Jimma University
Institute Of Technology

Afaan Oromo Named Entity Recognition Using Deep Learning

Ibsa Beyene Deressa

A THESIS SUBMITTED TO DEPARTMENT OF INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

JIMMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS

By: Ibsa Beyene

This is to certify that the thesis prepared by Ibsa Beyene, entitled Afaan Oromo Named Entity Recognition Using Deep-Learning Approach, and Submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology compiles with the regulations of the University and meets the accepted standards with respect to originality and quality.

Approved by board of Examining Committee:

	Name	Signature
Faculty Dean:	Dr. Worku Jima	
Advisor:	Dr. Teklu Urgessa	[illegible]
External Examiner:	Dr. Kula Kekeba	[illegible]
Internal Examiner:	Mr. Tafari kebebew (Msc)	
Chair Person:	Mr. Admas Abtew (Msc)	

Jimma, Ethiopia

January, 2020

DEDICATION

I dedicate this work to my father **Beyene Deressa Bidu** who passed away while suffering a lot to bring a bright future for his children and to my mother **Degitu Eticha Gonfa** who is the sign of true love. As long as I live, you'll be remembered from every dawn to dusk.

ACKNOWLEDGMENT

First and foremost, praises and thanks to **God**, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, **Dr. Teklu Urgessa(Ph.D.)**, for allowing me the opportunity to do research and providing invaluable guidance throughout this research.

I am extremely grateful to my parents and friends for their love, prayers, caring, and sacrifices for educating and preparing me for my future. My Special thanks go to my brother Evangelical preacher **Abebe Beyene Deressa**; I can say all my success is the result of your effort. You are not merely a brother you act as my father. Thank you Abeko.

Abstract

Named Entity Recognition (NER) is an essential and challenging task in Natural Language Processing (NLP), particularly for resource-limited languages like Afaan Oromo(AO). NER is a type of sequence tagging task that assigns a label to each entity that contains multiple tokens. In this research, we propose a variety of Long Short-Term Memory (LSTM) based models for sequence tagging. These models include LSTM networks, bidirectional LSTM networks, LSTM with a Conditional Random Field layer (LSTM-CRF, and bidirectional LSTM with a CRF layer (BiLSTM-CRF). We show that the BiLSTM-CRF model can efficiently use both past and future input features by using BiLSTM. The proposed approach aims at automating manual feature design and avoiding dependency on other natural language processing tasks for classification features. In this paper potential feature information represented as a word with their index is generated using the neural network from text files. These generated features are used as features for Afaan Oromo Named entity classification. Afaan Oromo NE corpus have been developed based on CoNLL's 2002, BIO tagging scheme. Four NE categories have been identified and used in this research work: person, location, organization, and miscellaneous. The miscellaneous category includes date/time, monetary value, and percentage.

We have used the AONER corpus of around 700 Afaan Oromo sentences, and from this corpus, we have used 567 sentences for training, 67 sentences for validation and, 70 sentences for testing of our work. We got an accuracy of 74.14 % using bidirectional LSTM and CRF layer (BiLSTM-CRF) for AONER model.

Keyword: Named Entity Recognition, Afaan Oromo NE corpus, BI-LSTM-CRF

LIST OF FIGURES

LIST OF TABLES

List of Acronyms & Abbreviations

CRF	Conditional Random Fields
HMM	Hidden Markov Model
IE	Information Extraction
IR	Information Retrieval
MUC	Message Understanding Conference
NEs	Named Entities
NER	Named Entity Recognition
NERC	Named Entity Recognition and Classification
GATE	General Architecture for Text Engineering
ML	Machine Learning
CoNLL	Conferences on Natural Language Learning
AONER	Afaan Oromoo Named Entity Recognition
NL	Natural Language
NLP	Natural Language Processing
POS	Part Of Speech
POST	POSTagger
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network

CONTENTS

1. CHAPTER ONE

1.1 Introduction

A language can be defined as a set of rules or a set of symbols. Symbols are combined and used for conveying information or broadcasting the information [1]. In its written form it serves as a means of recording information and knowledge on a long term-basis and transmitting what it records from one generation to the next, while in its spoken form it serves as a means of coordinating our day-to-day life with other[2].

Natural Language Processing (NLP) is a tract of Artificial Intelligence and Linguistics, devoted to making computers understand the statements or words written in human languages. Natural language processing came into existence to ease the user's work and to satisfy the wish to communicate with the computer in natural language. Since all the users may not be well-versed in a machine-specific language, NLP provides those users who do not have enough time to learn new languages or get perfection in it [1]. The goal of NLP is to be able to design algorithms to allow computers to "understand" natural language to perform some tasks such as Spell Checking, Keyword Search, Finding Synonym, Parsing information from websites, documents, etc [3].

The most explanatory method for presenting what happens within a Natural Language Processing system is in terms of the 'levels of language' approach. Some of the levels of NLP are Phonology (this level deals with the interpretation of speech sounds within and across words), Morphology (deals with the componential nature of words, which are composed of morphemes – the smallest units of meaning); Syntactic (focuses on analyzing the words in a sentence so that uncover the grammatical structure of the sentence). This requires both a grammar and a parser. The output of this level of processing is a representation of the sentence that reveals the structural dependency relationships between the words; Semantic (determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence), Discourse: examines the structure of different kinds of text and uses document structure to extract additional meaning (While syntax and semantics work with sentence-length units, the discourse level of NLP works with units of text longer than a sentence), Pragmatic: The practical or pragmatic level depends on a body of knowledge about the world that comes from outside the contents of the document (is concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding), and Lexical, (At this level, humans, as well as NLP systems, interpret the meaning of individual words) [4].

Our study focuses on the lexical level of language to detect and classify the named entity recognition for Afaan Oromo text documents. In addition to the complicated task of natural language processing, Afaan Oromo also lacks the structured text documents to be easily processed by computer.

The Named Entity Recognition (NER) task aims to identify and categorize proper nouns and important nouns in a text into a set of predefined categories of interest such as persons, organizations, locations, etc [6]. NER systems are often used as the first step in question answering, information retrieval, co-reference resolution, topic modeling, etc. In the taxonomy of computational linguistics tasks, it falls under the domain of ”Information Extraction”, which extracts specific kinds of information from documents as opposed to the more general task of ”document management” which seeks to extract all of the information found in a document [7]. NER performs what is known as surface parsing, delimiting sequences of tokens that answer the questions like ”who”, ”where” and ”how much” in a sentence [7], [8].

The term ”Named Entity”, now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6) [9]. At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense-related activities is extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization, and location names, and numeric expressions including time, date, money, and percent expressions.

Identifying references to these entities in a given text was recognized as one of the important sub-tasks of IE and was called "Named Entity Recognition and Classification (NERC)".

Approaches to the development of named entity recognition can be broadly classified into three. These are rule-based (linguistic approach), machine learning (statistical approach), and hybrid approach [10]. Rule-based approaches are based on handcrafted rules by language experts. Machine learning approaches use a collection of training examples to build a model from training data and a hybrid approach uses a combination of the two. Machine learning approaches are further divided into supervised, unsupervised, and semi-supervised. A Supervised approach uses labeled training data for building NER models. Unsupervised approaches do not need any structural information about the data[11]. Semi-supervised approaches use both labeled and unlabeled data for training. This study focuses on the supervised approach.

1.2 Statement Of The Problem

"Currently, the world is all about information, most of it is in digital format. The World Wide Web contains billions of documents and it is growing at an exponential pace [5]." The amount of data we produce every day is truly mind-boggling. According to Forbe's report, there are 2.5 quintillion bytes of data created each day at our current pace, but that pace is only accelerating with the growth of the Internet of Things (IoT). Over the last two years alone 90 percent of the data in the world was generated. This is worth re-reading! While it's almost impossible to wrap your mind around these numbers, we gathered together some of our favorite stats to help illustrate some of the ways we create these massive amounts of data every single day.

With so much information at our fingertips, we're adding to the data stockpile every time we turn to our search engines for answers. We conduct more than half of our web searches from a mobile phone now. More than 3.7 billion humans use the internet (that's a growth rate of 7.5 percent over 2016). On average, Google now processes more than 40,000 searches EVERY second (3.5 billion searches per day)! While 77% of searches are conducted on Google, it would be remiss not to remember other search engines are also contributing to our daily data generation. Worldwide there are 5 billion searches a day.

The contributing factors are the availability and accessibility of rapid growth of broadcast systems, Internet, and online information services [5].

Obviously, this explosion of information has caused a well-recognized information overload problem. For this matter, it is obvious that there is no time to read everything to make critical decisions based on available information.

Therefore, for this problem, there must be tools that provide timely access to, and digest of, various sources are necessary in order to alleviate the information overload people are facing.

NER is a technology for recognizing proper nouns (entities) in text and associating them with the appropriate types. In information extraction systems, NEs generally carry important information about the text itself and thus are targets for extraction. Since entity names form the main content of a document, NER is a very important step toward more intelligent information extraction and management [7].

The task has also very great significance in the Internet search engines and many of the language engineering applications such as Machine Translation, Question-Answering systems, Indexing for Information Retrieval, and Automatic Text Summarization [12]. These facts show that the NER system is the basic and relevant component for the development of most NLP related applications.

One of the main challenges in NER systems is the ambiguity of Named entities in text. The ambiguities lay at two levels: detection and classification. The first involves detecting named entities from non-named entities and the other challenge is classifying the entities into predefined categories such as a person, organization, location, date, address, and time expressions. Afaan Oromo Named entity recognition AONER has many challenges. Among these challenges ambiguity and nested named entities are common ones.

Afaan Oromo is one of the most widely used languages in Ethiopia, and also spoken in Kenya, Somalia, and Djibouti. Nowadays, textual information is highly increasing from time to time both in softcopy and hardcopy format. The textual information disseminated to the populations located at various countries is provided by various sources, such as; Internet, media, presses, books, journal articles, newspaper, etc.

Even though Afaan Oromo is among the major languages in Africa with a lot of vocabulary, it lacks enough research work on NER. As far as the researchers see there were only two research works on Afaan Oromo Named Entity Recognition (AONER) [12] and [13] the first was done by the statistical approach using Conditional Random Fields (CRF) algorithm. This approach has a shortcoming in classifying complex entities. The hybrid system developed by [13] depends on the knowledge of linguists to enhance the size of gazetteers. Some NER systems are incorporated into Parts-of-Speech (POS) taggers only, whereas they also often make use of lists of typed entities (like list of possible person names), which is impossible to list all entity types manually, or regular expressions for particular types (like address patterns). But our study is based on analyzing patterns of POS tags to forecast the predefined named entity types. This study intended at simplifying the process of manual feature design by using automatically generated features for classification. Therefore, this study tries to answer the following research questions:

1. RQ1: To what extent, additional features incorporated in this study, affect the performance and quality of the recognizer?
2. RQ2: Can we optimize the process of feature extraction for Afaan oromo NER using deep neural network?

1.3 Motivations For Conducting This Research

In recent years, deep learning (DL, also named deep neural network) has attracted significant attention due to their success in various domains. Starting with, DL-based NER systems with minimal feature engineering have been flourishing. Over the past few years, a considerable number of studies have applied deep learning to NER and successively advanced the state-of-the-art performance [14], [15]. On the other hand, although a NER study has been thriving for a few decades, to the best of our knowledge, there is no research conducted on DL-based AONER so far. Besides these, this study is inspired by the contribution of deep learning-based AONER tasks to other NLP studies.

1.4 Objectives

General Objective

The general objective of this research is to develop AONER by using the embedding layer that automatically generates word vector features.

Specific Objectives

To achieve the general objective the following specific objectives are set:

- To study the structure of Afaan Oromo language and the linguistic patterns that refers to NEs of the language.
- To Review literature and related work on areas of Natural language processing and Information Extraction.
- To identify an appropriate algorithm for Afaan Oromo Named Entity Recognition.
- Building deep neural network classifiers that use word vector features as input.
- To test and evaluate the performance of the developed model with different parameters
- To state conclusion and recommendation based on the experimental results.

1.5 Methodology

A number of methods are employed for the effective completion of this study. Some of the methods are:

1.5.1 Literature review

A literature survey of theoretical bases on the existing Afaan Oromo Named Entity Recognition Systems and other language papers have been reviewed. In addition to research papers: journal articles and conference papers are also reviewed.

1.5.2 Corpus Collection and Development

Afaan Oromo doesn't have a publicly annotated corpus for NER and other NLP tasks. The researcher collected data for Afaan Oromo text from different sources including newspapers, research papers, and books. The collected data are labeled into POST (which is used as basement for other features to be generated by the system) and Named entity tag classes for further experiments.

1.5.3 Tool Selection

For the development of Afaan oromoo NER we selected python programming language as backbone. Python is a programming tool that runs on the anaconda platform for python programming and we have used Jupyter notebook as text editor. some of the libraries used in this study are:

Pandas is a Python data analysis library and is used primarily for data manipulation and analysis. It comes into play before the dataset is prepared for training. Pandas make working with time series and structured multidimensional data effortless for machine-learning programmers.

TensorFlow is one of the best library available for working with Machine Learning on Python. Tensorflow makes ML model building easy for beginners and professionals alike.

Keras is one of the most popular and open-source neural network libraries for Python. Initially designed by a Google engineer, Keras was soon supported in TensorFlow's core library making it accessible on top of TensorFlow. Keras features several of the building blocks and tools necessary for creating a neural network such as: Neural layers, Activation and cost functions, Objectives, Batch normalization, and Dropout.

Scikit-learn is another actively used machine learning library for Python. It includes easy integration with different ML programming libraries like Numpy and Pandas. Scikit-learn comes with the support of Classification.

Numpy The Numpy library for Python concentrates on handling extensive multi-dimensional data and the intricate mathematical functions operating on the data. Numpy offers speedy computation and execution of complicated functions working on arrays.

