is a feed-forward NN like the language model of (Bengio, 2003) by removing the non-linear hidden layers. This model is known as a CBOW model. This method learns an embedding by predicting the target word based on nearby words. The nearby words are surrounding words which determine the context. Basically, in CBOW model, the average of the vectors of the surrounding words is given to the neural network

for predicting the target word, which appears in the output layer. The architecture is dubbed a bag-of-words model as the order in which words occur does not influence the prediction. Words before the target, history and words after the target, future, are evaluated (their vectors are averaged). The model architecture is shown in Figure 2.3.
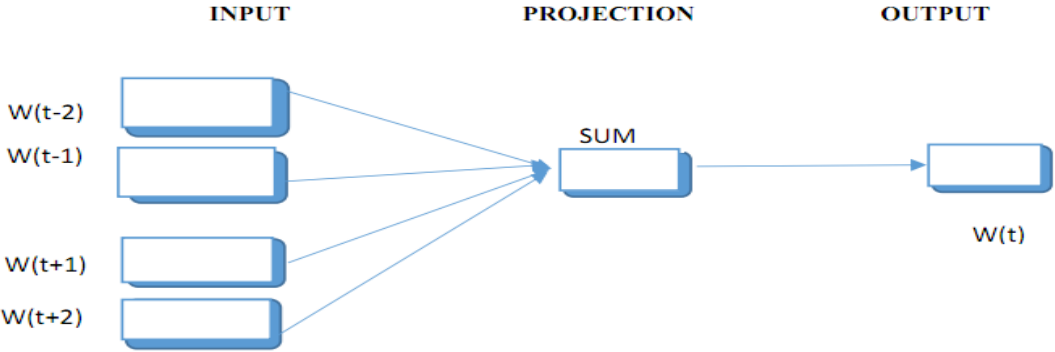
Figure 2. 3 CBOW architecture adopted from *(Mikolov T, 2013)*

(Mikolov S. K., 2013) SG's continuous model is the second architecture developed by This approach is nearly the opposite of the above process, but predicts the terms surrounding it rather than predicting the target phrase. This approach tries to predict the meaning, or words nearby, because of the target phrase. The continuous SG model takes up the word in the middle of a series of words. The architecture of continuous SG model is depicted at Figure 2.4
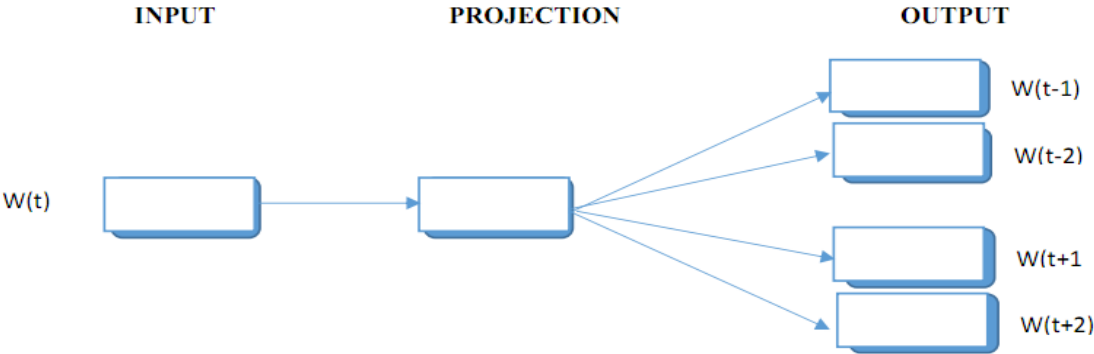
Figure 2. 4 Skip Gram architecture adopted from (Mikolov T, 2013)

CBOW model performs better in tasks involving small data sets between the two architectures as the model views the entire background as one observation, and smoothly

over the distribution data at the average level. On the other hand, for huge data collection, SG model is better and fine-grained, and eventually outperforms any other design. (Mikolov S. K., 2013) Proved that skip-gram works better.

2.8.1 Word Embedding Based Models

Several techniques for building word embedding models are proposed in literature (Basirat , 2018) The most common are word2vec, GloVe and fastText.

2.8.1.1 Word2vec

A shallow model developed by Word2vec (Mikolov T, 2013) was one of the first neural networks for successful word embedding training. Using the two popular neural models, it

learns low-dimensional vectors and predicts words based on their context: CBOW and SG. The implementation of these two-simple log linear models reduces time complexity, increase scalability and training efficiency. Word2vec starts with a series of random vector term this scan the dataset in orderly fashion keeping a background window around each word it's adjacent to word2vec uses target word and meaning to determine how they behave throughput the corpus. (Mikolov S. K., 2013)

The algorithm calculates the dot product between the target word and the background words and attempts to reduce the Stochastic Gradient Descent (SGD) metric efficiency. Every when in a similar context two words are met, their relation or spatial distance is enhanced. The more proof that two words are identical, the closer they will be when searching the corpus. (Pandey, 2020)

The problem here is that the model only provides positive reinforcement to make vectors closer. These leads, with a huge corpus at minimum state, to the state that all vectors would be concentrated in the same position. To address this issue Word2Vec initially proposed a Hierarchical SoftMax regulator at first then a Negative Sampling later. (Mikolov S. K., 2013). The latter is simpler and has been shown to be more effective. The basic premise is

that each time the distance between to vectors is minimized, a few random words are sampled and their distance to the target vector is maximized. This way, it is ensured that nonsimilar words stay far from each other.

2.8.1.2 GloVe

Glove is a well-known algorithm among word embedding models, such as word2vec. Glove goal is basically to construct word vectors that capture meaning in vector space and take advantage of global count statistics rather than just local information. Glove learns embedding based on word frequency through a matrix of co-occurrence and weight loss. (Jeffrey Pennington, 2014)

2.8.1.3 FastText

FastText is one of the models of text classification and word representation was developed by Facebook, using unsupervised techniques to make representations of words. It is word2vec extension that views the representation of words from a different angle. With a classification assignment, the issues of predicting meaning terms by Skip-gram can be discussed. Therefore, fast text model takes the real essence of word into account. FastText represents sentences as bag of words and train a classifier (AKLILU, 2019). One of the peculiar features of fastText is its ability to generate vectors for out-of-vocabulary words including unknown words and tokens. FastText considers not only the word itself but also groups of characters from that word and the sub word information such as character unigram, bigram, trigrams etc. during learning word representations

2.9 Related Works

Since it is important to know the current state-of-the-art in script identification it is crucial to conduct intensive literature reviews of previous studies both local and global will be conducted. Related literatures from different sources (books, journal articles, conference papers, the Internet) will be reviewed to understand the Multi label text classification, its development tools, techniques, procedures and methodologies and the challenges of MLTC.

2.9.1 Trends in Multi label Text Classification

(Duwairi, 2007) Has conducted a research on Arabic text categorization using three classifiers; namely, naïve Bayes, k-nearest neighbors, and distance-based classifiers. The performance of the classifier has been measured using recall, precision, error rate and fallout. The experimentation made on in-house collected Arabic text and result shows that the naïve Bayes classifier performs better than other two. Whereas (Mark HUGHES, 2013) has conducted a research Medical text classification using Convolutional neural networks researcher was present an approach to automatically classify clinical text at a sentence level using deep convolutional neural networks to capture complex features from text and also show automatically generate context based and rich representation of health information achieved remarkable result compare to static machine learning approaches.

(Mykowiecka, 2017) Use word2vec tool to change different parameters for tasks such as the detection of synonyms and analogies. They stated that word embedding can be used for linguistic analysis such as similarity and comparison for Polish terms and that the method's efficacy depends heavily on data set and parameter tuning. Arabic is one of the world's most widely spoken Semitic language. It has a strong connection to Amharic language. Several scholars have attempted to study and evaluate the embedding of Arabic word Several scholars have attempted to study and evaluate the embedding of the Arabic word. One is (Elrazzaz, 2017) on performing a methodical assessment of the Arabic word embedding.

They have performed both intrinsic and extrinsic evaluation on the embeddings. The word embedding of Italian, a highly inflective language spoken mainly in Italy, has been evaluated by (Tripodi, 2017) evaluating the efficiency of skip-gram and continuous bag of Italian language words, training taken from Italian Wikipedia. They took a word analogy test and evaluated the word embedding created word embeddings. The experiment is conducted by fine-tuning different hyper-parameters such as the size of the vectors, the window size of the word's context, the minimum number word occurrences and the number of negative samples.

(Beakcheol JangID, August 22, 2019) Has conducted a research on online news and Twitter he explores the performance of word2vec convolutional neural network to classify news articles and tweets into related and unrelated ones. Using two word embedding algorithms of word2vec, continuous bag of words and skip gram researcher construct CNN with the CBOW model and CNN skip gram he has experiment CNN without word2vec and with Word2vec experiments result shows word2vec significantly improve the accuracy and CBOW has greater performance than skip gram and CNN has good in news than tweets because typically news are more uniform than tweets . the CNN with CBOW accuracy was 93.51% whereas CNN with skip gram 91.61%.

(MD.ASLAM PARWEZ, 2019) Has conducted a research on Multi label classification of Microblogging Texts using CNN he explore applying CNN with word embedding compare with traditional machine learning approaches because they suffer with data sparsity curse of dimensionality in order to solve pretrained word embedding from generic and domain specific textual data source are used with the different channels of CNN layer word embedding are important to predict labels. Proposed model achieves 94.69 compare than SVM, Naive Bayes, Random Forest.

2.9.2 Multi label for low-resource Languages

Researches on Amharic Text classification have been done by many researchers. Different methodologies and approaches have been utilized and experimented. Relevant works of those researchers are presented here (Gambäck B. O., 2009) applied machine learning to Amharic text classification and examined the effect of operations like stemming and POS tagging on text classification performance for Amharic. They utilized a medium sized, hand-tagged Amharic corpus and found out that stemming has no significant influence on the performance of Amharic text classification in their work, bag-of-words approach was used for text classification experiment. The approach they utilized suffers a big sparsity which is resulted from vary nature of way bag-of-words do word representations.

(Worku, 2006 E.c) carried out a neural network approach on 9 categories with a total size of 1,583 items in the dataset using LVQ (Learning Vector Quantization) for an automatic Amharic news classification. The issues with LVQ is that processing required for classification takes time as more hidden units are often required. The authors also investigate the effect of increasing the number of categories and news items on the classification accuracy by TF and TF*IDF weighting schemes. As the experiment was conducted with three, six and nine news categories. On average he obtains result of 75.5% and 71.6% by TF and TF*IDF respectively.

(Surafel, 2003) , has conducted an experiment on the 11,024 Amharic news articles. In the experiments text preparation and preprocessing was done. Stop-words and rarely occurring words are removed from the collection. 66% of the data was used for training purposes and the remaining 33% was used for testing purposes. A naïve Bayes and k-nearest neighbor learning algorithms were used to categorize the Amharic news items. The results of the experiment show that best result achieved by naïve Bayes and k-nearest neighbor is on three categories (95.80% vs. 89.61%) and the least performance is obtained in 16 categories (78.48% vs. 64.50%). The reason for the drop of performance when the number of category increases is that when the number of categories increases the categories contain unevenly distributed data set. The researcher also reported that the naïve Bayes classifier has high performance for automatic Amharic news categorization.