1.5.4 Model Development

Using automatically generated features different classifiers are used to build AONER models.We used BiLSTM-CRF model to train and predict the model. This model can use both past and future input features thanks to a bidirectional LSTM component. Also, this model can use sentence-level tag information due to a CRF layer. ReLu(Linear activation function) which is computationally efficient means it allows the network to converge quickly and non linear that has a derivative function and allows for backpropagation as an activation function and RMSprop which is good, fast and very popular optimizer.

1.5.5 Model Evaluation

The performance of models developed is measured by using accuracy which is the ratio of the number of correctly classified instances to the total number of instances.

1.6 Scope

The goal of this research is to develop Afaan Oromo named entity recognition system. The three types of entities identified by Message Understanding Conferences (MUC) are ENAMEX, TIMEX, and NUMEX.

1. **ENAMEX** contains,*Person name* - named person or family. Eg. Ibsa Beyene etc.
   Location name - name of politically or geographically defined location. Eg, Jimma, Ethiopia etc.
   Organization name - named corporate governmental or other organizational entity. Eg. Jimma University, Non-Governmental Organization (NGO) etc.

2. **TIMEX** contains, *DATE* includes complete or partial date expression. Wiixata Ganama / Monday Morning/, Fulbaana 10 / September 10 / etc. *TIME* includes complete or partial expression of time of day Borganama / Tomorrow Morning

3. **NUMEX** contains- *numeric expressions*, monetary expressions and percentages; expressed in either numeric or alphabetic form.
   Money- monetary expression eg. 2 birr, $3 etc.
   Percent – percentage expression eg. 10%, 100% etc.

From the above MUC conference entity types, the scope of our study is limited to recognition and classification of named entities of person, organization, location (ENAMEX) and Miscellaneous (MISC) which categorizes TIMEX and NUMEX under one category.

1.7 *Application Of Results*

NEs are recognized as an important source of information for many applications in NLP. NER is a technology for recognizing proper nouns in text and associating them with the appropriate types. This is the problem of many researchers [12; 16]. The main challenge of this problem is to find the suitable NER systems and try to find the difference of their NE types. NER software serves as an important preprocessing tool for tasks such as information extraction, information retrieval, and other text processing applications. Based on these facts, the Afaan Oromo NER system plays a crucial role in information extraction based researches and applications associated with the language. Then it opens the track for future Afaan Oromo NLP studies. It also simplifies the development of the following applications for Afaan Oromo.

- Search Engines (Semantic based)
- Relation extraction
- Question-Answering Systems
- Indexing for Information Retrieval
- Automatic text Summarization

1.8 *Thesis Organization*

The rest of this thesis is organized as follows. Theoretical backgrounds in NER are explained in Chapter Two. A review of the literature, an overview of information extraction, Natural Language Processing, and feature spaces for NER systems are also discussed in this chapter. Chapter 3 discusses related works that have been done. An overview and language-specific topics regarding Afaan Oromo and related works that have been done so far for different languages in the area of NER task are discussed in this chapter. Then chapter 4 presents the design and implementation of the AONER system using the BiLSTM-CRF approach. The Chapter discusses the architecture of our proposed architecture and the phases of AONER. Chapter 5 presents the corpus development and the experimental results of the proposed system along with its discussion. Finally, Chapter 6 concludes our study with the recommendations and future works.

2. CHAPTER TWO

2.1 Literature Review

This chapter covers concepts that are vital to the understanding of Named Entity Recognition (NER). It starts with a brief introduction to natural language processing (NLP) and continues discussing Information Extraction (IE), IE components among which is NER is one. Then the chapter goes on discussing NER, which includes its architecture, the approaches used, feature sets, and evaluation metrics.

2.2 Natural language processing (NLP)

The amount of electronic data available on the web is increasing today. Even though huge data is available, the complexity of natural languages makes it difficult to access the text information. Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the desired tasks [17]. This is based on both a set of theories and a set of technologies. In recent years, the natural language text interpretation and processing technologies have also gained an increasing level of sophistication. NLP technologies are becoming extremely important in the creation of user-friendly decision-support systems for everyday non-expert users, particularly in the areas of knowledge acquisition, information retrieval, and language translation [16].

NLP is widely used in modern computer systems. It is concerned with the interactions between computers and human languages. As such, NLP is related to the area of human-computer interaction. Many challenges in NLP involves natural(human) language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation. Currently, the field of NLP includes a wide variety of linguistic theories, cognitive models, and engineering approaches. Computer science is one of these fields, which provides techniques for model representation, and algorithm design and implementation. These models classified words not only based on meaning, but also on their co-occurrence with other words.

NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications [4]. An NLP system must possess considerable knowledge about the structure of the language itself, including what the words are, how to combine the words into sentences, what the words mean, how these word meanings contribute to the sentence meaning, and so on. The system would need methods of encoding and using this knowledge in ways that will produce the appropriate behavior. Furthermore, the knowledge of the current situation (or context) plays a crucial role in determining how the system interprets a particular sentence [18].

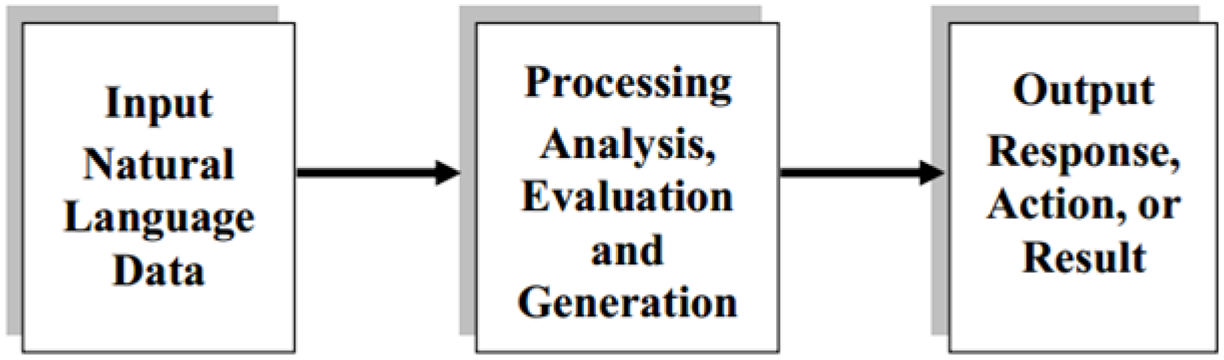

Input
Natural Language Data
Processing
Analysis, Evaluation and Generation
Output
Response, Action, or Result

Fig. 2.1: Information Flow in Nlp

The goal of NLP is "to accomplish human-like language processing". There are more practical goals for NLP, many related to the particular application for which it is being utilized. For example, an NLP-based IR system has the goal of providing more precise, complete information in response to a user's real information need. The goal of the NLP system here is to represent the true meaning and intent of the user's query, which can be expressed as naturally in everyday language as if they were speaking to a reference librarian. Also, the contents of the documents that are being searched will be represented at all their levels of meaning so that a true match between need and response can be found, no matter how either are expressed in their surface form [4].

2.3 Information Extraction (IE)

A large amount of information is being generated rapidly on the web every day and social networking applications are one of the main reasons for the increase of web data. All these data are usually unstructured in nature. With the increasing volume of publicly available information, companies need to develop processes for mining information that may be vital for their business. Unfortunately, much of this information is presented in the form of unstructured or semi-structured texts. Software tools are not able to analyze such texts and humans would take so much time to perform this task that the information would become obsolete by the time it was available. Information Extraction emerged as a solution to deal with this problem [19].

Currently, there is considerable interest in using these technologies for information retrieval, since there is an increasing need to localize precise information in documents, for instance, as the answer to a question, rather than retrieving the entire document or a list of documents. Information Extraction is the task of automatically extracting structured information like entities and the relations among entities from unstructured data and/or semi-structured documents [20].

Information extraction should not be confused with the more mature Technology of information retrieval (IR), which given a user query selects a (hopefully) relevant subset of documents from a larger set. The user then browses the selected documents in order to fulfill his or her information need. Depending on the IR system, the user may be further assisted by the selected documents being relevance ranked or having search terms highlighted in the text to facilitate identifying passages of particular interest. The contrast between the aims of IE and IR systems can be summed up as: IR retrieves relevant documents from collections; IE extracts relevant information from documents. IE concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text. The process takes texts as input and produces fixed-format, unambiguous data as output. This data may be used directly for a display to users, or may be stored in a database or spreadsheet for later analysis, or may be used for indexing purposes in IR applications such as Internet search engines like Google.

The two techniques are therefore complementary, and their use in combination has the potential to create powerful new tools in text processing [21]. Due to the complexity of information extraction, it is very important to decompose it into a number of tasks. Based on Message Understanding Conference-7(MUC-7), five tasks have been defined within general tasks of IE: NER (Finds and classifies names, places, etc.), Co-reference Resolution (CR)(Identifies relations between entities), Template Element Construction (TEC)(Determining the different attributes of a given entity), Template Relation Construction (TRC)(Finds relations between TE entities), Scenario Template Production (STP)(Fits TEC and TRC results into specified event scenarios [22].

Thus, the NER system is one of the most important preprocessing tasks for the other sub-tasks of information extraction.

To put it pictorially

• *Information Retrieval* gets sets of relevant documents. You analyze the documents.

• *Information extraction* gets facts out of documents. You analyze the facts.

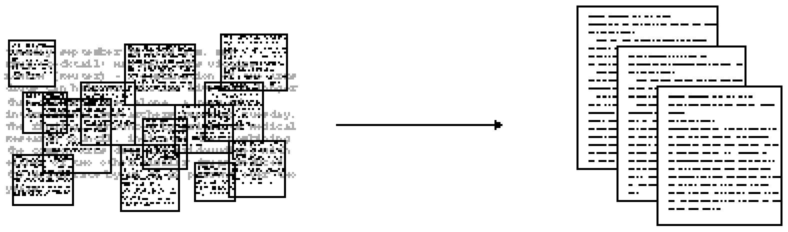


Fig. 2.2: Information retrieval

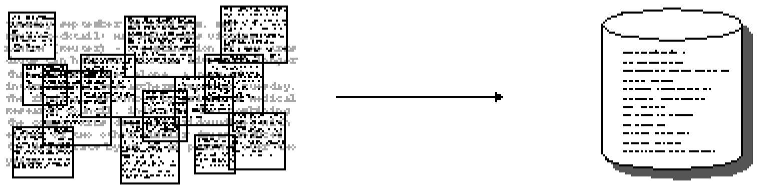


Fig. 2.3: Information extraction

2.3.1 Components of an Information Extraction System

A typical Information Extraction system has phases for input tokenization, lexical and morphological processing, some basic syntactic analysis, and identifying the information being sought in the particular application[23]. Depending on what one is interested in some phases may not be necessary. In addition to the modules in the left-hand column, IE systems may include modules from the right-hand column, depending on the particular requirements of the application [24].

Tokenization

The Tokenization module is responsible for splitting the input text into sentences and tokens. Tokenization is a trivial problem for European languages with clear word borders. However, in processing some languages like Chinese or Japanese, it is not evident from the orthography where the word boundaries are. Therefore, systems for these languages must necessarily be expanded by a Word Segmentation module.

Lexical and Morphological Processing

Morphology is a sub-discipline of linguistics and is interested in the structure of word forms. Many IE systems for languages with simple inflectional morphology, like English, do not have a morphological analysis component at all. In English, it is easy to simply list all inflectional variants of a word explicitly in the lexicon. For languages like French, with more complex inflectional morphology, a morphological analysis component makes more sense, but for a language like German, where compound nouns are agglutinated into a single word, morphological analysis is essential.

In the lexical analysis, the tokens determined by the Tokenization module are looked up in the dictionary to determine their possible parts-of-speech and other lexical features that are required for subsequent processing. The most important job of this module is the handling of proper names. Recognizing names can thereby be done with either handcrafted rules under the Knowledge Engineering approach or automatically trained rules derived from an annotated corpus.

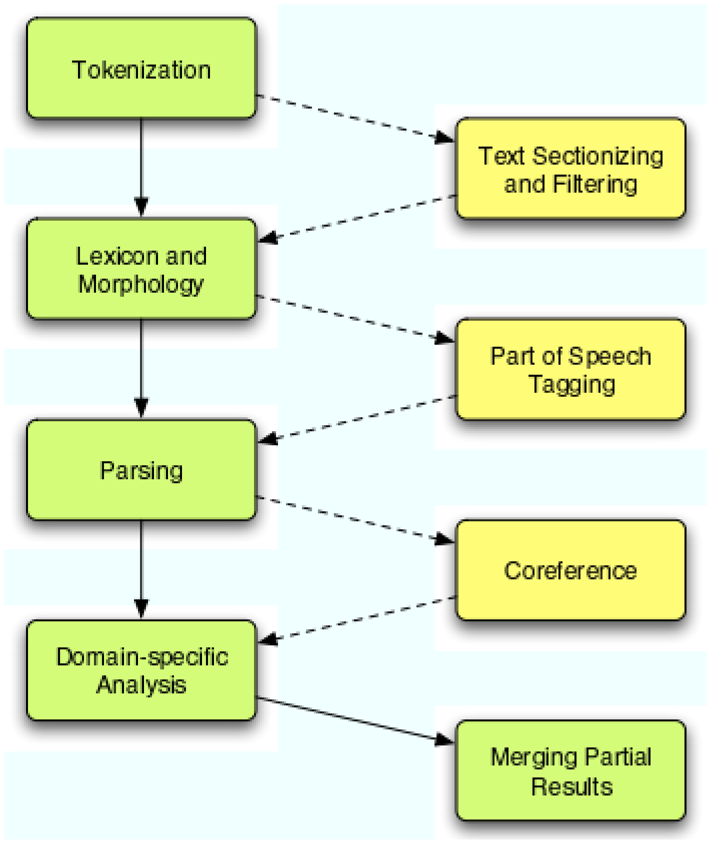


Fig. 2.4: Modules of an Information Extraction System

In addition to name recognition, this module must assign lexical features to words required for subsequent processing. This can be accomplished by either a lookup in a lexicon or by automatic taggers, like a parts-of-speech tagger. A parts of speech tagger annotates each word of a sentence with its parts of speech tag, such as noun, verb, adjective, and so on. It can avoid incorrect analysis based on disambiguation caused by rare word senses in comprehensive lexicons.

Parsing

Syntactic Analysis has the aim to identify the syntactic structure of the analyzed document. Most IE systems only accomplish a shallow, fragment analysis of the texts. But there have been even IE systems which totally skip the phase of syntactic analysis. IE systems are only interested in specific types of information in a text and ignore portions of text, which are not relevant for their task. Therefore, parsing these portions and finding irrelevant grammatical relationships will be unnecessary.

Coreference

The reason for this module is simply that application relevant entities will be referred to in many different ways throughout a given text and thus, success on the IE task was, to a least some extent, conditional on success at determining when one noun phrase referred to the very same entity as another noun phrase. Thus, a Coreference module should handle the following coreference problems:

1. Name-alias coreference. Names and their common variants must be recognized as coreferring, e.g., 'Silvia Miksch' and 'Prof. Miksch'.

2. Pronoun-antecedent coreference. Pronouns like 'she', 'he', 'they', and so on must be associated with their antecedents, resolving them to a domain relevant named entity if possible.

3. Definite description coreference. Handling this problem requires an arbitrary world knowledge to make the connection among descriptors. When building an IE system, it is reasonable to include ontological information for domain-relevant entities that enable such resolution in restricted cases, but doing it, in general, is unrealistic. Examples are

'Apple Computer' and 'the company' or 'Apple' and 'the Cupertino computer manufacturer'.

As with other IE modules, it is possible to build Coreference modules by both the knowledge engineering and automatic learning approach. The knowledge engineering approach is, for instance, applied by FASTUS [24]. For accomplishing the coreference task with the learning approach Decision Trees can be applied.

Domain-Specific Analysis

The domain analysis is the core of most IE systems. The preceding modules prepare the text for the domain analysis by adding semantic and syntactic features to it.

This module fills the templates, which are in general constructed as attribute-value pairs. Templates consist of a collection of slots (i.e., attributes), each of which may be filled by one or more values. These values can consist of the original text, one or more of a finite set of predefined alternatives, or pointers to other template objects. Typically, slot fillers representing dates, times, job titles, and so on are standardized by normalization rules.

For extracting facts and events, the system needs domain-specific extraction patterns (i.e., extraction rules or case frames). These patterns can be generated manually (by means of Knowledge Engineering) or automatically (by means of automatic learning). The portion of text that matches a defined linguistic pattern is memorized and the information is extracted by the guidance of the extraction rule from this portion of the text to fill the template.

For designing the domain-relevant pattern rules there exist two approaches, which can be characterized as (1) the atomic approach and (2) the molecular approach. The molecular approach is the most common one. It involves matching all or most of the arguments in an event in a single pattern. The development cycle of this approach starts with a small number of highly reliable rules that capture the common core cases of the domain, but ignoring a broad class of less frequently occurring relevant patterns. Further development is characterized by capturing ever-larger numbers of increasingly rare and problematic cases with increasingly general and possibly over generating rules. Thus, the system starts with high precision and low recall scores and evolves in increasing recall and progressively lower precision.

The atomic approach builds a domain module that recognizes the arguments to an event and combines them into template structures strictly on the basis of intelligent guesses rather than syntactic relationships. The development cycle can be characterized by assuming domain-relevant events for any recognized entities, leading to high recall, but much overgeneration, and thus low precision. Further development would result in improving filters and heuristics for combining the atomic elements, improving precision.

The atomic approach makes sense at tasks characterized by the following features: (1) entities in the domain have easily determined types and (2) the templates are structured so that there is only one or a very small number of possible slots that an entity of a given type can fill and only entities of a given type can fill those slots.

2.3.2 Merging Partial Results

usually, the sought-after information is spread among different sentences. In these cases information should be combined before creating the final templates. For this purpose some IE systems use a Merging module. This module uses an algorithm to decide which templates can be merged. Again, there are two merging strategies:

1. The Knowledge Engineering approach, where one performs data analyses to identify good merging principles, defines rules, and tests the results; and
2. Using some training mechanism to acquire the merging strategies automatically by the system.

2.3.3 Information Extraction Tasks

The MUC has laid out tasks that are considered as generic tasks that allow the evaluation of different IE systems so that those systems are evaluated based on the performance of the tasks set. The tasks also allow for consistent and reliable evaluation of IE systems. Some of the descriptions of those tasks based on [25], [9] are:-

1. Named Entity Recognition: The first step in most IE systems is the detection and classification of named entities, which are proper nouns, in a natural text. Depending on the domain in which the application of NER is used named entities can be referred to many things for instance in most generic news-oriented NER systems named entity types refer to places, persons, and organizations.

2. Relation Detection and Classification: It is about detecting and classifying the relations that exist between those entity types.

3. is finding and analyzing the events in which entities take part and how those events relate to time are important for the comprehensive extraction of information from a given text.

4.The task of template filling is to find documents that invoke particular scripts, which consists of prototypical sequences of sub-events, participants, roles, and properties, and followed by filling the slots in the associated templates with fillers extracted directly from the text. The fillers may consist of named entities or text segments extracted from a text.

2.4 Named Entity Recognition (NER)

Named entity recognition is the task of identifying named entities like a person, location, organization, drug, time, clinical procedure, biological protein, etc. in text. NER systems are often used as the first step in question answering, information retrieval, co-reference resolution, topic modeling, etc. The first NER task was organized by [9] in the Sixth Message Understanding Conference. The sixth message understanding conference defined named entity labels into three categories [26]. The categories were defined as follows:

- Named entities (ENAMEX): person, organization, location
- Temporal Expressions (TIMEX): date, time
- Number Expressions (NUMEX): money, percentage, quantity

As stated by [12] the core goal of NER is to locate the entities (things) in natural language text and specifies their type. It involves finding Named Entities, such as names of organizations, persons, locations, time expressions, and numeric expressions such as money and percentage expressions from structured and unstructured documents. Putting it in another way, describes NER as an automatic information extraction concerning a particular person, location, organization, title of movie, or book. While doing so, it provides answers to some of the "wh", questions like "who", "where", "when" and "how much" in a sentence. Generally speaking, NER performs what is known as surface parsing, delimiting sequences of tokens that answer these important questions [7].

2.5 Feature Space For NER

Features are descriptors or characteristic attributes of words designed for algorithmic consumption. We organize them along three different axes: Word-level features, List lookup features, and Document and corpus features [27].

Tab. 2.1: Word-level features

Features	Examples
Case	- Starts with a capital letter - Word is all uppercased - The word is mixed case (e.g., ProSys, eBay)
Punctuation	- Ends with period, has internal period (e.g., St., I.B.M.) - Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	- Digit pattern - Cardinal and Ordinal - Roman number - Word with digits (e.g., W3C, 3M)
Character	- Possessive mark, first person pronoun - Greek letters
Morphology	- Prefix, suffix, singular version, stem - Common ending
Part-of-speech	- proper name, verb, noun, foreign word
Function	- Alpha, non-alpha, n-gram - lowercase, uppercase version

2.5.1 Word-level Features

Word-level features are related to the character makeup of words. They specifically describe word case, punctuation, numerical value, and special characters. Table 2.1 lists subcategories of word-level features.

Tab. 2.2: List lookup features

Features	Examples
General list	- General dictionary - Stop words (function words) - Capitalized nouns (e.g., January, Monday) - Common abbreviations
List of entities	- Organization, government, airline, educational - First name, last name, celebrity - Astral body, continent, country, state, city
List of entity cues	- Typical words in organization - Person title, name prefix, post-nominal letters - Location typical word, cardinal point

2.5.2 List lookup features

Lists are the privileged features in NERC. The terms ”gazetteer”, ”lexicon” and ”dictionary” are often used interchangeably with the term ”list”. List inclusion is a way to express the relation ”is a” (e.g., Jimma is a city). It may appear obvious that if a word (Jimma) is an element of a list of cities, then the probability of this word to be a city, in a given text, is high. However, because of word polysemy, the probability is almost never 1 (e.g., the probability of ”Fast” to represent a company is low because of the common adjective ”fast” that is much more frequent).

Tab. 2.3: Document and corpus features

Features	Examples
Multiple occurrences	- Other entities in the context - Uppercased and lowercased occurrences - Anaphora, co-reference
Local syntax	- Enumeration, apposition - Position in sentence, in paragraph, and in document
Meta information	- Uri, Email header, XML section - Bulleted/numbered lists, tables, figures
Corpus frequency	- Word and phrase frequency - co-occurrences - multi-word unit permanency

2.5.3 Document and Corpus Features

Document features are defined over both document content and document structure. Large collections of documents (corpora) are also excellent sources of features. We list in this section features that go beyond the single word and multi-word expression and include meta-information about documents and corpus statistics.

2.5.4 Digit Pattern

Digits can express a wide range of useful information such as dates, percentages, intervals, identifiers, etc. A certain pattern of digits gives a strong signal about the type of named entities. For example, two digits and four-digit numbers can stand for years and when followed by an ”s”, they can stand for a decade. Digits followed by units stand for a quantity such as 10Kg.

2.5.5 Common word Ending

Morphological features are essentially related to the word affixes and root words. Named entities also have common suffixes features. For example, a system should learn that a human profession often ends in ”ist” like in cyclists, journalists, etc.

2.6 Named Entity Recognition Approaches

Named Entity Recognition (NER) is a sub-problem of information extraction and involves processing structured and unstructured documents and identifying expressions that refer to peoples, places, organizations, and companies. NER is a fundamental task and it is the core of the natural language processing (NLP) system. Nowadays more researchers use different methods such as Rule-based NER, Machine Learning-based NER, and Hybrid NER, to identify names from text [28].

2.6.1 Rule-based (Hand-made) Approaches

Hand-made Rule-based focuses on extracting names using lots of human-made rules set. Generally, the systems consist of a set of patterns using grammatical (e.g. part of speech), syntactic (e.g. word precedence), and orthographic features (e.g. capitalization) in combination with dictionaries. This approach relies on manually coded rules and compiled corpora. These kinds of models have better results for restricted domains, are capable of detecting complex entities that learning models have difficulty with. However, the rule-based NE systems lack the ability of portability and robustness, and furthermore, the high cost of the rule maintains increases even though the data is slightly changed. These types of approaches are often domain and language-specific and do not necessarily adapt well to new domains and languages.

2.6.2 Machine Learning-based NER Approaches

In Machine Learning-based NER system, the purpose of the Named Entity Recognition approach is converting an identification problem into a classification problem and employs a classification statistical model to solve it. In this type of approach, the systems look for patterns and relationships into the text to make a model using statistical models and machine learning algorithms. The systems identify and classify nouns into particular classes such as persons, locations, times, etc. based on this model, using machine learning algorithms. There are three types of machine learning models that are used for NER. Supervised, semi-supervised, and unsupervised machine learning model.

Supervised learning involves using a program that can learn to classify a given set of labeled examples that are made up of the same number of features. Each example is thus represented with respect to the different feature spaces. The learning process is called supervised because the people who marked up the training examples are teaching the program the right distinctions. The supervised learning approach requires preparing labeled training data to construct a statistical model, but it cannot achieve good performance without a large amount of training data, because of the data sparseness problem.

The term ”semi-supervised” (or ”weakly supervised”) is relatively recent. The main technique for SSL is called “bootstrapping” and involves a small degree of supervision, such as a set of seeds, for starting the learning process. For example, a system aimed at ”disease names” might ask the user to provide a small number of example names. Then, the system searches for sentences that contain these names and tries to identify some contextual clues common to the five examples. Then, the system tries to find other instances of disease names appearing in similar contexts. The learning process is then reapplied to the newly found examples, so as to discover new relevant contexts. By repeating this process, a large number of disease names and a large number of contexts will eventually be gathered [27].

The unsupervised learning method is another type of machine learning model, where an unsupervised model learns without any feedback. In unsupervised learning, the goal of the program is to build representations from data. These representations can then be used for data compression, classifying, decision making, and other purposes. Unsupervised learning is not a very popular approach for NER and the systems that do use unsupervised learning are usually not completely unsupervised. Unsupervised learning is also required when the classes are not known beforehand, when training examples are not available or the cost to produce training examples is too high, or when objects and features change dynamically.

2.6.3 Hybrid-Based NER Approaches

In a Hybrid NER system, the approach is to combine rule-based and machine learning-based methods and make new methods using the strongest points from each method. Although this type of approach can get a better result than some other approaches, the weakness of Rule-based NER remains the same that is when there is a need to change the domain of data.

2.7 Deep Learning (DL)

In recent years, DL-based NER models have become dominant and achieve state-of-the-art results. Compared to feature-based approaches, deep learning is beneficial in discovering hidden features automatically. Deep learning is a field of machine learning that is composed of multiple processing layers to learn representations of data with multiple levels of abstraction. On the other hand, it is all about representation learning with good features automatically. The key advantage of deep learning is the capability of representation learning and the semantic composition empowered by both the vector representation and neural processing. This allows a machine to be fed with raw data and to automatically discover latent representations and processing needed for classification or detection [29].

There are three core strengths of applying deep learning techniques to NER. First, NER benefits from the non-linear transformation, which generates non-linear mappings from input to output. Compared with linear models (e.g., log-linear HMM and linear-chain CRF), deep-learning models are able to learn complex and intricate features from data via non-linear activation functions. Second, deep learning saves significant effort in designing NER features. The traditional feature-based approaches require a considerable amount of engineering skill and domain expertise.

Deep learning models, on the other hand, are effective in automatically learning useful representations and underlying factors from raw data. Third, deep neural NER models can be trained in an end-to-end paradigm, by gradient descent. This property enables us to design possibly complex NER systems [30]. There are many types of deep neural networks and deep neural network architectures. This paper briefly discusses networks used in our research.