(Yohannes, 2007) Has conducted a research on 69,684 Amharic news items. The researcher uses a text classification tools called WEKA. The authors measure the performance using the Logic Model Tree (LMT) and Support Vector Machine (LibSVM). The LMT is a type of decision tree learning algorithms available in WEKA tool. Both LMT and LibSVM classifier showed good performance accuracy; 79.72% and 81.15% in the 15 news categories respectively. However, the time required to build a model is very due to the large number of data set of the experiment.

(AKLILU, 2019) researches are done the first time in Amharic word embedding to classification purpose he assumes that as we know Word embeddings capture different

linguistic characteristics, which are intrinsic, such as word analogy, word similarity, out-of-vocabulary words and odd-word out operations. The author also uses to train classifiers using FastText, a recent method to generate and evaluate word embeddings was utilized. Gives benefit to capture sub word information. The result shows that in Multiclass text classification on the model attained 97.8% F1-score; result being fluctuated based on parameters.

Tigrinya, a very similar Semitic language to Amharic, spoken in Ethiopia and Eritrea, has been studied to boost POS tagger by its word embedding (Tedla, 2017) . As Tigrinya has very little support for annotated resources, the authors built a new text corpus and investigated the optimal hyper parameters for Tigrinya to generate word vectors. They showed that the dimension affects the quality of word relatedness semantics and syntactic While boarder context gives better semantics connectivity, shorter context makes syntactic connectivity.

As far as we know from the review different article, there has been no use of Amharic word embedding analysis for multi-label classification of Amharic text so far. Nevertheless, Amharic neural vectors were used as a function to develop the Amharic Named Entity Recognition System for downstream tasks such as NER, (Demissie, 2017)) used Amharic neural vectors as a feature to design Amharic Named Entity Recognition system. And another (AKLILU, 2019) are done on word embedding using model fast text approaches to Multiclass text classification he has the first researchers for Amharic word embeddings are do and capture semantics of words using distributional representation.

2.10 Evaluation metrics

For Word Embedding, In the NLP community there are two evaluation methods that are often used: Intrinsic and Extrinsic evaluation methods (AKLILU, 2019). Intrinsic evaluations are experiments in which word embeddings are evaluated based on human judgments or their visual results on words relations. Word semantic similarity and word analogy, are the most popular method of word embeddings evaluation.

In a multi-label text classification problem an example may be associated with set of labels therefore classification of an example may be partially correct or partially incorrect (A. Santos, 2011) .This can happen when a classifier correctly assigns an example to at least one of the labels it belongs to, but does not assign to all labels it belongs to. Also, a classifier could also assign to an example to one or more labels it does not belong to (Shweta C. Dharmadhikari). The evaluation measures for single-label are usually different than for multi-label. Here in single-label classification we use simple metrics such as precision, recall, accuracy. In single-label classification, accuracy is just:

$$\frac{1}{N}\sum_{i=1}^{N}[\hat{y}^{(i)} = y^{(i)}]$$

(2.4)

In multi-label classification, a misclassification is no longer a hard wrong or right. A prediction containing a subset of the actual classes should be considered better than a prediction that contains none of them, i.e., predicting two of the three labels correctly this is better than predicting no labels at all.

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}^{(i)} \wedge y^{(i)}|}{|\hat{y}^{(i)} \vee y^{(i)}|}$$

(2.5)

Micro-averaging & Macro-averaging

Micro-averaging and Macro -averaging was Label based measurement. To measure a multi-class classifier, we have to average out the classes somehow. There are two different methods of doing this called micro-averaging and macro-averaging. In micro averaging all TPs, TNs, FPs and FNs for each class are summed up and then the average is taken. (Nooney, 2018)

CHAPTER THREE: METHODOLOGY

3.1 Introduction

The methods, strategies, algorithms and tools used in this study will be discussed in this chapter. Since word embedding involves a training dataset, the method of preparing the dataset is presented in detail. Selected techniques are provided for the evaluation of the word embedding in Amharic. The work's general design and the representation of the expected results to solve the issues raise.

3.2 Word Embedding

Words are essential units of letter formed language. We reflect sounds that use a series of characters that people can easily understand. Words are the atomic units of syntax that cannot be subdivided into smaller units in linguistic syntax hierarchy (Basirat, 2018). Word2vec is the tool for generating the distributed representation of words, which is proposed by (Mikolov T, 2013). When the tool assigns a real-valued vector to each word, the closer the meanings of the words, the similarity that vectors imply. Distributed representation means assigning for each word a real-valued vector and representing the word by the vector. When representing a word by distributed representation, we call the word embeddings.

Word Embedding is a type of word representation that enables machine learning algorithms to understand words with a similar meaning. Technically speaking, using the neural network, probabilistic model, or dimension reduction on word co-occurrence matrix, this is a mapping of words into vectors of real numbers. We can use in language modeling and the techniques Word embedding is a way to perform mapping using a neural network.

Word Embedding is also called as distributed vector space or vector space model as s distributed semantic model or distributed represented or semantic space. We mean that

when come across word semantic as we read these names which means categorizing similar words together. For example, fruits such as apple, mango, banana should be put in close proximity while books are far from those terms. Word embedding will, in broader sense, create the fruit vector that will be replaced far away from book vector representations.

Word embedding assists in the generation of applications, clustering of documents, text classification and processing of natural language tasks. in terms of the following tasks. Frist, Compute similar words mean word embedding is used to suggest similar words to the word being subjected to the prediction model. Along with that it also suggests dissimilar words, as well as most common words. Second, create a group of related words mean that used for semantic grouping which will group things of similar characteristic together and dissimilar far away. Third, feature for text classification mean Text are mapped into arrays of vectors which is fed to the model for training as well as prediction. Text-based classifier models cannot be trained on the string, so this will convert the text into machine trainable form. Further its features of building semantic help in text-based classification and Document clustering is another application where word embedding is widely used.

Word2vec learns word by predicting the context around it. Let's take the word" he loves Basketball", for example. We would like to measure the word2vec for the word: loves. Suppose loves = Vin.P(Vout / Vin) is calculated where, Vin is the input word. P is the probability of chance. Vout is the word output. Word loves to move about every word in corpus. It encodes the syntactic relationship and the semantic relation between words. This helps to find words similar to those and analogies. We measure all random features of the word loves. These features have change or updated with the help of a back-propagation method regarding neighboring or context words. Another way to learn is that if the context of two words is similar, or two words have similar characteristics, then those words are related. Natural language processing: There are many applications where word embedding is useful and wins over feature extraction phase of features including parts of speech tagging, sentimental analysis, and syntactic analysis

Word relationship may be presented either morphologically or in the form of a matrix. The morphological analyzer determines the relationships between words in morphology (structure and forms of speech) Words are translated into a real valued number called vectors in the form vector that words captured and presented in a way that semantically related words come closer in space and have if similar vectors. Whatever terms are captured and interpreted in a space-related way would come closer and have similar vectors. Representing words is so good it can even recognize main relationships such as:

King - Man + Woman = Queen

It can decipher what a man is to king, a woman is to a queen. Using these models, the respective relationships could be established. Mathematically speaking, a vector is a number which specifies the position of one point in space relative to another. These defined by magnitude and direction. because vectors are physical quantities, they can be compared with each other in different ways. The two infamous methods are Euclidean and cosine angle distance. Although the Euclidean distance is the actual distance between vectors in N-dimensional space, the cosine distance is the angle between vectors in space. The Euclidean distance between two vectors $a = (a1\ldots, a_n)$ and $b = (b_1\ldots, b_n)$ is computed as:

$$\|a - b\| = \sqrt{\sum_{i=1}^{n}(ai - bi)^2} \tag{3.1}$$

The cosine of the angle between the above vectors is calculated using the two vectors dot and cross product, as shown in the formula below.

$$\cos\theta = \frac{\sum_{i=1}^{n} aibi}{\|a\|\|b\|} \tag{3.2}$$

From various objects such as images, videos, numbers and words, vectors can be created. Before the advent of the incorporation of vectors into NLP, the principles of phonemes, morphemes, syntax and semantics etc. were used to understand the word structure. Soon

after the advent of machine learning and deep learning, the need for words to be described so that machines can understand words together. Word vectors represent words as multi-dimensional continuous floating points numbers where semantically related words are mapped in special space to adjacent points. This representation of words using vectors lends the capability to calculate distance, be it Euclidean or Cosine, as a measure of the relationship between words. There are different word embedding models available such as word2vec (Google), Glove (Stanford) and fastText (Facebook).

3.3 Word2Vec Approaches

Word2Vec is a single-input, output, and hidden-layer neural network.one of the Keras python library offers an embedding layer which can be used on text data for neural networks. It requires that the input data be integer encoded, so that each word is represented by a unique integer. Using the tokenizer API also given with Keras, this data preparation steps can be performed. the embedding layer is initialized with random weights and will learn to embed all words in the training data set. There are two models CBOW and Skip-gram, as defined before.

Word2vec, proposed at Google by (Mikolov T, 2013) is a shallow, two-layered NN that is used in processing texts and word embeddings produce. It has a single hidden layer and a fully connected feedforward network. Non linearity of neural networks is eliminated in word2vec.According to (Mikolov S. K., 2013) this model for learning distributed representations of word reduces computational complexity induced by the non-linearity of hidden layers in typical language models such as Recurrent Neural Net language model.

The authors preferred data efficiency, simplicity, and accuracy of representation of words that can help guess linguistic relationship of words based on past appearances. It generates word embeddings of words by sliding a window over a large corpus of text. Word2vec appears in 2 flavors, the CBOW and the SG model These two model architectures pretty much are similar algorithmically except that their predicting method is opposite.

3.3.1 Continuous Bag-of-words Model

This model architecture learns an embedding by predicting the target word based upon neighboring words, called context. Specific words define its meaning. The background is defined by multiple terms for a given probe word, based on the window size. In reality, this model uses the terms surround it and attempts to predict the target words.

Given words $w1$, 2… w_{k1} CBOW model learns to predict all words w_k from their nearby words (wk-2... wk-1, wk+1... wk+2). Having regard to the phrase: "*I will have orange juice and eggs for breakfast*." and window size of 2, if the target word is juice, its neighboring words will be *(have, orange and eggs)*.Our input and target word pair would be *(juice, have),(juice ,orange),(juice, and),(juice, eggs)*. Note also that within sample window, proximity of the words to the source word plays no role. So, have, orange, and eggs will treat the same while training. What CBOW basically does is predicting the target word after the surrounding words. The idea is given a context for predicting, which words will appear most likely along the way.
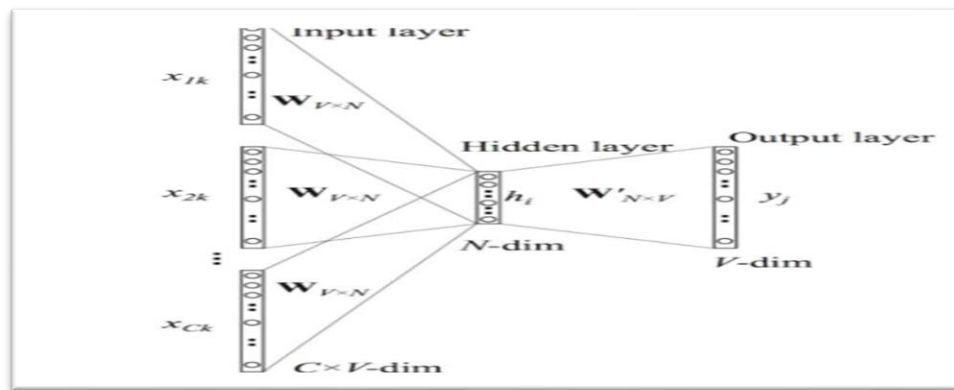
Figure 3. 1 CBOW model diagram (*adopted from:(Olejnik, 2017))*

The model diagram above (Figure 3.1) shows CBOW architecture in clear way. Let C be size of window; V be size of vocabulary. The input is 1-hot encoded context words {1, 2…} in an N-dimensional vector h, the output would be 1-hot encoded word y. The input vectors are fully connected to the hidden layer via a × weight matrix W and the hidden layer is connected to the output layer via a × weight matrix W'. This model is dubbed continuous

because it modifies the property of bag-of-words that doesn't matter the order of the words. Now it matters at CBOW It is important because the order of words is closely related to the context. And the context of words or sentences is the substance of every communication in NLP. On the other side, the model uses continuous distributed representation of the context (Mikolov T, 2013)

3.3.2 Continuous Skip-gram Model

This architecture is similar to the first CBOW (Mikolov T, 2013) but in the reversed way. It predicts the nearby words given the target word. Given words w1, w2, ..., Wi, Skip-gram model learns to predict nearby words of the current target word W_k. (Mikolov S. K., 2013). Considering our simple sentence from earlier, *"the quick brown fox jumps over the lazy dog"*. If we used the CBOW model, we get pairs of *(context window, target word)* where if we consider a context window of size 2, we have examples such as *([quick, fox], brown), ([the, brown], quick), ([the, dog], lazy)* and so on. Now given that the purpose of the skip-gram model is to predict the context from the target word, the model usually inverts the context and goals, and tries to predict each context word from its target word. Consequently, the task becomes to predict the context *[quick, fox]* given target word '*brown*' or *[the, brow*n] given target word '*quick*' etc. Thus, the model tries to predict the context window words based on the target word.
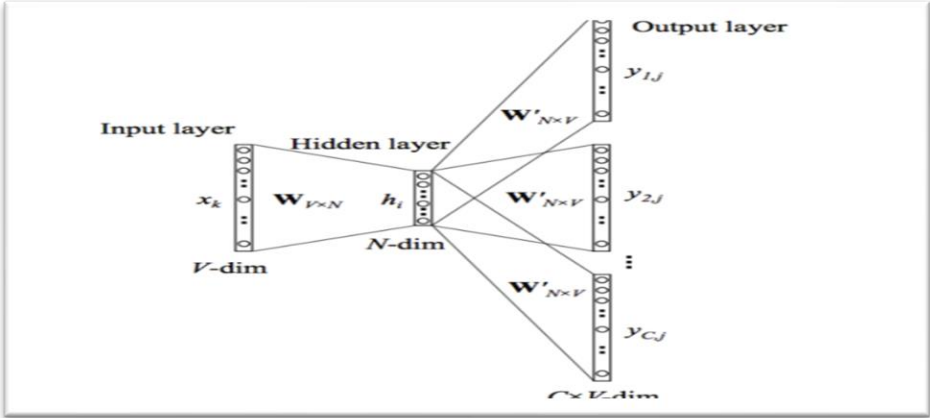
Figure 3. 2 Skip-Gram Model diagram (*adopted from:(Olejnik, 2017))*

We now need to model this Skip-gram architecture as a deep learning classification model for learning, so that we take the target word as our input and try to predict the context words. That becomes somewhat complex as we have multiple words in our context. We further simplify this by breaking down each (target, context words) consists of a single word only. Hence our earlier dataset is transformed into pairs (brown, quick), (brown, fox), (quick, the), (quick, brown) etc.

We feed our skip-gram model pairs of (X, Y) where X is our input and Y is our label. It is achieved by using [(target, context), 1] pairs as positive input samples where target is our word of interest and context is a context word that appears near the target word and the positive mark 1 means that this is a contextually important pair. We also feed in to [(target, random), 0] pairs as negative input samples where target is again our word of interest but random is just a randomly selected word from our vocabulary that has no meaning or connection with our target word. Therefore, the negative label 0 shows that this is a pair unrelated from meaning. We do this so that the model can then learn which pairs of words are important in context and which are not and generate similar embedding for semantically similar words.
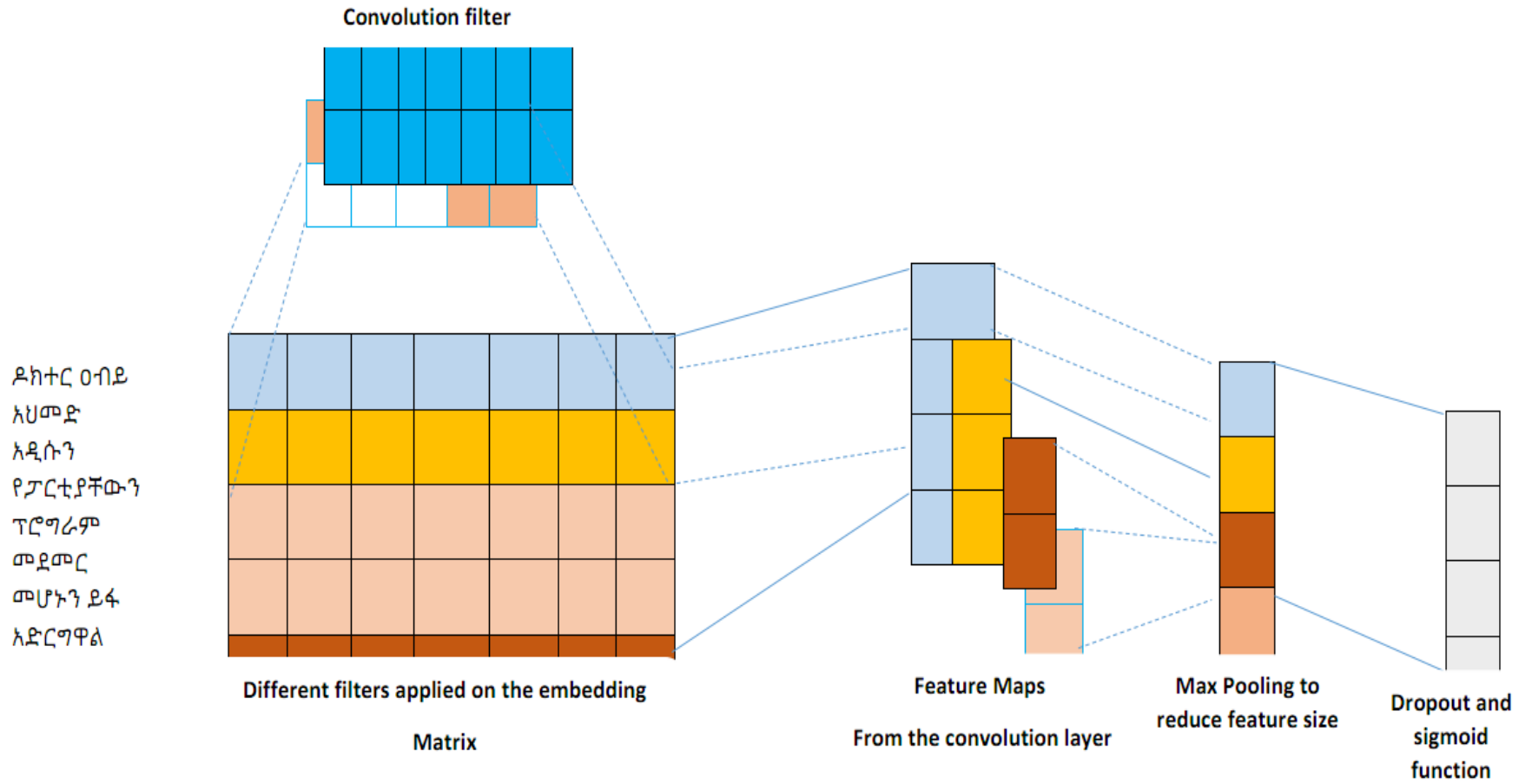
Figure 3. 3 Representation of inputs, outputs operations of CNN Models (adopted from (Kim, 2014))

3.4 System Architecture

The architecture of Multi label Amharic text classification has three basic components. These components are the pre-processing, word embedding, CNN training, and lastly the prediction module. The pre-processing module has the following sub components tokenizer, normalization, and stop-word removal. The word embedding component or module has basic sub component preparing pretrained word2vec that are domain specific and related news texts and selecting parameters. Finally loads to CNN layers as input. The CNN training module consists of different layers, choose optimizers, uses CNN as feature learning and adjusting parameters. The prediction module accepting the value from the trained model predict labels of new Amharic news text to achieve such task we use sigmoid function because it uses for multi label text classification. Sigmoid function outputs probability of each class and it allows for independent probabilities. The overall architecture of the model is shown in Figure 3.4 below.
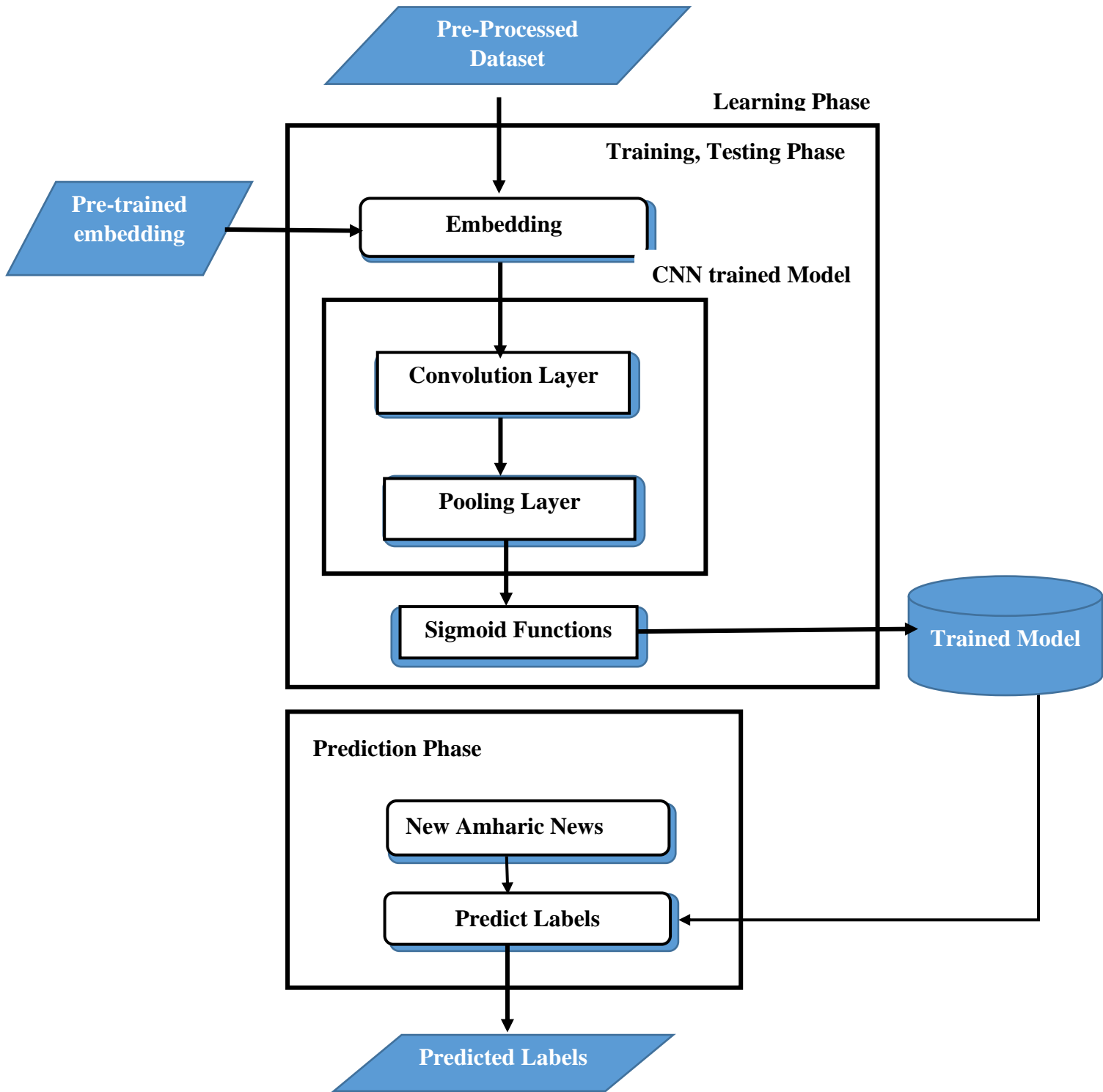
Figure 3. 4 Proposed architecture