2.7.1 Recurrent Neural Networks

Traditional neural networks have a major limitation in considering the sequential relation of inputs and outputs. It is assumed that each input and outputs are independent of each other. To overcome this limitation recurrent neural networks (RNN) are proposed. While CNN's are naturally good at dealing with grids, RNNs were specifically developed to be used with sequences [31]. CNN's do allow capturing some order information, but it is limited to local patterns, and long-range dependencies are ignored[32].

The idea behind recurrent neural networks is making use of sequential information. For example, if we want to predict the named entity-tag of the current word we have to know the tag of words that occurred before it. Recurrent neural networks have a memory about what has been calculated so far and use it on current output computation. Theoretically, RNN makes use of information in arbitrarily long sequences but in practice, it is limited to few time steps [33]. As shown in Fig.2.5, an RNN can be viewed as a chain of simple neural layers that share the same parameters.

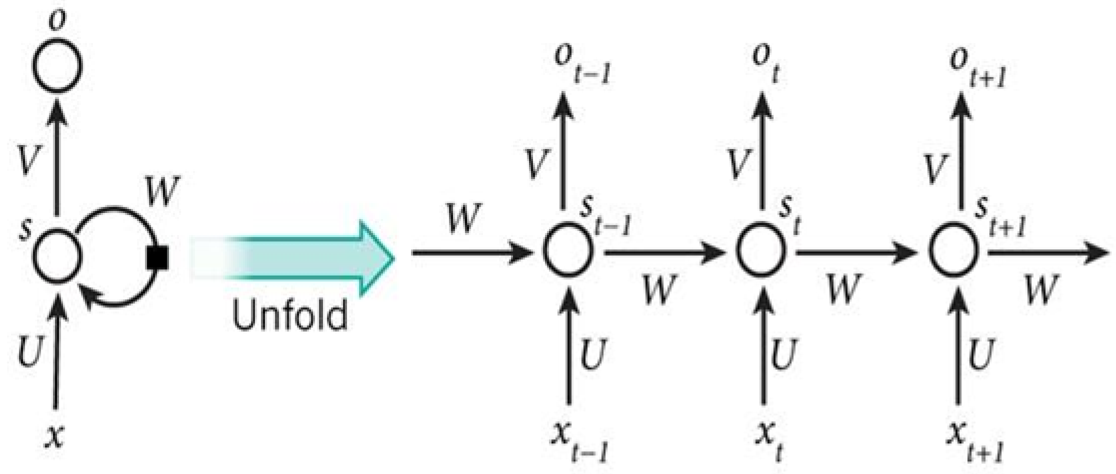


Fig. 2.5: A recurrent neural network and the unfolding in time of its forward computation

Where xt is the input at time step t, st is the hidden state at time t, and it is the memory of the network which is calculated based on the previous hidden state and the input at the current step st = f (uxt + wst−1). The function f is usually a non-linear function such as tanh or ReLu. Ot is the output at time t [33]. Unlike a traditional deep neural network, which uses different parameters at each layer, an RNN shares the same parameters (U, V, W above) across all steps. This reflects that it is performing the same task at each step, just with different inputs. This greatly reduces the total number of parameters the network needs to learn [33].

RNN has shown great success in many NLP tasks. The most commonly used type of RNNs is long short term memory recurrent neural networks (LSTM RNN), which are much better at capturing long-term dependencies than typical recurrent neural network discussed before. In this study, we apply Long Short-Term Memory [34], [35] to sequence tagging. Long ShortTerm Memory networks are the same as RNNs, except that the hidden layer updates are replaced by purpose-built memory cells. Subsequently, they might be better at finding and take advantage of long-range dependencies in the data.

2.7.2 Bidirectional LSTM Networks

In sequence tagging task, we have access to both past and future input features for a given time, we can thus utilize a bidirectional LSTM network as proposed in [35]. In doing so, we can efficiently make use of past features (via forward states) and future features (via backward states) for a specific time frame. We train bidirectional LSTM networks using backpropagation through time (BPTT)[36]. The forward and backward passes over the unfolded network over time are carried out similarly to regular network forward and backward passes, except that we need to unfold the hidden states for all time steps. We also need special treatment at the beginning and the end of the data points.

2.7.3 CRF Networks

We combine an LSTM network and a CRF network to form an LSTM-CRF model. This network can efficiently use past input features via an LSTM layer and sentence level tag information via a CRF layer. A CRF layer is represented by lines that connect consecutive output layers. A CRF layer has a state transition matrix as parameters. With such a layer, we can efficiently use past and future tags to predict the current tag, which is similar to the use of past and future input features via a bidirectional LSTM network.

2.7.4 BiLSTM-CRF Networks

Similar to an LSTM-CRF network, we combine a bidirectional LSTM network and a CRF network to form a BiLSTM-CRF network. In addition to the past input features and sentence level tag information used in an LSTM-CRF model, a BiLSTM-CRF model can use the future input features. The extra features can boost tagging accuracy as we will show in experiments.

2.8 Afaan Oromo Language

2.8.1 Background

Afaan Oromo is a mother tongue for Oromo. The Oromo people are dominantly inhabited in the Oromia region. As an indication of strong ties of the Afaan Oromo language with other regions in Ethiopia, Oromia is the region that neighbors all the regions in Ethiopia except the Tigray region. Oromoo people are also inhabited in neighboring countries like Kenya and Somalia. Afaan Oromoo is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic family. Afaan Oromo has a very rich morphology like other African and Ethiopian languages [37]. It is used by Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 34.5% of the total population [38]. The language has become the official language in Oromia regional offices and is also an instructional language starting from elementary to university level.

Afaan Oromo has a very rich morphology like other African and Ethiopian languages. The writing system of Qubee (Latin-based alphabet) has been started since 1842 [37]. It is one of the major African languages that is widely spoken and used in most parts of Ethiopia and some parts of other neighboring countries like Kenya, Tanzania, Djibouti, Sudan, and Somalia. It is the second-largest Cushitic language in the Africa continent next to Hausa. Now a day, it is an official language of the Oromia state (which is the largest Regional State among the current Federal States in Ethiopia).

2.8.2 Writing system of Afaan Oromo language

Afaan Oromo's writing system is almost phonetic since the way it is spoken in the way it is written. i.e. one letter corresponds to one sound. The language uses the Latin alphabet (Qubee) which was formally adopted in 1991[39], and it has its own consonants and vowel sounds. Afaan Oromo has thirty-three consonants, of these seven of them are combined consonant letters: **ch, dh, ny, ph, sh, ts** and **Zh**. The combined consonant letters are known as **'qubee dachaa'**. Vowels have two natures in the language and they can result in different meanings. The natures are short and long vowels. A vowel is said to be short if it is one and a long vowel if it is two, which is the maximum. For example, Laga (river) and Laagaa (throat). The

Afaan Oromo alphabet is characterized by capital and small letters like the English alphabet. In Afaan Oromo, as in the English language, vowels are sound makers and do stand by themselves [13].

Afaan Oromo and English are different in sentence structuring. Afaan Oromo uses subject-object-verb (SOV) language. SOV is the type of language in which the subject, object, and verb appear in that order. Subject-verb-object (SVO) is a sentence structure where the subject comes first, the verb-second, and the third object. For instance, in the Afaan Oromo sentence "Mooneerraan bilisa bahe". "Mooneeraa "is a subject, "bilisa" is an object and "bahe" is a verb. Therefore, it has SOV structure. The translation of the sentence in English is "Mooneerraa has got freedom" which has SVO structure. There is also a difference in the formation of adjectives in Afaan Oromo and English. In Afaan Oromo adjectives follow a noun or pronoun; their normal position is close to the noun they modify while in English adjectives usually precede the noun. For instance, namicha gaarii (good man), gaarii (adj.) follows namicha (noun).

Punctuation marks used in both Afaan Oromo and English languages are the same and used for the same purpose with the exception of the apostrophe. Apostrophe mark (') in English shows possession, but in Afaan Oromo it is used in writing to represent a glitch (called hudhaa) sound. It plays an important role in Afaan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels appeared together like "ba'e" to mean ("get out") with the exception of some words like "ja'a" to mean "six" which is identified from the sound created. Sometimes apostrophe mark (') in Afaan Oromo interchangeable with the spelling "h". For instance, "ba'e", "ja'a" can be interchanged by the spelling "h" like "bahe", "Jaha" respectively still the senses of the words are not changed [24].

As stated in [39], in addition to the 33 symbols, it is recommended to know the following principles associated with "Qubee" [39]:

1. Two vowels in succession indicate that the vowel is long, e.g. bitaa (left);
2. Gemination (a doubling of a consonant) is phonemic in Afaan Oromo, e.g. damee (branch), dammee (sweety);
3. h is not geminated at all;
4. The same word can have two or more forms depending on its context, e.g. nama kadhu (ask person), namaa kadhu (ask for person);
5. When it occurs at the end of a word, the single "a" is pronounced schwa (inverted e) whereas it is pronounced (delta) elsewhere;
6. Understandably, instead of diacritic signs, the combined letters are used so as to align them with typewriter characters.

2.9 Word Categories of Afaan Oromo

Words are the basic part of a given language. The arrangement of words or their combination depends on the rule of the grammar of that language. The combination of these words on the basis of the language gives us sentences. The meanings of these sentences depend on each word of the sentence and the way they are arranged. However, the extent to which a given word determines the meaning of a sentence depends on the contribution of that word. All words do not have equal contributions to sentence meaning. Their contribution depends on their category and their feature. Based on the category of the word; we can find out the contribution of that word. In addition to this, we can easily identify the rules to be applied to this word in case of morphological change [40].

In order to categorize a given word into some class, there are three clues that can be used. These are the meaning of the word, the form (shape) of the word, and the position or environment of the word in the sentence. The meaning of the word criterion is used to deduce the meaning the words imply. Some of these meanings can be action meanings that are conveyed by verbs, entity words that can be implied by nouns, state words that can be expressed by adjectives or that can denote the manner in which something is done.

For instance, words like **mana**, means 'house', **gurbaa** means 'boy', **Laga** means 'river' can be grouped into the same group or category based on the above criteria because they all refer to the name of a person, place, or objects which can be generalized as entities. On the basis of this common feature (i.e. entities), the above words are grouped as a noun. In a similar manner the words like **Fiigi** means 'run', **Barreessi** means 'write', **Dubbisi** means 'read' imply common actions. Based on these common actions they can be categorized as a verb. A form or shape criterion uses morphological information of the word to be categorized. The fact that certain affixes are used with a particular group of words makes this criterion possible to use.

In English, abstract nouns end with bound morpheme /-ion/ like repetition, presentation, demarcation, etc. In the same manner, adjectives end with /-able/ affix as readable, movable, tolerable, etc. Inflection is one common form of morphological change. This can be a change of singular noun to its plural form by adding morpheme –s at the end of the word or it can be a change of verb from its present form to its past form by adding /-ed/post-fix [40].

In Afaan Oromo, nouns can be inflected by adding /-oota/, /-oolee/, /-eenn/, /eewwan/ etc by adding at the end of the noun. For example, a word like **saree** which means 'dog' **sa'a** which means 'cow' form their plural as sar**oota** 'dogs' and saa**wwan** 'cows'. Some verbs can add a post-fix /-e/to change their tense into the past. For example,**qab** which means 'to catch' can change to its past form by adding /-e / as **qabe** which means 'caught'.

The other type of word categorization criterion is the position of the word in the sentence. This is also known as the environment of (surrounding of the word) the word in the sentence. It is the criterion that carefully looks at where the word has occurred in the sentence and word surrounding or neighboring.

For example, in the Afaan Oromo language, verbs come at the end of the sentence. However, adjectives come after the noun they modify. For example, **farda diima** meaning 'red horse', **Diima** is an adjective equivalent to English red whereas **Farda** is the same as 'horse' which is a noun. Therefore, if we know that **diima** acts as an adjective in the sentence we can directly deduce that the preceding word is a noun. In general, word categories can be identified by looking at the meaning (semantic) of that word, by looking at the form (morphology) of that word, or by looking at the actual position (syntactic) of that word [40].

As stated by Mohammed and other recent papers, Afaan Oromo has five-word classes: noun, adposition, adverb, conjunction, and verb. Each of these classes again can be divided into other sub-classes. For instance, noun class is categorized as a proper noun, a common noun, and pronoun. Preposition and postpositions are sub-classes of ad-positions. The subclasses in turn can be divided into subclasses and the subdivision process may continue iteratively depending on the level and aim of the investigation. In this work (including subclasses of the above classification) eight-parts of speech (i.e. noun, pronoun, adjective, preposition, adverb, conjunction, verb, and introjection) and numerals are used for word categorization. Each of these word classification is discussed below in detail.

2.9.1 Noun (Maqaa)

A noun is any word which names a person, place, thing, idea, animal, quality, or activity [41]. In Afaan Oromo most of the time a sentence begins with a noun which starts with a capital letter and it uses a noun as a subject followed with subject markers (-ni, or -n,). For instance, words that are bolded in the following sentences are nouns.
Sangaan marga dheeda. (An ox grazes the grass).
Bunni dinagdee biyyaa keenyaati. (Coffee is the economy of our country).
Moreover, two numbers are recognized in Oromo nouns: singular and plural. A singular noun is marked by zero-morphemes whereas a plural noun is marked by various forms [42]. The following are illustrative examples.