3.4.1 Pre trained word embedding

The first step are pre-trained word embeddings, which is distributed representation of words as real-valued vectors learned from a text corpus using neural networks and techniques of matrix factoring. Those are low-dimensional, dense vector representations of words in a continuous embedding space where in semantically related words appear near each other and similarity represent vectors. In addition, word embedding preserves both semantic and syntactic words based on their contexts, and has proven to be very effective in NLP applications.

Word embedding got its popularity by the groundbreaking works of (Mikolov S. K., 2013) using word2vec based on two CBOW and skip-gram models in this research we use Continuous Bag of Words. In general, we are collected datasets from Amhara Mass Media Agency news document, for word embedding we collect 19,000 sentences which produce 32,242 word with their vectors, and in order to compare pretrained word embedding data with we are use word2vec file from internet which containing about 304,469 words and vectors. finally, for the purpose of text classification we use 1200 labeled sentences. For online down word2vec files free download form fastText for different languages in Amharic (*https://fasttext.cc/docs/en/crawl-vectors.html*).

3.4.2 Corpus for Amharic News Document

Generally, In NLP tasks role of Corpora is very important specially for Deep learning approaches and having label data has enormous impact on research, success or failure have depended on the quality of correct data. Raw data that is simple plain where the linguistic information is implicit is unannotated corpora. Annotated corpora, on the other hand, adds extra explicit information to the text such as categories, and labeled by the Experts.

The collected data are in the format of word files. Here example of our dataset which have one news consists of different thing such as day, time, header, footer and finally information source. We use only the news text as represent as body. Sample news format before preprocessing represent as such shown in figure 3.5 below.

በአማራ ክልል ጠንካራ የወባ መከላከልና መቆጣጠር ስራ በመስራቱ ወባ በወረርሽኝ መልክ እንዳይከሰት ማድረግ ከመቻሉም በላይ ስርጭቱ እየቀነሰ መምጣቱን የክልሉ ጤና ጥበቃ ቢሮ አስታወቀ።

የቢሮ ኃላፊው **ዶክተር አበባው ገበየሁ** እንደገለጹት ወባን ለመከላከል ልዩ ትኩረት ሰጥተው በመንቀሳቀሳቸው፣ የተጠናከረ የኬሚካል ስርጭትና የአጎበር ስርጭት በመደረጉ ወባ የሚያስከትለውን ጉዳት መቀነስ ተችሏል ብለዋል፡ : ሃገሪቱ እስከ 2030 ዓ.ም ወባን ሙሉ በሙሉ ለማጥፋት ብሄራዊ የወባ ማስወገድ ፍኖተ ካርታ አዘጋጅታ በመንቀሳቀስ ላይ እንደምትገኝም ተገልጿል።

የአማራ ክልል ወባን የማጥፋት መርሃ ግብር በደቡብና ሰሜን ወሎ በሰሜን ሽዋና በደሴ ከተማ አስተዳደር በይፋ ተጀምሯል።

መረጃውን ያደረሰን የደቡብ ወሎ ዞን የመንግስት ኮሙኒኬሽን ጉዳዮች ጽ/ቤት ነው።

Figure 3. 5 Sample news before preprocessing

3.4.3 Pre-processing corpus for Amharic News Text

Pre-processing involves planning the corpus or sample in a format appropriate for the process of training and evaluation. As mention earlier the dataset are in format of word *(docx.)* files. So, we had changed the word format in to *(. txt)* contains many sentences by the year. First removing header/footer part *(አቴቪ ....),* Punctuation marks, stop word Removal, normalization this done for all further work. And second import **(.txt)** to Excel and save it as file as **(. Csv format)**. Sample Python code to remove the header and change to text *(.txt)* format

```
In [2]:  import glob as g
         path = "C:/Users/Habtu/Preprocessing/Changed/"

         def checkATV(line):
             flag,ctr = False,0
             for word in line.split():
                 if(word.strip() == 'አቲቪ'):
                     flag = True
                     break
             return flag
         def checkLine(line):
             flag,data,ctr = False,"",0
             for word in line.split():
                 if(len(word.strip())>0):
                     data = data+word.strip()+" "
             if(len(data.split())<4):
                 flag = True
             return flag,data
```

Figure 3. 6 Sample snippet of code to remove ATV news Header's

The next phase is saving all sentences in single-word file, separating the sentences as (,) from each other and importing data to excel and saving label as (.**CSV**)
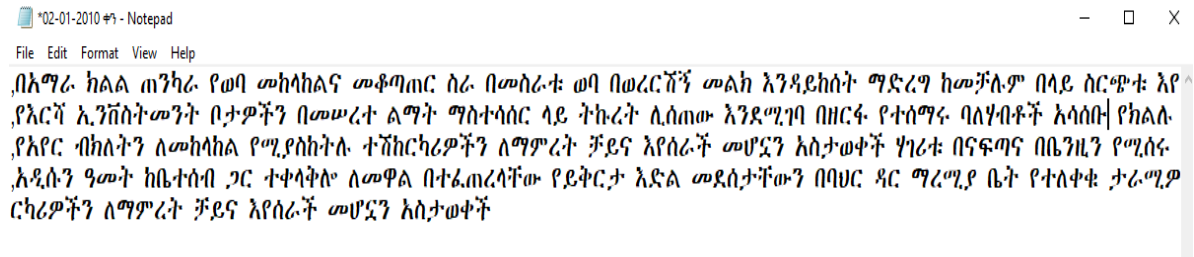
Figure 3. 7 Sample snippet converted to text dataset

Then the next step will be preprocessed step are followed first by removing Punctuation marks Amharic and English Punctuation marks, normalizing then removing Stop words from List of Amharic Stop Words finally save the preprocessed file in (. Csv format).

Normalization

In Amharic, there are different characters, which has the same sound and meaning. Representation of the same object or concept with them results in increase the size of feature. For example, the entity sun can be written in different words like ጸሃይ፣ ጸሐይ፣ ጸ ሀይ፣ ጸሓይ፣ ጸ ኅይ፣ ጸ ኃይ፣ ፀ ሃ ይ፣ ፀ ሐይ፣ ፀ ሀይ፣ ፀ ሐይ፣ ፀ ኅይ፣ፀ ኃይ but they did not have semantic difference according to (Alemu Kumilachew, 2017).

42

Algorithm 3.1: Character normalization

Input: raw Amharic news text in .txt format

Output: normalized character sequence

do

read a line of text from a fine and store on temp

for each character of temp

if character is in the normalized table

replace with the common character presented

endif

end for

while (not end of file)

To handle this problem, we prepare a list of Amharic compound words separated with a space. If the token read from the file is in the compound word list and followed by „-‟ or space and the next word is also in the list with the same index, then remove the space or hyphen and merge them.

Algorithm 3.2: Compound-word normalization algorithm

Input: two consecutive strings separated with any of Amharic word separator

Output: combined form of two strings/tokens

do

read words from files

if sequence of words exists in the compound word list

if the string next to token is also in compound word list and their index is equal

delete "-", space

token =token + next string

end if

end if

while (not end of file)

The shorthand/acronyms/abbreviation representation of words like ጸ/ቤት፣ ት/ቤት etc is replaced with the expanded words to normalize abbreviations. The algorithm used to normalize abbreviations is illustrated as follows

Algorithm 3.3: Abbreviation normalization

Input: abbreviated token

Output: extended form of the abbreviated word

do

    read token from files

    if token is in short form list as key

      replace with the corresponding long form list value

    end if

while (not end of file)

Tokenization

Tokenization is the process of identification of all the individual words of a document. In Amharic, punctuation marks and spaces are used to indicate the beginning and ending of tokens. Tokens can be defined as the last chopping pieces" of document unit.

Algorithm 3.4: Tokenization

Input: Line of string

Output: token list

read from a file

do

  read character sequence from a file

  if character is space, newline, ፤, ፣, ፥, ።, tab, carriage return or.

    assign character sequence before the characters to token

  end if

while (not end of file)

Stop word Removal

Tokens that contain numbers or digits are also considered as stop-words in our case. The stop-words list used in this study are collected from Amharic document processing works done previously

Algorithm 3.5: Stop-word removal

Input: token list

Output: list of non-content baring tokens

do

   read token from list of all features

   if token in stop-word list and contains digits

      remove token from features list

    end if

while (not end of file)

This module accepts list of words and then removes the stop-words. This process removes most frequently occurring words from the document that do not change the meaning of the document. In this thesis, two kinds of stop words were identified: news specific stop-words such as አመልካቷል, አስታውቋል, ገልጸዋል, ታውቋል, etc. and common stop-words, which are manually identified from the collection.

3.4.4 Feature extractor

As feature extractor the CNN are taken the word and word vectors from word2vec will be input to feature to extract. The main operation in our feature extractor is lookup operation. It takes tokenized words from preprocessing stage and retrieves corresponding feature vector from the output of Word2vector tool. For training purpose, it reads tagged words and retrieves their word vector, then combines it with its labels for training. For plain text it also reads tokenized words and retrieves their feature vector.

3.4.5 CNN training phase

Nowadays, the use of feature vectors, such as word embedding's, as an input to neural networks for text classification and clustering has shown a remarkable performance gain. In this paper, we present neural network models for multi-label classification of Amharic News data. The proposed models are based on convolutional neural network (CNN) architectures, which utilize pre-trained word embeddings from generic and domain-specific textual data sources. The word embeddings are used individually and in various combinations through different channels of CNN to predict class labels.

After doing preprocessing phase next will be feed to Words of preprocessed corpus and dataset to word2vector to generate word vectors using algorithms of CBOW model. CNN have successfully in image, computer vision area but now also success in text classification area. Instead of image pixels, the input to most NLP tasks are sentences or documents represented as matrix. Each row of matrix corresponds to one taken, typically a word, but it could be a character. That is, each row is vector that represent a word. Typically, these vectors are word embeddings (low dimensional representations) such as word2vec, Glove, but they could also be one -hot vectors that index the word into a vocabulary. For a 10-word sentence using a 100-dimensional embedding we would have a 10X100 matrix as our input.