Singular	**plural**	**pl. Marker**
1. mana 'house'	manneen 'houses'	-een
2. sa'a 'cattle'	saawwan 'cattle'(pl)	-wan
3. ganda 'village'	gandoota 'villages'	-(o)ota
4. waraabessa 'hyena'	waraabeyyii 'hyenas'	-yyii
5. aanaa 'district'	aanaalee 'districts'	-lee

2.9.2 Pronoun (Bamaqaa)

In Afaan Oromoo, like in other languages, Pronouns are words that can be used in place of nouns. Pronouns are marked for number and gender. For instance, ishee 'she' is feminine (singular), isa 'he' is masculine (singular), isaan 'they' is plural and can be masculine or feminine, nuyi 'we' is plural and can be masculine or feminine are some.
For example:
Caalaan baay'ee dheeraa dha. (Chala is very tall)
Inni baay'ee dheeraa dha. (He is very tall)

2.9.3 Adjective (Ibsa Maqaa)

An adjective modifies a noun or a pronoun by describing, identifying, or quantifying words. In Afaan Oromo an adjective usually follows the noun or the pronoun which it modifies [13]. Some examples of adjectives in Afaan oromo are: gabaaba, furdaa, hedduu mara, gurraacha, adii, etc.
Caalaan **gurraacha** dha. (Chala is black)
Galaaneen **furdoo** dha. (Gelane is fat)
Gurraacha (black) and furdoo (fat) are adjectives in the above sentences

2.9.4 Verb (Xumura)

The verbs are words that expresses action, a state of being, and/or a relationship between two things. In their most powerful and normal position, they are found at the end of the sentence and belong to a distinct category. In each of the following sentences the verb is bolded:
Beektuun barattuu **dha**. (Bektu is a student)
Leensaan **dhufte**. (Lensa has come)
Caalaan sangaa **gurgure**. (Chala sold an oxe)

2.9.5 Adverb (Ibsa xumuraa)

Afaan Oromo adverbs are words that are used to describe verbs. Adverbs usually precede the verbs they modify or describe. In the language, adverbs could be categorized as adverbial time, adverbial place, and adverbial condition (manner). Adverbial time answers the question 'when?' in a sentence. Some of them are, har'a (today), bor (tomorrow) iftaan (after tomorrow) kaleessa (yesterday), bara egere (next year) and etc [41].

Tolaan **kaleessa** dhufe. (Tola came yesterday)
Caalaan **bor** deema. (Chala will go tomorrow)
In the above example bor (tomorrow) and kaleessaa (yesterday) are adverbial of time. Adverbial of place answer the question of "where?" in the sentence. Forexample, achi (there), gubbaa (above), jidduu (middle), bira (together), Gadi (below), and the like are some.
Isheen **iddoo** baankii jirti (she is at bank).
Tolasaan **fagoo** jira. (Tolasa is far away).
Iddoo (at) and fagoo (far away) are adverbs of place in the above sentences. Adverbial of manner answer the question of "how?" in the sentence. Some examples of adverbial of manner are: Baayyee (very), dafee (Fast), Cimaa (hard), suuta (slowly), qalbiidhaan (carefully).
Inni **suuta** deema. (He walks slowly)
Namichi **tasa** du'e . (The man accidentally died.)
Suuta (slowly) and tasa (accidentally) are adverbial manners in above sentences.

2.9.6 Preposition ('Dur Duubee')

Words that can have full meaning only in combination with some other words like the nouns, an adjectives or verbs are termed as pre-positions/postpositions. They do not take any affix and belong to the closed part-of speech [40]. Some of them are irra (on), Jala (below), Akka (as), Egasii (since), hamma (until).

2.9.7 Conjunction ('Walqabsistotaa')

A conjunction is a word that can be used to join or connect two phrases, clauses, and sentences. It is categorized in to coordinating conjunctions and subordinating conjunctions. Coordinating conjunctions are used to connect two independent clauses. Subordinating conjunctions are those conjunctions that are used to join the main clause with the subordinate clause [43].
garuu 'but', **moo** 'or', **kanaafuu** 'therefore', **haata'u malee** 'however/so', **ta'ullee** 'even though', **akkasumas** 'besides/in addition to' etc. are some of them.

2.9.8 Introjections

Introjections in the language are words that have unique functions to express emotion, sudden surprise, pleasure, sadness, and so on. Emotions, pleasure, sorrow or suddenly happening situations are their expressed in many languages by introjections. Afaan Oromo's introjections include **ishoo** (for happiness), **wayyoo** (for sadness), **ah** (for silent new events or situations happened).

2.10 Afaan Oromo Phrases

A phrase can be defined as a syntactic combination of one or more words. It is restricted by the terms of the constituents (elements in a phrase) and the lexical categories. According to traditional grammar, the phrase couldn't have subject and verb. Contrary to this rational idea [5] justified, a phrase must have ahead, which can be subject or verb. And phrase can be constructed from one word (i.e., headword); this criticizes other public sayings as "phrase must be constructed from two and more words". Therefore, the phrase in Afaan Oromo can be built from a group of words that has a head, with different constituencies. For example, all the following sentences are phrases.

- sangaa=ox
- sangaa gurraacha =black ox
- sangaa gurraacha guddaa= black big ox

- sangaa gurraacha guddaa kalessa argine sana =black bix ox that we saw yesterday.

All sentences in this example are noun phrase formed from a head ox, the other words are constituencies. Note that all concatenated and co-occurred words cannot be a phrase, but they should fulfill three criteria: movement, replacement, and attachment [5].

Movement

Order of words in Afaan Oromo sentences is allowed by Afaan Oromo grammar. When move/interchange place those phrases are move with each other. i.e. when it moves in sentence moves as one part. See the following examples.
a. Ispoortiin [fayyaa namaatiif] gaariidha. [water is good for health of man]
b. [Fayyaa namaatiif] ispoortiin gaariidha. In both sentence the part [Fayyaa namaatiif] [for health of man] is phrase. Changing their positions is used to take an attention to the message they are delivering.

Replacement

To be phrases, those words that move as one word must be replaced by other words. One of this known replacement word is a pronoun. Let us see the following example; "[Dargaggeyyiin seexaan guutuu]utubaa hawaasaa tokkoti"(The visionary youths are the pillar of community)
"[Isaan] utubaa hawaasa tokkoti"(they are the pillar of a community)

Interconnection(Attachment): no other word entered between the words forming phrase. Example: "Sangaa gurraacha guddaa kaleessa argine sana"(The black big ox that we saw yesterday). If we insert the word called "qalle"(slaughter) between any two words of the phrase it results ungrammatical phrase except at the end of a sentence. This means: "Sangaa qalle gurraacha guddaa kaleessa argine sana" – we can't say.
"Sangaa gurraacha guddaa kaleessa argine sana qalle"- we can say.

2.11 Afaan Oromoo phrase classes

There are five classes of phrases in Afaan Oromo, which are noun phrases, verb phrases, adjectival phrases, adverbial phrases, and prepositional phrases [44]. From word groups, certain phrases are built.

2.11.1 Gaalee maqaa (Noun phrase)

Noun phrases are consist of a noun or pronoun and other related words that modify the noun or pronoun. It consists of nouns or pronouns as a headword and other words that come after or before the noun. The simplest NP consists of a single noun (e.g. Gammachuu) or pronoun such as **Isa** [he], **Nuyi** [we], **Isaan** [they], etc. A complex NP can consist of a noun (called the head) and other constituents (like complements, specifiers, adverbial and adjectival modifiers) that modify the head from different aspects. Example:
Caalaan sangaa diimaa furdaa sana dheengadda bite
S
(NP (N Caalaa) (dP -n))
(VP (NP (NP (N sangaa) (N diimaa)) (ADJ furdaa))
VP (AVP (AVB (dheengadda)) (VB bite))

2.11.2 Gochibsii Afaan oromoo (verbal phrase)

When we say verbal phrase we must see the main components of sentences [Hima] in Afaan Oromo. In Afaan Oromo sentence structure verbs are found at the end of sentences. For example, In sentence Tolasheen xalayaa barreessite.[Tolashe wrote letter],barreessite[wrote] is head word and xalayaa [letter] is predicate. When categories sentence Leensaan xalayaa barreessite. into noun phrase and verb phrase, Tolasheen is a noun phrase and xalayaa barreesite is a verb phrase. The headword to phrase xalayaa barresssite is verb barressite that decided the phrase is a verb phrase.

2.11.3 An adjective phrase in Afaan oromoo

To take part of a sentence is as an adjective phrase or word or group of words are adjective phrases; the headword of that phrase must be adjective. Example, Horaan akkuma abbaa cimaadha. [Hora is strong as his father].akkuma abbaa cimaa is an adjective phrase in the verb phrase akkuma abbaa cimaadha.For this adjective phrase headword is a word cimaa. Because; concentration of adjective phrase akkuma abbaa cimaa is based on strengthens of Hora not about Hora like his father. That is why cimaa is headword to the phrase.

2.11.4 Adverbial phrases in Afaan oromoo

An adverb phrase is a phrase that is an adverb is headword to phrases. In Afaan Oromo, adverbs are used to modify the coming verbs. Adverbs always come before the modified verb but it should be noted that any words that come before verbs cannot be always considered as an adverb. In Afaan Oromo the the main difference of adverb phrases from other phrase is adverb phrase is never come with other words [44]. Adverbs indicate manner, time, place, cause, or degree. Sometimes it is a single word in sentences. In the following example all the underline one is adverb phrases.
Example
Yadataan**kaleessa** deeme.(Yadeta has gone yesterday)
In the above senescence, the underlined word is an adverb.

2.11.5 Prepositional phrase in Afaan oromoo

Afaan Oromo pre-and post- positional phrase (PPhr) is constructed from a pre- or post-position (PP) head and other constituents such as nouns, noun phrases, verbs, verb phrases, etc. that means no phrase construct from only pre- or post-position. In Afaan Oromo prepositional phrases are not similar to other languages like English. As in [45], there are not many possible forms for prepositional phrases in English, though adverbs can act as modifiers to prepositional phrases.
Examples: Postpositional phrases with dependent post position
A)Tabba**tti**
B)Meeshaa**dhaan**
C)Abbid**aan**
D)Farda**an**
E)Muka**rra**

In the above examples, all are postpositional phrases. For these phrases head words are the bold once. Those are dependent ones. But they have a right to guide structure that creates postpositional phrases. In the next

example let will see the prepositional and postpositional phrases that were created from independent prepositional.
Example: prepositional and postpositional phrases with independent pre and post- positional.
A)Tabba **gubbaa**
B)**Gara** kam?
C)**Gara** manaa
D)Mana **jala**
E)**Waa'ee** mataa
F)Garaa **keessa**

The bold words are headword to those phrases. Positions are either left or right of the structure. The difference of prepositional phrases from other phrases are headword can be either left or right of the structure. Also, dependent and independent prepositional can be taken as head in the structure at the same time. However, it guides the structure by order or can't guide the structure at the same time. Let us take a look at the following example Example: prepositional phrase with independent and dependent prepositional in the same structure.
Gara gaaraa**tti**
waa'ee koo**tiif**
In the above phrases a and b there is independent and dependent prepositional. But both can't guide the phrase. In both phrases the word **gara** and **waa'ee** is guide pre-and post- positional phrases **garaatti** and **kootiif**.

3. CHAPTER THREE

3.1 Related Works

Named entity recognition has been worked on quite intensively for the past few years and has been incorporated into several products. A lot of researches in NER task have been conducted for English, European, Asian, and other languages. There have been different NER systems development attempts taken by different authors throughout the world. In this section, an overview of NER will be discussed, by categorizing works in local and foreign languages.

3.1.1 Named entity recognition on local languages

We found two types of research on Afaan Oromo and five types of research on Amharic named recognition from the literature reviewed.

Mandefro Leggese conducted the first research on Afaan Oromo in 2010 and the second research conducted by Abdi Sani in 2015. The first research on Amharic was done by Moges Ahmed in 2010. The second research was done in 2013 by Besufikad Alemu. The third one was done by Mikiyas Tadele in 2014 and the fourth was done in 2017 by Dagimawi Demissie. The fifth one was done by Omer Dawed in 2017.

The researcher[12] used a supervised machine learning method and CRF model to develop an Afaan Oromo language NER system. The features used were both internal and external, normalized tokens, part-of-speech tags, word-shape features, position features, and token prefixes and suffixes. Adopted from CoNLL (Conference on Computational Natural Language Learning), four types of Afaan Oromo language NEs were identified: Person, Location, Organization, and Miscellaneous. As they described due to compatibility issues with previously proposed work for Oromo language POS tagging which is an external feature in their work they have used a ling pipe tool that is originally developed for English. The processing was classified into preprocessing, training, and recognition phase. They have reported that for all processing they have used news-based Afaan Oromo corpus which consists of 23,000 tokens from ”Bariisaa” and ”Kallacha Oromiyaa” newspapers. After generating the model they reported that on the first experiment the performance obtained is 77.41% Recall, 75.80% Precision, and 76.60% F1-measure. They have conducted their experiments by adding the corpus size and reducing features. They found that recall and F1- measure increase while precision reduced while removing features in the experiment has shown an inverse change in all performance measurements. Finally [12] reported that features play a vital role than the change in the training data size.

Also in [13] Afaan Oromo Named entity recognition was done using a hybrid approach. The rule-based component has parsing, filtering, grammar rules, white list gazetteers, blacklist gazetteers, and exact matching components. The same corpus developed by[12] was used for training. The best performance achieved from this research was 84.12% precision, 81.21% recall, and 82.52% F-Score.

A conditional random field machine learning approach was used for ANER by [46]. Feature sets used by the researcher were word and tag context features, part of speech tag of tokens, prefix, and suffix. The researcher conducted a different combinations of these features to determine the best performing feature sets. For determining these features four different scenarios were conducted. In the first scenario, all features were used. The second scenario used all feature sets except the POS tag. In the third and fourth scenarios, all features were considered except prefix and suffix respectively. From the experiments on these scenarios, the third scenario achieved the highest F-score of 74.61%. Based on the result the researcher concluded that an important combination of features for Amharic named entity recognition is the third scenario which considers all features except prefix features. The reason [46] concluded, was that POS tag and Suffix are most relevant to identify Amharic NEs. Based on the results he recommended the use of a large corpus and including a rule-based component (hybrid approach) to improve the performance of the Amharic named entity recognition system.

The researcher[11] also used conditional random field machine learning approaches. Previous and next words, named entity-tag of a word, word pairs, word shape, prefix, and suffix features were used for the experiments. The highest F-measure achieved was 80.66%. The features used were the previous word, next word, prefix, suffix, previous word NE tag, and next word NE tag. The researcher recommended using Stanford and ling pipe tools, and using a large corpus, and adding a rule-based component to improve the performance of Amharic named entity recognition.

A hybrid approach is used for ANER in [10]. Decision tree (J48) and support vector machine (SVM) was used for classification. Also, a rule-based component having two rules based on the presence of trigger words was included in the model. The features used are words, NE tag of words with a window size of two, prefix, suffix, POS tag, and nominal feature which indicates whether a word is a noun or not. From the experiments the highest F-score achieved was 96.1% for the J48 decision tree and 85.9% for SVM.

Machine learning was used by [47] to develop ANER. This paper approach differs from all researches in the process of feature generation. They used automatic feature generation using neural networks other than the NE tag feature and the approach is not dependent on any NLP tasks for features. From the experiments the highest F-score achieved was 95.5% using the SVM classifier.

Amharic Named Entity Recognition System Using Statistical Method was done by Omer in 2017[48]. The experiments were conducted on the total of 18,191 tokens for training data, development data, and test data following CoNLL2002 format, BIO tagging scheme and LingPipe library is used as a tool for developing the NER for Amharic language. They have conducted five different kinds of experiments by varying their feature sets and tested the performance of the proposed Conditional Random Field (CRF) based Amharic Named Entity Recognition (ANER) system. The highest performance achieved in their work is 78.24% which is the features of baseline experiments by increasing Previous and next words' prefix and suffix with maximum length from 2 to 4 and the worst is 68.29% which is the experiment without Part Of speech (POS) tags of tokens. Based on the experimental results they have concluded that the tool and algorithm might have impact

Tab. 3.1: Named entity recognition on Local languages

Research	Approach	Features	F1 Score(%)
Named Entity Recognition for Afaan Oromo , 2010, Mandefro Legesse	CRF, Rules	position features, Normalized features, words shape, POS tag, prefix, suffix	76.60
Afaan Oromo Named Entity Recognition Using Hybrid Approach, 2015, Abdi Sani	CRF, Rules	morphological, POS tag, word length, dot flag, capitalization, Gazetteers flags, nominal, NE tag	82.52
Named Entity Recognition for Amharic Language, 2010, Moges Ahmed	CRF	context features, POS tag, prefix, suffix	74.61
A Named Entity Recognition for Amharic, 2013, Besufikad Alemu	CRF	context features, word pairs, word shape, prefix, suffix	80.66
Amharic Named Entity Recognition Using a Hybrid approach, 2014, Mikiyas Tadele	SVM, J48	context features, POS tag, prefix, suffix, nominal feature	85.9, 96.1
Amharic Named Entity Recognition Using Neural Word Embedding as a Feature, 2017 Dagimawi Demissie	SVM	RNN, LSTM, BLSTM, SVM	95.5
Amharic Named Entity Recognition System Using Statistical Method, 2017, Omer Dawed	CRF	POS tags of tokens, prefix and suffix	78.24

on the performance of ANERs and the combination of POS tags of tokens, prefix and suffix with a length of four are important features to recognize NEs.

3.1.2 Named entity recognition on foreign languages

We will explore a little of NER tasks studied for three different foreign languages. The Deep learning approach was used by [6] on Arabic named entity recognition. They introduced a neural network architecture based on bidirectional Long Short-Term Memory (LSTM) and Conditional Random Fields (CRF) and experimented with various commonly used hyperparameters to assess their effect on the overall performance of the system. Their model gets two sources of information about words as input: pre-trained word-embedding and character-based representations and eliminated the need for any task-specific knowledge or feature engineering and obtained state-of-the-art results on the standard ANERcorp corpus with an F1 score of 90.6%.

The researcher[49] conducted Neural Architectures for Named Entity Recognition based on character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora. Their models obtain state-of-the-art performance in NER in four languages namely English, Dutch, German, and Spanish without resorting to any language-specific knowledge or resources such as gazetteers. They have used English, Dutch, German, and Spanish corpus to evaluate their approach. The results achieved were 90.94%, 81.74%, 78.76%, 85.75% F-scores on four languages respectively.

Deep learning approaches by focusing on hybrid bidirectional LSTM (BLSTM) and Convolutional Neural Network (CNN) architecture was developed for the Indonesian language[50]. The work was managed to extract the information from articles into 4 different classes; they are Person, Organization, Location, and Event and it was the first Indonesian NER that utilizes deep learning. This research was conducted by using news articles and some other articles about the history of Indonesia taken from Wikipedia. The corpus consists of 4,139 sentences and the amount of words is 81,173. It consists of 10,889 words of entity classes and 70,284 words of non-entity classes ("O"). Finally, they achieved an f-measure of 79.43% as the best accuracy score.

Tab. 3.2: Named entity recognition on foreign languages

Research	Approach	Features	F1 Score(%)
Arabic named entity recognition using deep learning approach Ismail El Bazi 2018.	BiLSTM, CRF	pre-trained word embeddings, character-based representations	90.6
Neural Architectures for Named Entity Recognition, Guillaume Lample, et al. 2016	BiLSTM, CRF	character based word representation, word vectors	90.94
Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs 2018.	BiLSTM-CNN	word embedding, N-gram	79.43

4. CHAPTER FOUR

4.1 Methodology and Design of AONER System

This chapter describes the methods and the overall design of our system which is used to conduct the experiment. It also describes the approach, data sets, and tools that we have used in this study. Finally, we will present a detailed explanation of the phases along with the subcomponents included in each phase.

4.2 Design of AONER

Figure 4.1 illustrates the architecture of the Deep Learning-based NER system for Afaan Oromo. There is no common and steady structure that represents a NER system. This should be due to one or greater of the next three reasons: the writing style and character code used in languages, and the domain dependability of the NER task. The method followed by a system can also enforce a NER system to have different architectures.

In AONER architecture, processes characterize the evolution of the system. AONER is designed in such a way that, begin with learns properties and parameters associated with NEs from the training data. It then receives input plain text and predicts the possible NEs out of the plain text. The structure has two processes the learning (training) process and testing process. Considering the behavior of Afaan Oromo language, our proposed architecture for AONER is shown below.

In AONER design, a phase is simply a module that holds related components together. Components are related to each other based on the task they perform, they're respective input and their general contribution in a process.

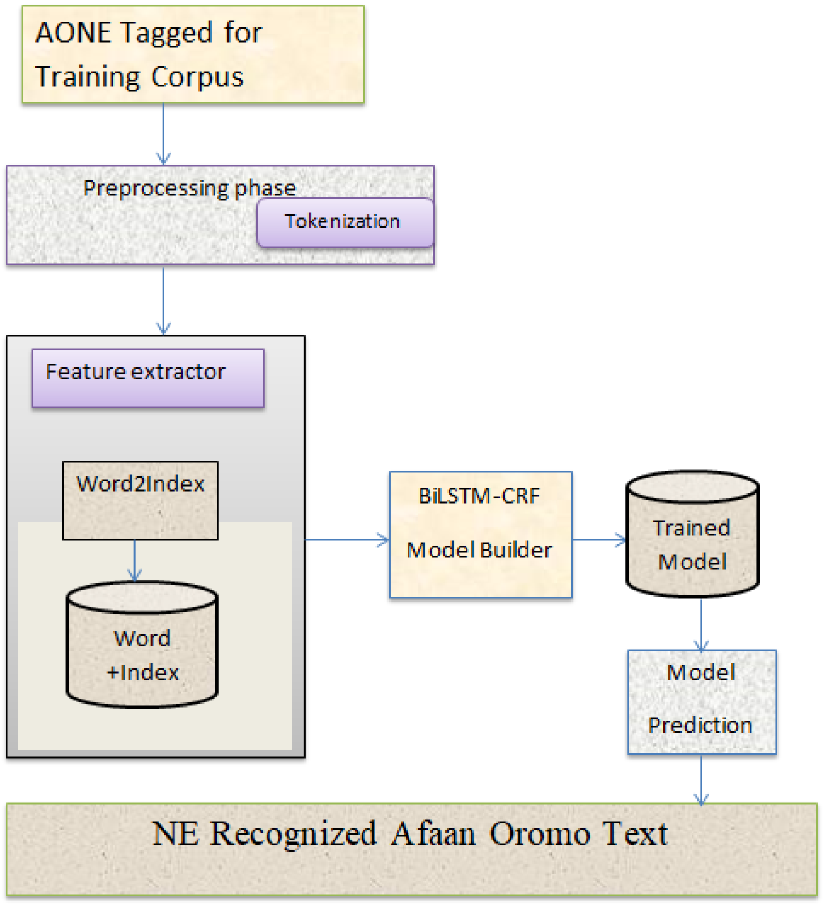


Fig. 4.1: Overall Architecture for AONER

4.2.1 Pre-processing Phase

Performing pre-processing on the collected data is a basic and important task before modeling the main AONER model. The idea behind the need for this pre-processing phase is to obtain a structured representation of the original text and to reduce the vocabulary size without removing any essential constituents of the original text.The other thing we have considered as a pre-processing task is expanding abbreviations(except for popular abbrevations) to make the corpus more informative.

The word segmentation is also another pre-processing task. Since our AONER bases its prediction and classification on the data provided, the quality of its prediction and classification depends on the quality of the data. This phase also contains "Parser" and "Tokenizer" modules that we deal with Next.

Parser

During the development of the corpus, we first went through all the obtained news articles and selected those sentences that have at least one NE. The obtained sentences are then tokenized and each token was tagged with its NE tag in accordance with the CoNLL's 2002 standard. NE tagging during this time was done manually. This makes the task prone to errors. Some of the errors that might arise are errors like failing to be consistent with the rules of CoNLL and mistyping. If the data is a victim of these errors, it will mislead the other tasks that depends on the data or even it will halt the progress into the next component. Parser is the essential component that checks the NE tagged training corpus and report if there are errors. It mainly checks the NE tagged data against the CoNLL's 2002 rules and standards.

We used CoNLL's BIO scheme for tagging the training data. CoNLL has developed a legal sequence of tags that every NE corpus should abide by. The Parser will check the data whether it is legally tagged or not. For example, for the following two tags created for the sentence "Yunivarsitiin Jimmaa magaalaa Jimmaatti argama.", the tagging on the left is legal while the tagging on the right violates the rule of the CoNLL 2002 standard. Because it has to be preceded by B-LOC tag or it has to be B-LOC.

Tab. 4.1: BIO legal Tag sequence

Tag	Legal following Tags
O	O, B-X
B-X	O, I-X, B-Y
I-X	O, I-X B-Y

Yunivarsitiin	NNP	B-ORG	Yunivarsitiin	NNP	B-ORG
Jimmaa	NNP	I-ORG	Jimmaa	NNP	I-ORG
Magaalaa	NNP	O	Magaalaa	NNP	O
Jimmaatti	NNP	**B-LOC**	Jimmaatti	NNP	**I-LOC**
argama	VB	O	argama	VB	O
.	.	O	.	.	O

The above Table shows that during the legal tagging of the BIO scheme begin (B) and inside(I) tags have the same legal followers. B-Y is to refer to the first token of another NE type. In addition to illegal annotating, other errors might happen like:

- The distance between a word and its tag might be larger than a single white space.
- There might not be a space between a word and its tag.
- A word might be left untagged.

The corpus has to be parsed and checked against all of these. Failing to do so makes the BIO Encoder to wrongly treat the words and their tags during the process of encoding. This, in turn, results in faulty feature extraction, wrong parameter estimation, and generates a faulty model. Once the training data is checked by the parser and everything is found to be correct, the data is made ready for generating the tokens and the tags, token/tag sequence, which is done by the BIO Encoder [48].

Once the training data is checked by the parser and everything is found to be correct, the data will be made ready for generating the tokens and the tags, token/tag sequence, which is done by the BIO Encoder.

Tokenizer

Tokenization can basically be defined as the process of breaking up a text into its constituent elements, tokens and is the sub-task of the pre-processing module. Hence tokenization is a process of segmenting a given text document into a piece of sentence, word, sub-word. For our purpose the model follows a three level chunking process. The first level is segmenting the input document into a set of sentences. The sentences are segmented based on language-dependent sentence delimiters such as "., ? ,!". The second level is segmenting the sentence into a set of words. The process of segmenting the sentence into words is based on whitespace. The final step is segmenting words into their sub-word constituents.

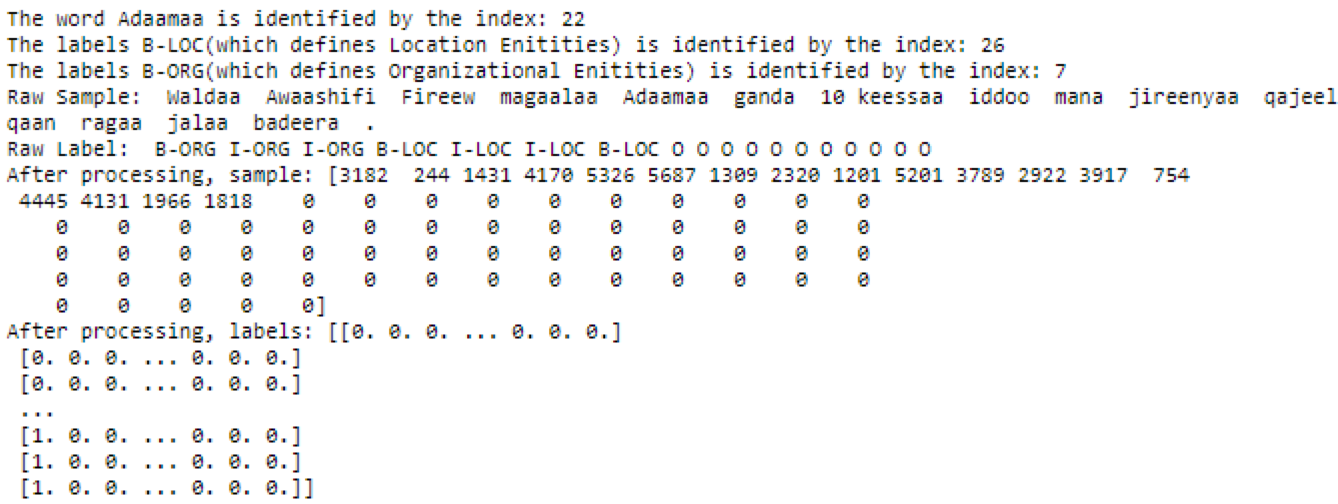
In fact, there are circumstances in which two words are treated as a single word when they are joined to each other by a hyphen (-) and they are commonly called "jecha tishoo". For example "gara-jabeessa" and "bu'a-qabeessa" are treated as a single word. Tokenization on our corpus is done during corpus development. The tokenizer will treat doublewords that are connected to each other by a hyphen (-) as separate words and the hyphen mark is treated as a token. Though a hyphen does not have a direct effect on NEs, treating it as a token will enable us to group all punctuations under a single category. Forexample, the word "af-gaaffii" has three tokens. The tokenizer treats words that contain an apostrophe (') as a single word as in "Ya'aaballoo".