The output of the convolution operation is termed as feature map. Convolutional networks usually comprise three stages in which the layers first perform several convolutions in parallel, then make a non-linear transformation of convolution output using non-linear activation function, and finally capture important features using pooling function without concerning about the position of the features in non-linearly transformed output. Such networks recognize local features from the input that is helpful for prediction, and combine them to form a vector, which is further used to predict labels. During the training phase, the gradients from the loss function are propagated back and used to tune the parameters of the network.

Following are description of each layer in training phase to learn the characteristic features that characterize news documents which are categories in which. In addition to the layers or functions depicted in the proposed CNN approach (Figure 3.11), A convolutional neural network (CNN) is a deep learning architecture that is commonly used for hierarchical document classification. Although originally built for image processing, CNNs have also been effectively used for text classification. These convolution layers are called feature maps and can be stacked to provide multiple filters on the input. To reduce the computational complexity, CNNs use pooling to reduce the size of the output from one layer to the next in the network. Different pooling techniques are used to reduce outputs while preserving important features. (Scherer & Behnke, 2010)

We are concerned about one-dimensional convolutions where the input text is mapped into a sequence of embedding vectors corresponding to word sequences of the text. The convolution layer moves a sliding window of size k (i.e., convolution filter of width k) through the sequence of word embedding vectors, and performs linear transformation accompanied by the introduction of non-linearity using non-linear activation function to capture indicative information. Corresponding to each window, the pooling operation picks only that information of different types that are useful for prediction.

There are many variants of pooling operations. Max pooling operation is one of the most extensively applied pooling operations, which selects the max value from each type of feature maps, whereas the other pooling called average pooling takes average instead of picking the max value. (MD.ASLAM PARWEZ, 2019) Besides these pooling operations, there are dynamic pooling in which feature maps are split into distinct groups, and pooling is performed on each group, and then the pooled values are concatenated to form vectors. Similarly, there is a hierarchical pooling in which successive convolution and pooling operations are alternately performed, and k-max pooling wherein k maximum values are pooled from the feature map. The final layers in a CNN are typically fully connected. In general, during the back-propagation step of a convolutional neural network, both the weights and the feature detector filters are adjusted. So, we used for the CNN architecture

for text classification which contains word embedding as input layer 1D convolutional layers, 1D pooling layer, fully connected layers, and finally output layer.

Fully connected layer

Neurons in this layer are fully connected to all neurons in the previous layer. Fully connected layers are used to compute class scores that will be used as output of the network (Adam, 2017)) We have applied fully connected layer before the classifier (usually Sigmoid functions) is applied. We have applied one fully connected layer to compute the final output probabilities for each class before applying to the Sigmoid Function.

Activation layer

The activation layer is not a technical layer since no parameters are learned in activation layers. The output of the activation function is always the same as the input dimension since activation function is applied in an element-wise manner (Adrian, 2017). Hence, the width, height, and depth of the output layer is the same as the width, height, and depth of the input layer respectively. After a linear layer fully connected layer, it is common practice to apply nonlinear activation functions. Rectifying linear unit (ReLU) is the most widely used activation function in CNNs (G. Ian, 2012)

Adam Optimizer

The Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Straightforward to implements and computational efficient, limited memory requirements Adam is different to classical stochastic gradient descent. Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training.

Sigmoid Function

Sigmoid Function are used for multi label text classification the sigmoid function is another logistic function that has a characteristic "S-curve", or a smoothed-out version of a step function. Sigmoid are often introduced into neural nets to provide non-linearity to the model and are typically used for clustering, pattern classification, and function approximation.

Unlike SoftMax which gives a probability distribution around k classes, sigmoid functions allow for independent probabilities. When looking at a sigmoid function as a neuron in a neural network, input values of a sigmoid neuron can be any value between 0 and 1 and the output is the sigmoid function. These classification tasks are not mutually exclusive and each class is independent. Therefore, Sigmoid function allows for multi-label classification whereas SoftMax was good for multi-class classification the output of the final fully-connected layer is given as input from sigmoid function was 6-Way used for news categorize. (Usman Malik, 2020)

Early stopping

One of the forms of regularization use to evade overfitting is early stopping by training the model with a repetitive method. Such an approach improves the performance of data. There are many hyperparameters, one of which is how many times the full passes of data sets need to be used. Such parameters help to tune the accuracy of this approach. In text processing tasks, we are concerned about one-dimensional convolutions where the input text is mapped into a sequence of embedding vectors corresponding to word sequences of the text.

Generally, we construct CNN text classifier after doing Word Embedding of the given document and the overall section of the proposed CNN Model architecture has been elaborated in the following section 3.6
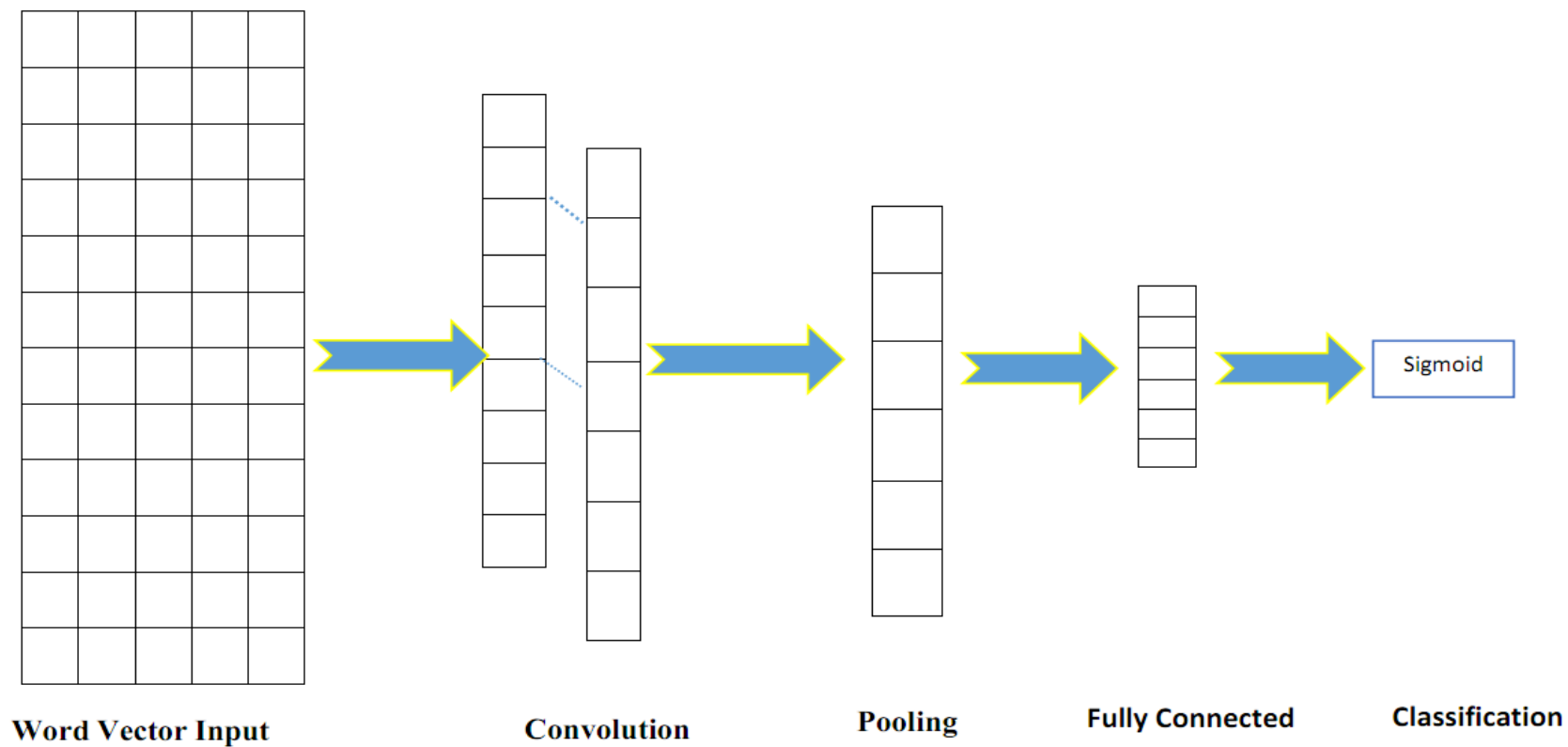
Figure 3. 8 ML Amharic Text Classification using CNN

3.5 CNN proposed model architecture

3.5.1 Input layer

The input to the model is the word sequence using array of text before doing such process applying tokenization and eliminating punctuations, symbols, numbers, and stop word removal needed. The cleaned tokens are presented in this layer as input and then mapped in the next layer to corresponding word vectors using different word embedding.

3.5.2 Embedding layer

Embedding Layer, for text classification problems, is a special component of the CNNs. A n embedding layer's purpose is to convert the text inputs into an appropriate form for the C NN. Here, every word in text document is transformed into a fixed-size dense vector. The embedding layer uses different embedding to map input token sequences to word vectors. Word embedding provides word representation that can be fed to the convolution layer that learns to figure out text classes from these input vector sequences. A cleaned text with k words / tokens is represented by a sequence of k vectors where vectors suit pre-trained w vectors. We made the length of news equal by padding zero values form a news text matrix of k x n dimensions with k tokens and n-length embedding vector.

3.5.3 Convolutional Layer

Convolutional Layers, are major components of the CNNs. A convolutional layer consists of a number of kernel matrices that perform convolution on their input and produce an output matrix of features where a bias value is added. The learning procedures aim to train the kernel weights and biases as shared neuron connection weights.

The convolution layer applies filters of different width to the news embedding matrix to extract distinctive features as vector corresponding to each filter and creates a feature map. Convolving the same filter through embedding of every word of a news extract features that

are independent of word positions. Filters at higher layers capture syntactic or semantic associations between phrases that are far apart in a news text. The ReLU is rectified linear unit function applied to a layer to inject non-linearity to the system by making all the negative values to zero. It helps in fast training of the model without making any significant difference in accuracy. (Kumar Shridhar, 2020)

3.5.4 Max Pooling Layer

Pooling Layers, are also integral components of the CNNs. The purpose of a pooling layer is to perform dimensionality reduction of the input text features. Pooling layers make a subsampling to the output of the convolutional layer matrices combing neighboring elements. The most common pooling function is the max-pooling function, which takes the maximum value of the local neighborhoods. These max-pooled feature values are then concatenated to compose a new higher-level feature vector, which acts as input to the next layer. The major benefit of including such max-pooling layer in the network is that it reduces the number of parameters or weights, and controls overfitting. (MD.ASLAM PARWEZ, 2019)

3.5.5 Dropout Layer

The dropout layer (Nitish Srivastava, 2014) drops out a random set of neurons by keeping neurons with some probability during the training phase. Dropout has the effect of making the training process noisy, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs.

3.5.6 Dense Layer

It is the last layer of the CNN architecture. The outputs of the dropout layer of the network are fed into the fully connected dense layer, which outputs the class probability distribution. The class probabilities are computed using the Sigmoid activation function Binary cross

entropy loss function is applied to train the classier that helps to interpret labels for categorization of news into different categories.