4.2.2 Feature Extractor

This stage is where the words semantic and syntactic relations are learned. In the input formatting tasks such as converting segments to id, extracting embedding to respective tokens, positional encoding, and merging token and positional embedding are performed. These all learnings are learned by the embedding layer of the neural networks which learns the pattern-based learning system.

The input formatting layer accepts the tokenized input and converted to ID. Hence, each segment is mapped to its ID in the dictionary using word to index as shown below for sample. This dictionary has all the unique words(terms) as keys with a corresponding unique ID as values. It produces the fixed sized word2index representation for each word. The training sytem is held depending on the type of part of speech tagging of each word and predict the category of entity type. After the training process is done, all the words with their corresponding features will be logged to an output file.

Lookup operation is the key operation in our feature extractor. Tokenized words from preprocessing stage are taken and retrieves corresponding feature vector from the output of Word2Index tool. It reads tagged words for training purpose and retrieves their word vector, then combines it with its named entity tag for training and testing the model.



```
The word Adaamaa is identified by the index: 22
The labels B-LOC(which defines Location Enitities) is identified by the index: 26
The labels B-ORG(which defines Organizational Enitities) is identified by the index: 7
Raw Sample:  Waldaa  Awaashifi  Fireew  magaalaa  Adaamaa  ganda  10 keessaa  iddoo  mana  jireenyaa  qajeel
qaan  ragaa  jalaa  badeera  .
Raw Label:  B-ORG I-ORG I-ORG B-LOC I-LOC I-LOC B-LOC O O O O O O O O O O O
After processing, sample: [3182  244 1431 4170 5326 5687 1309 2320 1201 5201 3789 2922 3917  754
 4445 4131 1966 1818    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0]
After processing, labels: [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [1. 0. 0. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]]
```

Fig. 4.2: How Word2Index works

4.2.3 Training phase

The training phase contains necessary components that are used in the process of training the AONER component. Training file containing words with their feature vector, POS, and named entity tag are the input for this process. After the training data is fed, the model building process starts to form a model considering the features and their named entity tag.

The experimental result in this model is carried on the basis of word2Index which has all the unique words as keys with a corresponding unique ID as values. The embeddings are trained on raw data which gives the model more on understanding the language syntactic and semantic features. Features are represented in the form of floating numbers, consequently algorithms could be used.

4.2.4 prediction Model

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 word2Index of words in a given plain Afaan Oromo text. It predicts the named entity by taking this input, Tag of a word, so with their expected tag, the output is target words.

5. CHAPTER FIVE

5.1 EXPERIMENTATION AND RESULT ANALYSIS

Below we present our proposed model experimental setup and empirical evaluation. The experimental set-up, tools, training procedure, and dataset preparation and evaluation metrics are described. We have included the result of the proposed model and analysis of the model. the chapter also covers the discussion on the result.

5.2 Dataset Collection and Preparation

Any kind of research conducted in NLP requires resources and among those resources the most important and vital one is data. As it has been defined in [3], [5], [6] NLP that uses a Deep learning approach needs a huge amount of data. The success or failure of most NLP applications depends on the quality and availability of appropriate data.

Since most of the study on the state of the art is Using the corpus-based approach with different ML algorithms, it is evident why data is important. However, low resourced language like Afaan Oromo doesn't have this kind of pre-trained model which is publicly accepted as language representation. Due to this problem, we have prepared the corpora from different sources that include OBN (Oromia Broadcasting Network), OMN (Oromia, Media Network), VOA (Voice Of America), newspapers (Bariisaa, Bakkalcha Oromiyaa, and Oromiyaa), social media like Facebook pages of legal media. Having this in mind we have collected 700 Afaan Oromo sentences to build AONER. AS shown in figure 5.1 below we've selected a sentence in such a way that a sentence has to have at least one entity type. From the total of 14,208 tokens, 5047 are identified entity names.

Entities identified	**Number**
Person	646
Location	903
Organization	2076
Miscellaneous	1422
Other	9161
Total NE	5047
Total Size	14,208

Tab. 5.1: Overview of Tagged Entities

Besides data are collected from different categories it includes all categories of news, literature, office work letters, and the bible are used. After collecting the datasets, we have undertaken some basic pre-processing tasks on the entire dataset.

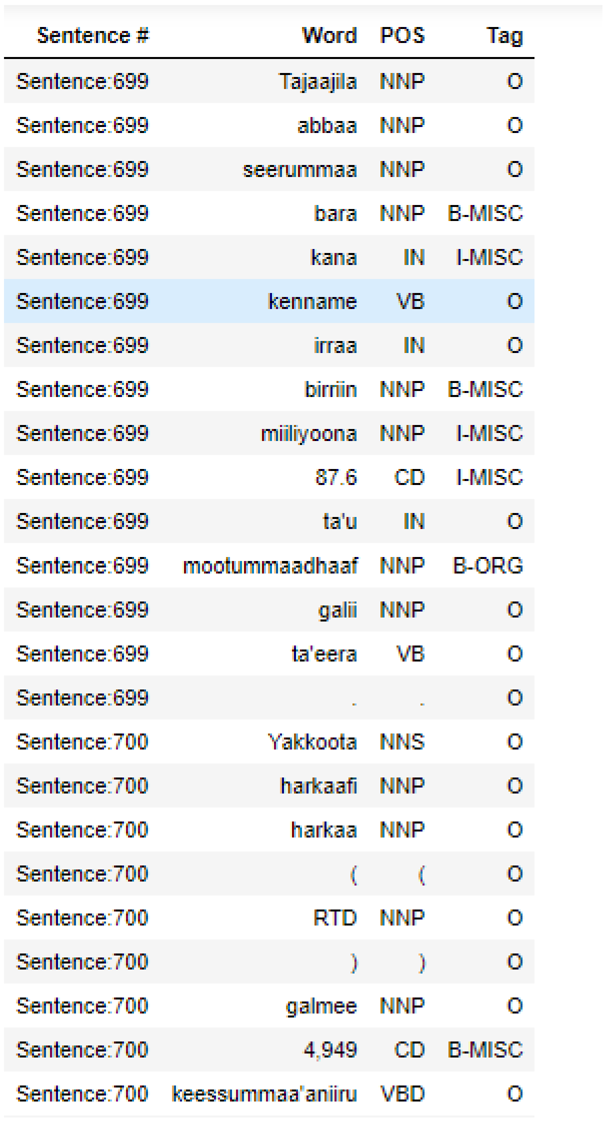
5.3 Data Format Sample

Sentences that contain at least one NE have been taken from data sources for our corpus and annotated with NE tag sets with BIO notation. The data used was already subjected to some linguistic preprocessing as tokenization and manual named entity tagging of training data, development, and test data following CoNLL2002 format.

Every word in a sentence will be kept on one line together with the POST and Named Entity tag. Since for the Afaan Oromo language, there is no widely prepared and readily available tagset for natural language processing purposes, we have used the tagsets built for English and localized it with the assistance of computational linguistics for the Afaan Oromo part of speech tagging(POST)[25].

The other reason why we follow English tagsets than using Afaan Oromo tagsets is that there are no widely agreed-upon tagsets for Afaan Oromo as far as our knowledge is concerned, even if some tagsets are developed by few researchers.

The sample format is taken from the training data file is shown below:



Sentence #	Word	POS	Tag
Sentence:699	Tajaajila	NNP	O
Sentence:699	abbaa	NNP	O
Sentence:699	seerummaa	NNP	O
Sentence:699	bara	NNP	B-MISC
Sentence:699	kana	IN	I-MISC
Sentence:699	kenname	VB	O
Sentence:699	irraa	IN	O
Sentence:699	birriin	NNP	B-MISC
Sentence:699	miiliyoona	NNP	I-MISC
Sentence:699	87.6	CD	I-MISC
Sentence:699	ta'u	IN	O
Sentence:699	mootummaadhaaf	NNP	B-ORG
Sentence:699	galii	NNP	O
Sentence:699	ta'eera	VB	O
Sentence:699	.	.	O
Sentence:700	Yakkoota	NNS	O
Sentence:700	harkaafi	NNP	O
Sentence:700	harkaa	NNP	O
Sentence:700	(	(	O
Sentence:700	RTD	NNP	O
Sentence:700	)	)	O
Sentence:700	galmee	NNP	O
Sentence:700	4,949	CD	B-MISC
Sentence:700	keessummaa'aniiru	VBD	O

Fig. 5.1: Sample of our Data Format

5.4 Experiment and result analysis

Tools

In order to develop a model system, python is used as a programming tool that runs on the anaconda platform for python programming and Jupyter notebook. If there is one tool that every data scientist should use or must be comfortable with, it is Jupyter Notebooks. Jupyter Notebooks are powerful, versatile, shareable, and provide the ability to perform data visualization in the same environment. Jupyter Notebooks allow us to create and share documents, from codes to full reports. They help to streamline our work and enable more productivity and easy collaboration. Due to these and several other reasons, Jupyter Notebooks are one of the most popular tools.

Python has enjoyed a steady rise to fame over recent years and is now jostling for the position of one of the most popular programming languages in the world. Favoured for applications ranging from web development to scripting and process automation, Python is quickly becoming the top choice among developers for artificial intelligence(AI), machine learning, and deep learning projects.

One of the aspects that makes python such a popular choice in general, is its abundance of libraries and frameworks that facilitate coding and save development time. It is renowned for its concise, readable code, and is almost unrivaled when it comes to ease of use and simplicity, particularly for new developers. Python is an open-source programming language and is supported by a lot of resources and high-quality documentation.

The libraries that we've use to develop the model are illustrated below. Pandas is a Python data analysis library and is used primarily for data manipulation and analysis. It comes into play before the dataset is prepared for training.

TensorFlow is one of the best library available for working with Machine Learning on Python. Tensorflow makes ML model building easy for beginners and professionals alike.

Keras is one of the most popular and open-source neural network libraries for Python. Keras features several of the building blocks and tools necessary for creating a neural network such as: Neural layers, Activation and cost functions, Objectives, Batch normalization, and Dropout.

Scikit-learn is another actively used machine learning library for Python. It includes easy integration with different ML programming libraries like Numpy and Pandas. Scikit-learn comes with the support of Classification.

The Numpy library for Python concentrates on handling extensive multi-dimensional data and the intricate mathematical functions operating on the data. Numpy offers speedy computation and execution of complicated functions working on arrays.

LaTeX is a language to describe documents. It is a typesetting engine. Unlike Word, which is controlled by mouse, various menus and buttons, LaTeX uses text commands to define the way your document is typeset. Superficially, one of the advantages of LaTeX over Word or OpenOffice is the high typographical quality of the documents that you'll be able to produce.

While developing our systems we used a computer with the following specifications:
Windows® 10 Ultimate™ 64-bit operating system
RAM size 4 GB
Hard disk size 1TB and
Intel (R) Core(TM)i3-6100U processor 2.30GHz.

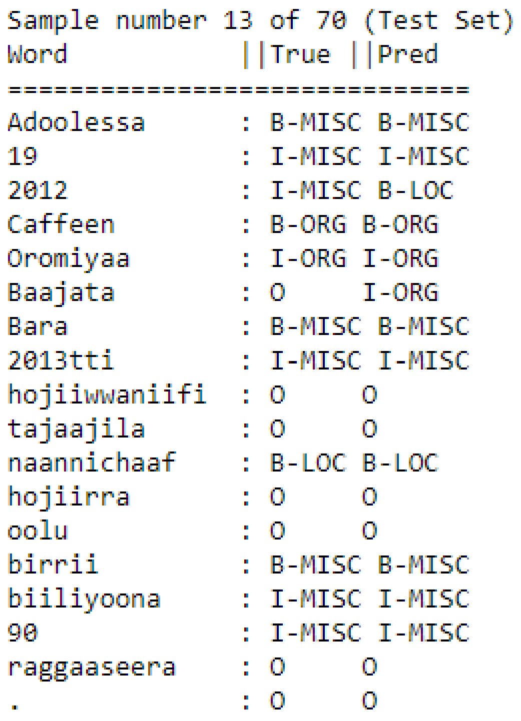
5.5 Evaluation Metrics For NER

A common approach for evaluating machine learning models is through comparison of predicted outputs of the model with that of labeled data by humans. It employs a means of comparing the output of the system against annotated data by trained language analysts. There are many ways to determine the accuracy of the model.

In our case, we divide the dataset into a training set and test set. Build the model on the training set and then use the test set as a holdout sample to test the trained model using the test data. Compare the predicted values with the actual values by calculating the error. Error is defined as actual or observed value minus the predicted value. This measure is easy to understand because it provides the error in terms of percentages. The performance of our model is measured by using accuracy which is the ratio of the number of correctly classified instances to the total number of instances.

$$Accuracy = \frac{(CorrectEntitiesIdentified)}{(TotalEntitiesIdentified)} x100$$

Evaluation for test dataset was done manually by taking randomly generated 14 sentences (just for sample) from the test dataset as a sample as shown below. For instance in the below sentence, from the total of 18 tagged tokens 16 tags are correctly tagged. Therefore, the accuracy for this sentence is (16/18)x100= 88.8%. The model scored 92% accuracy on training data and 74.14% accuracy on test dataset with the BiLSTM-CRF.