The parameters are updated using Adadelta optimization algorithm. Once the training is complete, the parameters and learned weights are saved into a file so that it can be loaded later and can be used for prediction of classes of unlabeled news. Fully-Connected Layer, is a classic Feed-Forward Neural Network (FNN) hidden layer. It can be interpreted as a special case of the convolutional layer with kernel size 1×1. This type of layer belongs to the class of trainable layer weights and it is used in the final stages of CNNs.

3.6 CNN Testing Phase

The testing phase follows the same procedure as we follow in the training phase. The difference lies in the dataset we are going to apply. We follow the same procedure and the same technique starting from the preprocessing of the data. If we follow any other way, it will lead us to incorrect prediction since the network may be presented with patterns (inputs) it cannot categorize. Similarly, the feature learning is also done in the same manner as in the training phase by using the learning model constructed from the training phase. The testing process and dataset is used to know how well our network predicts the new news text sentences for class of multi label probabilities.

3.7 Trained Prediction Model

Prediction is done by using the knowledge from the learning model, which is constructed by using the training of samples. The training dataset is used in the training phase. By using the knowledge constructed in the learning model, we can classify or predict News text in the testing dataset into a specific label class. The prediction model is the final trained model from the training. The input for prediction is the output of feature extractor which is a file containing the word vector of words in a given plain Amharic text. By taking this input it predicts the labels text classification of a word, therefore the output is target words with their predicted labels.

CHAPTER FOUR: EXPERIMENTS AND RESULT DISCUSSION

4.1 Introduction

In this study an attempt is made to construct a model for Multi label Amharic news text classification process. Detail implementation procedure, dataset preparation and experimental results are presented below.

4.2 Data sets

Since there is no publicly available Amharic document corpus for evaluating the designed classifier, we have collected 1200 Amharic news text from 6 major news categories. Six The news is collected from the Amhara Mass Media Agency from year 2010-2012 E.C. all news documents in the dataset are Multi labeled and also use for pretrained word embedding after preprocessing the news contents represents with corresponding labels. Here are sample list top five sentences. All news documents are represented as such because of multi label dataset. Four categories are *('Economy', 'Social', 'Agriculture', 'Politics')* whereas six categories *('Economy','Social','Agriculture','Politics','Governance','Development').*

	id	Contents	Eco	Soc	Agri	Pol	Gov	Dev
0	1	የምስራቅ ኢሲያና አውሮፓ ሃገራት የርስበርስ የኢኮኖሚ ትስስራቸው እየተጠናና...	1	0	0	1	0	0
1	2	የምዕራብ ጎጃም ዞን አርሶ አደሮች የገበያ ትስስር እና ወቅቱን የጠበቀ የ...	1	0	0	0	1	0
2	3	ለወጣቶቹ የስራ እድል አለመፈጠር የኢኀጋጋይነት መንፈስ መንደል የአመራር ...	0	0	0	0	1	0
3	4	በህክምናው ዘርፍ ያለው የተዛባ መዋቅር ሰራችንን በኦግባቡ ለመስራት አላስ...	0	0	0	0	1	0
4	5	የኦሞራ ምርጥ ዘር ኢንተርፕራይዝ በቻግኒ የምርጥ ዘር ብዜት ጣቢያ አምስት...	0	0	1	0	0	0

Figure 4. 1 multi label categories in data sets representation (. csv format)

(The news text sentence annotation was done like this if one sentence is member of class label it represented 1 whereas if the sentences are not class label is represented 0.) based on Expert decision.

4.2.1 Data preparation

A manually prepared and labelled datasets is used to build the CNN classification model. data preprocessing including Stop words, numerals, punctuation marks and symbols were removed and cleaned prior to the training and testing phases during data preprocessing. The line has a label row, followed by the data / document corresponding to it. And make embedding of pre trained word from related documents to word representation before feed to input layer.

Figure 4. 2 Count multi label categories in data sets

The number of multi label in dataset was maximum is 3 from six classes. One sentences may be three label or two label or one label. Graph shows count which categories belong to other label represent as 1 whereas 0 means the categories which are one label of each categories count in corpus.

4.3 Development tools

Many development tools are used in the process of this research. The tools that has been used include Jupyter Notebook, Tensorflow deep learning library, Keras deep learning library, Genism library, Scikit learn machine learning library.

4.4 Amharic Word Embedding code Parameters

The idea of embeddings in a nutshell is to have a compact representation of words *(and sometimes categorical variables)* that is learned from the context while training the network (Koehrsen, 2018).Two option to use word embedding first one train your network to learn this representation of words (better to have large dataset). Second use existing pre-trained embeddings (Glove, Word2Vec). In the research use those option. To make word embedding specify the input_dim which is the number of words/features in our sequences. and output_dim which is the number of dimensions to use to represent each word. dimension most commonly (300 dimensions used). The word embedding parameter we are used in our code are shown as table 4.1

Table 4.1 shows parameter we used in word embedding

Parameters	*Value*
Embedding dimension	**300**
Min_count	**2**
Workers	**4**
Window size	**10**
Sg	**0 (CBOW)**

As the table 4.1 shown have different parameters that we used in pretrained word embedding first word embedding have 300 dimensions as elaborate in above description that means the length of dense vector to represent each token(word). Whereas min_count means we are used 2 as min_count value. Min_count was count of words to consider when training the model words with an occurrence less than this count will be ignored. Other parameters were workers which we used in our code was 4. Workers mean that the number of threads use while in training depends on the performance of computer. Last parameters are Sg (0) this are parameter that specify whether use continuous bag words model from word2vec or Skip

gram. If value is 1 means that algorithm use skip gram model. As discussed from section 3.3.1 we are select CBOW model.

```python
In [15]: import gensim
         from gensim import models
         from gensim.models import Word2Vec
```

```python
In [16]: EMBEDDING_DIM=300
         model = gensim.models.Word2Vec(sentences=review_lines, size=EMBEDDING_DIM, window=10, min_count=2, workers=4,sg=0)
         words=list(model.wv.vocab)
         #print('vocabulary size: %d' % len(words))
         #model=gensim.models.word2vec(sentences=review_lines, size=EMBEDDING_DIM, window=4, workers=4, min_count=1)
         #words=list(model.wv.vocab)
         #Note:sg: (default 0 or CBOW) The training algorithm, either CBOW (0) or skip gram (1)
```

Figure 4. 3 Snippet code Embedding Model

```python
In [14]: # apart from the input sentence, the only additional paramter
         # we'll set is to specify use all possible cpu to train the model
         workers = cpu_count()

         start = time()
         word2vec = Word2Vec(review_lines, workers=workers)
         elapse = time() - start
         print('elapse time:', elapse)

         # obtain the learned word vectors (.wv.vectors)
         # and the vocabulary/word that corresponds to each word vector
         word_vectors = pd.DataFrame(word2vec.wv.vectors, index=word2vec.wv.index2word)
         print('word vector dimension: ', word_vectors.shape)
         word_vectors.head()

         elapse time: 0.1348111629486084
         word vector dimension:  (647, 100)
```

Out[14]:

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93
ወረጻ	-0.006447	-0.097675	0.039956	0.021999	-0.016908	-0.000015	0.003286	0.024857	-0.040488	0.008257	...	-0.023661	-0.036241	-0.002930	-0.043919	-0.
ኮ	-0.003341	-0.107761	0.043325	0.023627	-0.027852	0.004452	0.009003	0.030131	-0.038465	0.011247	...	-0.025955	-0.043545	-0.001635	-0.048808	-0.
ኮተኛ	-0.008807	-0.110914	0.041733	0.024686	-0.022023	0.006769	0.005020	0.029439	-0.042415	0.010258	...	-0.028126	-0.040549	-0.002999	-0.052779	-0.
ለሞቱ	-0.005016	-0.103331	0.044837	0.024828	-0.022456	-0.001840	0.005535	0.023909	-0.043270	0.012015	...	-0.023989	-0.047823	-0.004398	-0.049385	-0.
�	-0.003024	-0.108926	0.043655	0.023276	-0.028621	0.008106	0.003656	0.032627	-0.037636	0.013288	...	-0.027843	-0.050453	-0.004719	-0.049843	-0.

5 rows × 100 columns

Figure 4. 4   Snippet Sample Learned Vectors

```python
In [25]: word2vec.wv.most_similar(positive=['ያጠኑበት'], topn=4)
```

```
Out[25]: [('የጓንጭ', 0.9994049072265625),
          ('በመጠቀም', 0.9994038343429565),
          ('ያጠኑበትያዊ', 0.9993935823440552),
          ('ተጠና', 0.9993822574615479)]
```

Figure 4. 5 Sample snippet Word Similarity

Finally output of word embedding are save as vector (.vec) file format and here are sample

Figure 4. 6 Sample snippet Amharic pre trained Word Vector

4.5 Experimental scenarios

The Experimental Scenarios are divided in to three-part Preprocessing Data set, Word Embedding, proposed CNN implementation. The first steps are preprocessing like steps of tokenization, stop word Removal, creating sequences and padding. Output of such process consider as Input layer and consider as preprocessed dataset prepare, load pretrained word embedding. Secondly, Word embedding was converting text to vector form here is main work because good representation of word in distributional manner have effect on research work totally. So, we proposed to use CBOW model approaches of word2Vec and prepare pre-trained word embedding from different related documents have given an advantage to contain more words in embedding vocabulary.

Finally Experiment done to apply Proposed Multi label Amharic News using CNN Approaches by accepting input form Embedding layer combine and labeled training dataset learn, classify the news document to their label. And evaluation performance. The first and second group of experiments to answer Frist research question. Whereas third group of experiments are trying to answer research question two and three.

4.6 Training CNN Model Parameters

For training, we considering Amharic news document at sentences level with an input size of MAX_NB_WORDS = 10000. A dropout rate of 0.15 was used to regularize the network parameters with training epoch typical value 15.

Model Parameters: during designing of the neural network the number of different parameters of the network need to be decided. The model parameters that are required for the proposed network are described below: Number of neurons in the hidden layer: number of processing unit or nodes in the hidden layer.

Learning rate: -training parameter that controls the size of weight and bias changes in learning of the training algorithm. [0.001]

Batch size: the number of training instance per batch. The typical value depends on the training data. [8]

Adam Configuration Parameters

Adam are use as default algorithm and here is Adam Configuration Parameters that are using in model training part. Frist one **alpha**. Also referred to as the learning rate or step size. The proportion that weights are updated [0.001]. second **beta1**. The exponential decay rate for the first moment estimates [0.9]. **third beta2**. The exponential decay rate for the second-moment estimates [0.999]. This value should be set close to 1.0 on problems with a sparse gradient. Fourth **epsilon**. Is a very small number to prevent any division by zero in the implementation. [1e-8]

The work of training is to find the best weight for the deep neural network at which the network produces high accuracy or a very small error rate. The outcome of any deep model neural network somehow depends on how the model was trained and the number of layers

Epoch: determines when training will stop once the number of iterations exceeds epochs. When training by minimum error, this represents the maximum number of iterations. [14] The following table 4.2 shows what are paramter we used in CNN while in training as summary all thing are expressed in above statements.

Table 4. 2 shows hyper parameters for CNN training

Parameters	*Value*
Word embedding dimensions	300
No of filter size	64
Batch size	8
No of epochs	15
Learning rate	0.001
Dropout	0.15
Pooling size	2

```
In [19]: print('Building CNN')
         model = Sequential()
         model.add(Embedding(nb_words, embed_dim, weights=[embedding_matrix], input_length=max_seq_len, trainable=True))
         model.add(Conv1D(num_filters, 7, activation='relu', padding='same'))
         model.add(MaxPooling1D(2))
         model.add(Conv1D(num_filters, 7, activation='relu', padding='same'))
         model.add(GlobalMaxPooling1D())
         model.add(Dropout(0.15))

         #model.add(Dense)
         model.add(Dense(32, activation='relu', kernel_regularizer=regularizers.l2(weight_decay)))
         model.add(Dense(6, activation='sigmoid'))
         #model.add(Dense(num_classes, activation='softmax'))
         #model=Model(input)
         adam = optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
         model.compile(loss='binary_crossentropy', optimizer=adam, metrics=['accuracy'])
         model.summary()

         # summarize
         """from keras.models import Model
         print(model.summary())
         plot_model(model, show_shapes=True, to_file='multichannel.png')
         return model"""
```

Figure 4. 7 Building CNN layer model

Usually, the model is created with the certain number of layers, and entire layers are being involved in the training phase. The layer-wise training model starts with adding one layer of convolutional and pooling layer, followed by fully connected layer and applies the back-propagation algorithm to find the weights. In the next phase of the layer-wise training

model, the next layer of convolutional, pooling layer is added and the back-propagation algorithm is applied with previously found weights to calculate weights for the added layer. After adding entire layers, a fine tuning was performed with the complete network to adjust the entire weights of the network on a very low learning rate. The back-propagation algorithm starts with some random weights, and during training it sharpens the weighs by updating them in each epoch. The layer-wise training model provides nice rough weights initially as the network starts with first layers and, further, it adds remaining layers to find the weights for remaining layers.

```
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
_____
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 50, 300)           2673300
_____
conv1d_1 (Conv1D)            (None, 50, 64)            134464
_____
max_pooling1d_1 (MaxPooling1 (None, 25, 64)            0
_____
conv1d_2 (Conv1D)            (None, 25, 64)            28736
_____
global_max_pooling1d_1 (Glob (None, 64)                0
_____
dropout_1 (Dropout)          (None, 64)                0
_____
dense_1 (Dense)              (None, 32)                2080
_____
dense_2 (Dense)              (None, 6)                 198
=================================================================
Total params: 2,838,778
Trainable params: 2,838,778
Non-trainable params: 0
```

Figure 4. 8 CNN model summary

4.7 Test Result

Different performance metrics have been used to evaluate the performance of the proposed solution or model. Precision, recall, and f1-score are used extensively for measuring the performance of the proposed model. Besides, we have also calculated the micro-average, macro-average, and weighted-average for all the aforementioned performance metrics.

Precision: is the proportion of true positives against the whole positives. Mathematically, it is expressed as:

$$Precision = TP/TP + FP \qquad (4.1)$$

Recall or sensitivity: is the proportion of true positives against the whole true or correct data. It quantifies how well the model avoids false negatives. It is also known as true positive rate or hit rate.

$$Recall = TP/(TP + FN) \tag{4.2}$$

F1-score: is the weighted average the precision and recall. The relative contribution of precision and recall to the F1-score are equal. (Mequanent Argaw, 2019)

$$F1 - score = 2 * (precision * recall)/(precision + recall) \tag{4.3}$$

Micro-average, macro-average, and weighted-average for all the aforementioned performance metrics can also be calculated and used for additional analysis of results. Macro-average precision or recall is just the average of the precision and recall (respectively) of the model on different classes.

$$Macro - average\ precision = \frac{P1+P2+\cdots+PN}{N} \tag{4.4}$$

$$Macro - average\ recall = (R1 + R2 + \cdots + RN)/N \tag{4.5}$$

Micro-average precision or recall is calculated by summing up the individual true positives, false positives and false negatives for each class.

$$Macro - average\ percision = \frac{TP1+TP2+\cdots+TPN}{(TP1+TP2+\cdots+TPN)+(FP1+FP2+\cdots+FPN)} \tag{4.6}$$

$$Macro - average\ recall = \frac{TP1+TP2+\cdots+TPN}{(TP1+TP2+\cdots+TPN)+(TN1+TN2+\cdots+TNN)} \tag{4.7}$$

Experiment 1: was done to answer ***RQ1***: - the experiment aims to test what the categories are increase the accuracy also increase. Two experiment have been done, first on pretrained and second on word2vec downloaded files. The result shown as table 4.2

Table 4 3 shows accuracy performance of word embedding

	Accuracy
CNN_word2vec	**97.39%**
CNN_pretrained_emmbeding	**97.69%**

As seen in the table 4.2 the CNN_pretrained_embedding was good perform that downloaded file word2vec because it was pretrained in the area. This implies the pre trained embedding as features for CNN have high accuracy compare with not pretrained word embedding these experiments are done on all text classification datasets we have 1200 news sentences with six labels.

Experiment 2: was done to answer RQ2. The experiment aims to test what the label categories are increase the accuracy also increase in general as compare to traditional machine learning algorithms. Two experiment has done first was done no of label are 4 and second no of label was 6. As we describe earlier in section 4.1 generally, we have six major classes that share multi label properties those are *('Economy', 'Social', 'Agriculture', 'Politics','Governance','Development')* for second experiment we use all six labels whereas the four labels *are('Economy','Social','Agricultures','Politics')* are used in first experiment consideration the accuracy achievements are shown as table 4.3

Table 4 4 shows accuracy performance of proposed model different no labels

	No labels in news sentences	
	4	**6**
CNN_word2vec	98.36%	97.39%
CNN_pretrained_emmbeding	97.10%	97.69%

An experiment was done with the different number of label and data size using neural word embeddings on CNN pretrained embedding was achieve good result. The results in Table 4.2 and Table 4.3 show that neural word embeddings can be used to develop Amharic text classification. Even if a performance improvement through using word embedding from domain perspective are more important for desire result observed from this experiment, the next table shows that increasing labels are increase also performance. The two experiment has done first on 800 news datasets with label 4 and secondly with 1200 news dataset with label 6. The proposed CNN model obtains 97.69% training accuracy with testing accuracy 88.75%.

4.8 Findings and discussion

In this section, the performance of the CNN architecture was investigated for training and validating Amharic Text classification. ConvNets have a large set of hyper-parameters and finding the perfect configuration for your problem domain is a challenge. Different configurations of the proposed network were explored and attempted to optimize the parameters based on the validation and training set accuracy. From the collected dataset 20% for testing, 10% for validation and 70% for training the proposed convolutional neural network architecture discussed earlier was used the training and validation datasets are evenly distributed over the underlying 6 classes. For the multi label text classification are two major options to handle using single dense output layer and multiple dense output layer. We are used the first approaches which is a single dense layer with six outputs with a sigmoid activation functions and binary cross entropy loss functions. Each neuron in the output dense layer will represent one of the 6 output labels. The sigmoid activation function will return a value between 0 and 1 for each neuron. If any neuron's output value is greater than 0.5, it is assumed that the news text belongs to the class represented by that particular neuron. (Usman Malik, 2020)

4.9 Visualizing Losses and Accuracy

By changing different parameter values such as epoch size, batch size, optimizer selection, dropout and the layers of the network until the best fit model is found. In this section the results of experiments are presented. Based on experimental analysis. We perform different conv, max pool and FC layer configuration and observe the training and validation accuracy for a given CNN layer configuration to select the best depth of the proposed CNN. We selected the CNN layer configuration with the best performance result for our training and validation dataset

The accuracy of the CNN model with layer other than the proposed are less accurate as experimentally evaluated. Then we improve the performance of the selected CNN architecture by tuning other parameters of the network. The other network parameter we

change it during our experimental analysis was the batch size selection. The other parameter which has an effect on the performance of our proposed model was the optimizer selection. Optimizers are used for weight update of the network and they have their own behaviors. For our proposed model the Adam optimizer gives better results compared with stochastic gradient decent. and Adam optimizer is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data (Jason Brownlee, 2017)
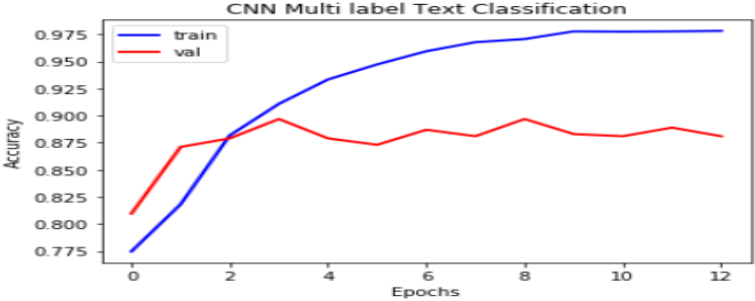
a) Training accuracy
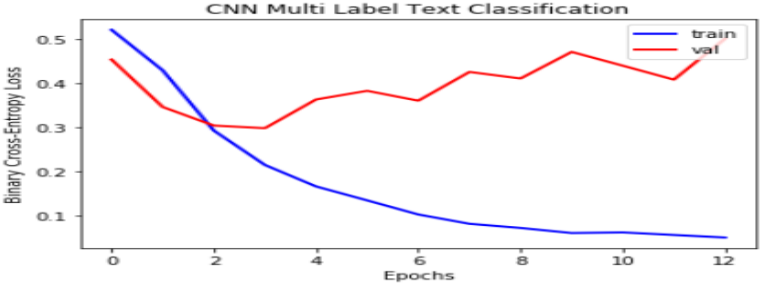
b) Training loss

Figure 4. 9 Training accuracy and loss

As clearly shown below in the training loss and accuracy curve in figure 4.8 training and validation accuracy increases while training and validation loss decreases nearly linearly. There was no sign that an account for indication of overfitting occurrence. This is due to the effect of the activation function (ReLU), Adam optimization algorithm and Early Stopping. Both training and validation accuracy, as well as the training loss and validation loss, goes neck on the neck until the last (14) epoch. The gaps between the training accuracy and validation accuracy as well as the training loss and validation loss is small throughout the curve.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this study, we have proposed CNN models for Multi label Amharic News text Classification using CNN. These embeddings are used individually and in various combinations through different channels of CNN to predict class labels. The CNN models produce optimal features to represent texts by considering their contextual information, which is later used to analyze unlabeled texts. We have compared our proposed models in terms of specified parameter settings against one of the existing models and found that the proposed models perform better in terms of accuracy. The proposed CNN models could be useful to label Amharic News text.

Researchers for developing Amharic classifiers are highly concentrated on improving the classification accuracy as the number of categories increased by using different learning algorithms. They tried to solve the problem of computation time using statically machine learning algorithms using different method. However, all researcher in low resource language are done on multi class, hierarchical and flat classification type whereas we designed a model for multi label text classification and also tried to implement using deep learning algorithm. In order to reduce overfitting, we used most common Adam optimizer, Regularize tuning parameters. The experimental analysis is conducted over 1200 Amharic news under 6 major news categories. We used Python as experimental environment for word embedding as well as text classification purpose. The classification accuracy is improved with compare non pretrained word embeddings in increase label. The challenge in this study is as data size are small compare with other due to no published multi label dataset.

5.2 Contribution of the Study

Some of the contributions of the study are listed below:

- The study shows how to develop a multi label Amharic text classification using CNN deep learning algorithms for Amharic news text classification by preparing pretrained word embedding. These research address multi label text classification and support also multi class text classification to low resources language other researcher done in flat classification and hierarchical text classification.

- One of the major problems in machine learning are data specially in low resource language we prepared some datasets for word embedding as well as multi label Amharic text classification tasks.

5.3 Recommendations

We did many tasks to design model for Multi label Amharic text classification using CNN system for Amharic news text classification. Based on the finding of the study, we the following ways as forward. In other languages, it is common to prepare corpus for every NLP task for the research purpose like '***Reuters-21578***' for English text classification. However, we did not get any corpus prepared for Amharic news text classification. Due to this, we devote much of our time on dataset preparation.

- We recommend to uses ELMo deep contextualized word representation because it helps complex characteristics of word use and uses vary across linguistic context and word vectors learned functions uses deep bidirectional language model character-based representation significantly improvements than word embedding representation.

- We recommend researchers to use other deep learning algorithms such as RNN, HAN because of its advantage RNN is class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Whereas HAN has two levels of attention mechanisms applied at the word and sentence-level, enabling it to attend differentially to more and less important content when constructing the document representation.

- We recommend researchers to use different types of word2vec such as Glove and FastText because of their advantages Glove goal is basically to construct word vectors that capture meaning in vector space and take advantage of global count statistics rather than just local information. Whereas fastText considers not only the word itself but also groups of characters from that word and the sub word information such as character unigram, bigram, trigrams etc. during learning word representations.

REFERENCES

A. Santos, A. C. (2011). A comparative analysis of classification methods to multi-label tasks in different application domains. *International journal of computer Information systems and Industrial Management Application, 3*, 218-227.

Adam, P. J. (2017). Deep Learning: A Practitioner's Approach. *Sebastopol: O'Reilly*.

Addis, A. (2010). Study and Development of Novel Techniques for Hierarchical Text Categorization. *University of Cagliari, Italy.* .

Adrian, R. (2017). Deep Learning for Computer Vision with Python. *PyImageSearch*.

AKLILU, Y. Y. (2019). EXPLORING NEURAL WORD EMBEDDINGS FOR AMHARIC LANGUAGE . *NEAR EAST UNIVERSITY* , 1-20.

Alemu Kumilachew, A. ,. (2017). A Comparative Study of Flat and Hierarchical Classifcation for Amharic News Text Using SVM. *I.J. Information Engineering and Electronic Business*, 36-37.

Antonie, M. a. (2002). Text Document Categorization by Term Association . *Proceedings of the 2002 IEEE International Conference on Data Mining*, 30-50.

Basirat, A. (2018). Principal Word Vectors. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-353866

Beakcheol JangID, I. K. (August 22, 2019). Word2vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE*, 14-15.

Bender, M. B. (1976). Language in Ethiopia. *Oxford University Press , London*.

Bengio, Y. D. (2003). A Neural Probabilistic Language Model. 19.

Berger, H. (2004). A Comparison of Text Categorization Methods Applied to N-Gram Frequency Statistics. *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, 4-10.

Berger, M. J. (2019). *Large Scale Multi-label Text Classification with Semantic Word Vectors.* Retrieved from https://cs224d.stanford.edu/reports/BergerMark.pdf

Blei, D. M. (2003). Latent Dirichlet Allocation. 28.

C. C. Aggarwal, C. Z. (2012). Mining text data. *Springer Science & Business Media*, 1-5.

Collobert, J. W. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* , 12:2493–2537.

Collobert, R. &. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.

Demissie, D. (2017). Amharic Named Entity Recognition Using Neural Word Embedding as a Feature. *Addis Abeba University*.

Duwairi, R. (2007). Arabic Text Categorization. . *International Arab Journal of Information Technology*, 1-7.

Elrazzaz, M. E. (2017). Methodical Evaluation of Arabic Word Embeddings. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.* Retrieved from http://aclweb.org/anthology/P17-2072

Farzindar A, I. D. (2015). Natural Language Processing For Social Media Synthesis Lectures on Human Language Technologies. *http://doi.org/10.2200/S00659ED1V01Y201508HLT030*, 8 ,1-166.

Freitas, d. C. (2009). A tutorial on multi-label classification techinques volume Foundations of Computational Intelligence. *Studies in Computational Intelligence 205, Springer*, 177-179.

G. Ian, Y. B. (2012). Deep Learning.

Gambäck, B. &. (2010). Experiences with developing language processing tools and corpora for Amharic. *ResearchGate*, 2-3. Retrieved from https://www.researchgate.net/publication/229034442

Gambäck, B. O. (2009). Methods for Amharic Part-of-Speech Tagging. *Proceedings of the First Workshop on Language Technologies for African Languages.* Retrieved from http://www.aclweb.org/anthology/W09-0715

Graves, A. M. (2013). Speech Recognition with Deep Recurrent Neural Networks. *ArXiv:1303.5778 [Cs]*. Retrieved from http://arxiv.org/abs/1303.5778

Harris, Z. S. (1954). Distributional Structure WORD. Retrieved from https://doi.org/10.1080/00437956.1954.11659520

Harris, Z. S. (1954). Distributional Structure. WORD. 10(2–3), 146–162. Retrieved from https://doi.org/10.1080/00437956.1954.11659520

J. Fan, Y. F. (2008). High dimensional classification using features annealed independence rules. *Annals of statistics, 36*(6), 2605.

Jason Brownlee. (2017, july 3). *Machine Learning Mastery*. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Jeffrey Pennington, R. S. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2-4.

K. Alex, S. I. (2012). ImageNet Classification with Deep Convolutional Neural Networks," Advances in neural information processing systems. 1097-1105.

Kanj, S. (2017). Learning methods for multi-label classification. *HAL*, 12-13.

Kassie, T. (2009). WORD SENSE DISAMBIGUATION FOR AMHARIC TEXT RETRIEVAL: A CASE STUDY FOR LEGAL DOCUMENTS.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751.* Retrieved from http://www.aclweb.org/anthology/D14-1181

Koehrsen, W. (2018, Oct 15). *Neural Network Embeddings Explained*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 60-65. Retrieved from https://doi.org/10.1007/BF00337288

Koller, S. (1997). Hierarchically classifying documents. *14th national conference on .* Stanford: Stanford University.

Krizhevsky, A. S. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25*, pp.(1097–1105). Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Kumar Shridhar, J. L. (2020). ProbAct: A Probabilistic Activation Function for Deep Neural Networks. *arXiv*, 2.

Mandelbaum, A. &. (2016). Word Embeddings and Their Use In Sentence Classification Tasks. Retrieved from http://arxiv.org/abs/1610.08229

Mark HUGHES, I. L. (2013). Medical Text Classification using Convolutional Neural Networks . *IBM TJ Watson Research Center*, 1-5.

MD.ASLAM PARWEZ, M. A. (2019). Multi-Label Classification of Microblogging Texts Using Convolution Neural Network. *IEEE*, 68685-68689.

Mequanent Argaw. (2019). Amharic Parts-of-Speech Tagger using Neural Word Embeddings as Features. *Addis Abeba University*, 40-41.

Mikolov T, C. K. (2013). Efficient Estimation of Word Representations in Vector Space. Retrieved from http://arxiv.org/abs/1301.3781

Mikolov, S. K. (2013). Distributed Representations of Words and Phrases and their Compositionality. *ArXiv:1310.4546 [Cs, Stat].* Retrieved from http://arxiv.org/abs/1310.4546

Mikolov, T. K. (2010). Recurrent Neural Network Based Language Mode.

Mikolov, T. Y. (2013). Linguistic Regularities in Continuous Space Word Representations. 6.

Mykowiecka, A. M. (2017). Testing word embeddings for Polish.

Nitish Srivastava, G. H. (2014). Dropout: A Simple Way to Prevent Neural Networks from overfitting . *Journal of Machine Learning Research* , 1929-1958.

Nooney, K. (2018, june 8). *Medium*. Retrieved from https://towardsdatascience.com/: https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff

Oludare Isaac Abiodun, A. J. (2017). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE*, 9-10.

Pandey, P. (2020, March 18/2018). *Understanding the Mathematics behind Gradient Descent*. Retrieved from Meduim Data science: https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e

Popa, I. Z. (2007). Text Categorization for Multi-Label Documents and Many Categories Washington DC,USA. *IEEE*.

Pritchard, J. K. (2000). Inference of Population Structure Using Multilocus Genotype Data. 13.

S.Niharika, V. L. (2012). A SURVEY ON TEXT CATEGORIZATION. *International Journal of Computer Trends and Technology*, 39-41.

Salton, G. &. (1988). Term-weighting approaches in automatic text retrieval. . *Information Processing & Management*.

Scherer, D. A., & Behnke. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *In Proceedings of the Artificial Neural Networks–ICANN 2010*, (pp. 92–101). Thessaloniki, Greece,15–18.

SEBASTIANI, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 1-2.

Shweta C. Dharmadhikari, M. I. (n.d.). A Comparative Analysis of Supervised Multi-label Text Classification Methods. *IJERA, 1*(4), 1952-1961.

Singh, M. (2017, Oct 7/31/2020). *Data Science Group*. Retrieved from medium: https://medium.com/@manjeetsingh0

Surafel, T. (2003). Automatic categorization of Amharic news text: A machine learning approach. *Department of Information Science, Addis Ababa University*.

Tedla, Y. &. (2017). Analyzing word embeddings and improving POS tagger of tigrinya.

Tewodros, H. (2003). Amharic text retrieval An Experiment Using Latent Semantic Indexing (LSI) with Singular value Decomposition (SVD). *M Sc Thesis, Addis Abeba University ,Ethiopia*.

Tripodi, R. &. (2017). Analysis of Italian Word Embeddings.

Usman Malik, U. (2020, April 24). *Stuck Abuse*. Retrieved from stackabuse.com: https://stackabuse.com/python-for-nlp-multi-label-text-classification-with-keras/

Worku, K. (2006 E.c). Automatic Amharic Text news Classification :a neural network approach. *Bahir Dar University*.

Yohannes, A. (2007). Automatic Amahric Text Classification. *Addis Abeba University*, 1-5.

Younes. Z, A. F. (2011). A Dependent Multilabel Classification Method Derived from the -Nearest Neighbor Rule. *EURASIP Journal on Advances in Signal Processing 2011*, 22, 50, 73, 83.

Yufeng Liu, H. H. (2011). Hard or Soft Classification? Large-Margin Unified Machines. *Journal of the American Statistical Association*, 106:493, 166-177, DOI: 10.1198/jasa.2011.tm10319.

Zelalem, S. (2001). Automatic Classification of Amharic News Items: The Case of Ethiopian News Agency. . *School of Information Studies for Africa, Addis Abeba University*.

Zhang Y, W. B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *ArXiv:1510.03820 [Cs].* Retrieved from http://arxiv.org/abs/1510.03820

Zhang, X. Z. (2015). Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems 28*, (pp. 649–657). Retrieved from http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf

APPENDIX 1

AMHARIC PUNCTUATION MARKS AND BASIC ETHIOPIC NUMBERS

Arabic Number	Ethiopic Number	Amharic Punctuation Marks and their description
1	፩	. (Period)
2	፪	። (Word space)
3	፫	… (Ellipsis)
4	፬	። (Full stop)
5	፭	፣ or ፥ (Comma)
6	፮	፤ (Semi-colon)
7	፯	፦ (Preface colon)
8	፰	፧ (Question mark)
9	፱	※ (Section mark)