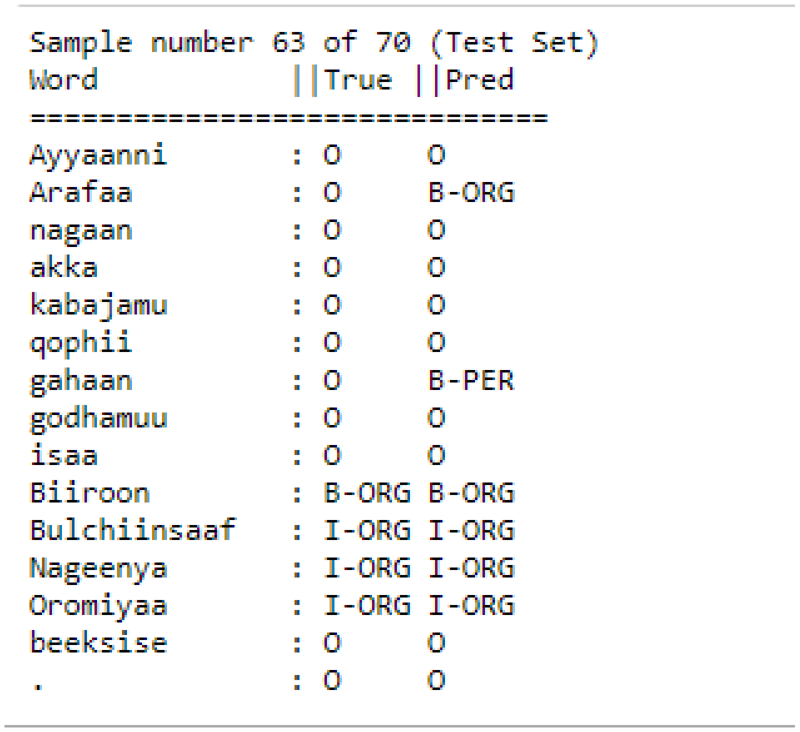


```
Sample number 13 of 70 (Test Set)
Word           ||True ||Pred
==============================
Adoolessa      : B-MISC B-MISC
19             : I-MISC I-MISC
2012           : I-MISC B-LOC
Caffeen        : B-ORG B-ORG
Oromiyaa       : I-ORG I-ORG
Baajata        : O     I-ORG
Bara           : B-MISC B-MISC
2013tti        : I-MISC I-MISC
hojiiwwaniifi  : O     O
tajaajila      : O     O
naannichaaf    : B-LOC B-LOC
hojiirra       : O     O
oolu           : O     O
birrii         : B-MISC B-MISC
biiliyoona     : I-MISC I-MISC
90             : I-MISC I-MISC
raggaaseera    : O     O
.              : O     O
```

Fig. 5.2: Accuracy for Test dataset

How the model predicts on Test Set

We have imported the data from a .csv file and then split it across three sets: Train, Validation, and Test. The train data will be used to train the model while the validation model will be used to test the fitness of the model. The Test Set must not be involved in the training exercise at the parameter or hyperparameter level. From the AONER corpus of around 700 Afaan Oromo sentences, 567 sentences were used for training, 63 sentences for validation and, 70 sentences for testing of our work. It tests randomly selects 70 sentences for testing the model and at each execution, it will test on a different sample as is shown below.



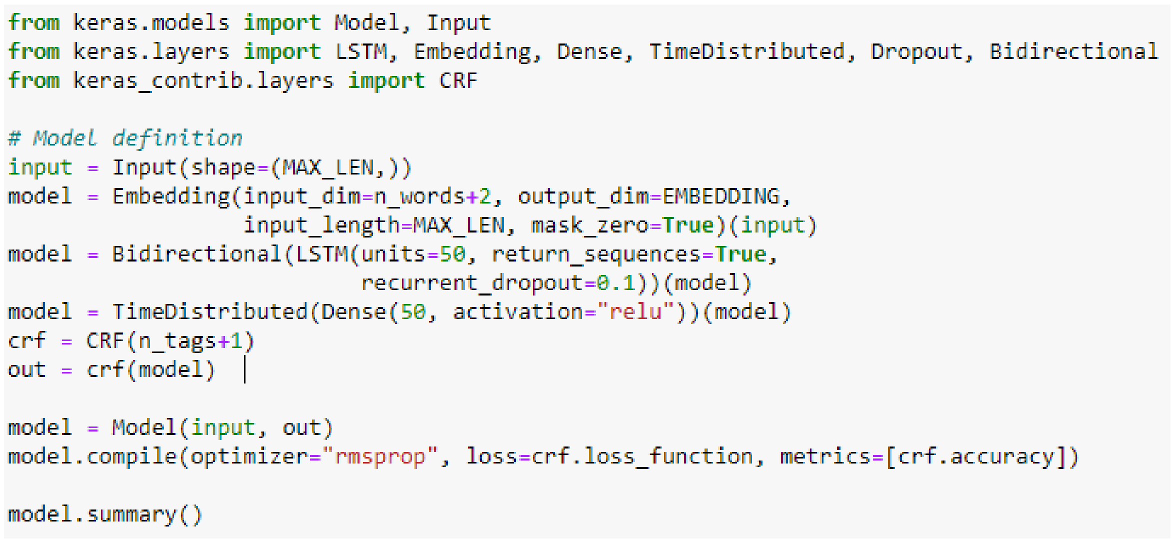
```
Sample number 63 of 70 (Test Set)
Word           ||True ||Pred
==============================
Ayyaanni       : O     O
Arafaa         : O     B-ORG
nagaan         : O     O
akka           : O     O
kabajamu       : O     O
qophii         : O     O
gahaan         : O     B-PER
godhamuu       : O     O
isaa           : O     O
Biiroon        : B-ORG B-ORG
Bulchiinsaaf   : I-ORG I-ORG
Nageenya       : I-ORG I-ORG
Oromiyaa       : I-ORG I-ORG
beeksise       : O     O
.              : O     O
```

Fig. 5.3: model prediction on Test Set

5.6 Discussion of Experimental Results

As described under section 5.2, we used 700 AONER dataset. All experimental setups are done using human-annotated datasets with the help of computational linguistics. We used 10-fold cross-validation to train and validate all experimental setups over 10% training splits. Then, we split the dataset in 90% training and 10% test set using the train-test split technique after shuffling.

In our experiment the performance of our model is measured by using accuracy which is the ratio of the number of correctly classified instances to the total number of instances. To train the model batch size of 128 and 500 epochs is used. We have also seen the results with various activation functions and optimizers.



```python
from keras.models import Model, Input
from keras.layers import LSTM, Embedding, Dense, TimeDistributed, Dropout, Bidirectional
from keras_contrib.layers import CRF

# Model definition
input = Input(shape=(MAX_LEN,))
model = Embedding(input_dim=n_words+2, output_dim=EMBEDDING,
                  input_length=MAX_LEN, mask_zero=True)(input)
model = Bidirectional(LSTM(units=50, return_sequences=True,
                           recurrent_dropout=0.1))(model)
model = TimeDistributed(Dense(50, activation="relu"))(model)
crf = CRF(n_tags+1)
out = crf(model)

model = Model(input, out)
model.compile(optimizer="rmsprop", loss=crf.loss_function, metrics=[crf.accuracy])

model.summary()
```

Fig. 5.4: model parameters

Finally, we got a better result while using ReLu(Linear activation function) which is computationally efficient means it allows the network to converge quickly and non linear that has a derivative function and allows for back-propagation as an activation function and RMSprop which is good, fast and very popular optimizer. RMSProp(Root Mean Square Prop), is an adaptive learning rate optimization algorithm. It utilizes the magnitude of the recent gradient descents to normalize the gradient. In RMSProp learning rate gets adjusted automatically and it chooses a different learning rate for each parameter.This algorithm divides the learning rate by the average of the exponential decay of squared gradients.

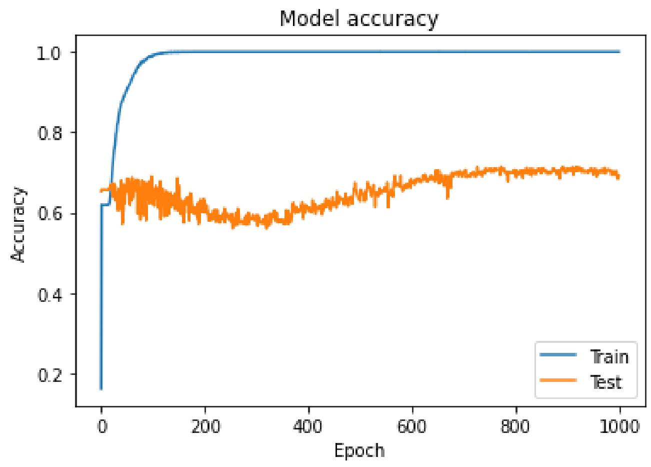


Fig. 5.5: model

The Neural Network (NN) language models are essential since they excel at dimension reduction of word representations and thus help improve performances in NLP applications immensely. Long Short-Term Memory (LSTM) neural networks and BiLSTMs are used in the model. LSTMs can learn long-term dependencies through a special unit called a memory cell, which not only can retain information long time but has gates to control which input, output, and data in the memory to preserve and which to forget. Extensions of this are bi-directional LSTMs, where instead of only learning based on past data, as in unidirectional LSTM, learning is based on past and future

Tab. 5.2: Different Model results for Trainig

Experiments	precision(%)	Recall(%)	f1-score(%)	Accuracy(%)
LSTM-CRF	89	88	89	92
BiLSTM-CRF	91	92	91	92

information, allowing more freedom to build a contextual language model.

For achieving the best results, Bi-LSTM and CRFs models are combined with a word-level embedding in a structure. A pre-trained look-up table produces word embeddings, and a separate Bi-LSTM for each word sequence renders a character-level embedding, both of which are then combined to acquire as word representations. These vectors then become the input to a bi-directional LSTM, and the output of both forward and backward paths, are then combined through an activation function and inserted into a CRF layer. This layer is ordinarily configured to predict the class of each word using BIO-format (Beginning-Inside-Outside).

The other reason why we choose the BiLSTM-CRF model is its ability to predict on the test dataset is very high compared to LSTM-CRF. This model can use both past and future input features thanks to a bidirectional LSTM component. Also, this model can use sentence-level tag information due to a CRF layer.

6. CHAPTER SIX

6.1 Conclusion and Recommendation

6.2 Conclusion

The main conclusions of this work are drawn together and presented in this section.

Named Entity (NE) Recognition (NER) is to classify every word in a document into some predefined categories and in the taxonomy of computational linguistics tasks, it falls under the domain of "information extraction", which extracts specific kinds of information from documents. NER has found many application areas such as information retrieval and extraction systems, automatic summarization, question-answering system, machine translation and others.

The AONER system is designed based on a deep learning approach: neural network-based. We implemented the deep learning component using BiLSTM-Crfs algorithm. For training the system, we developed Afaan Oromo Named Entity corpus of 700 sentences which is around 14,200 tokens. The corpus is developed using CoNLL's 2002 standard. We have distinguished four NE categories in this research work. These are person, location, organization and miscellaneous. The implemented system has different components. This technique has been validated and standardized by parser as recommended by CoNLL's standards.

The BIO Encoder component distinguishes tokens from their labels and produces token/tag sequence. The produced token/tag sequence will be provided to Feature Extractor components. Feature Extractor extracts features using POST as a classifier from the produced tokens/tag sequence. The output of the feature extractor is used to generate Afaan Oromo NE model trained with Afaan Oromo texts. The model generation is done by the Model Builder. The input plain text entered is first preprocessed and provided to the NE feauture extractor. The feauture extractor extracts the possible NE tokens from the preprocessed text with the assistance of the trained model. Finally, our model achieved 74.14% accuracy using BiLSTM-CRF which is a promising result.

This work is the first of its type for the Afaan Oromo language using deep learning approach. Generally, all the features mentioned above gave us the best combination of features for AONER system.

6.3 Contribution Of The Work

The main contributions of this thesis works are:

- We have presented an innovative approach for the Named Entity Recognition problem and the proposed model scores a promising performance for the Afaan Oromo language.
- We have shown that deep learning models can be adapted and work for Afaan Oromo Named Entity Recognition.
- Supporting the development of Afaan Oromo NE corpus with a realistic scope. On the other hand, we have prepared 700 manually annotated Afaan Oromo Named Entity Recognition datasets which is around 14,200 tokens that can be used for evaluating subsequent similar works.
- Our work is the first to apply a bidirectional LSTM CRF (BI-LSTM-CRF) model for AONER.

6.4 Recommendation

The development of a corpus for Afaan Oromo is in its newborn child stage. Hence, there are several areas of research for Ethiopian language and Afaan Oromo in specific that should be recommended for future researchers in the area of natural language processing.The developed AONER system has portions that require further improvements that we want to recommend them as listed below.

- As the researchers noted earlier, the standardized and promptly accessible corpus is exceptionally vital for natural language processing. Therefore, preparation of a standardized corpus is recommended for further researches in the area of natural language processing for Afaan Oromo.

- Since the study covers some of the named entities such as person name, organization, location, and miscellaneous without considering the subcategories: so other researchers can continue this study by incorporating these categories.

- In this work, the small size of the corpus was used for training entity recognition. Hence, another research that uses a large size corpus than ours is recommended to improve the accuracy of the work.

6.5 Future Work

NER systems are vital components for systems like Information Retrieval, Question-Answering, and Text Summarization. The task is very complex for such under-resourced languages. Actually, NEs carry important information about a document or a sentence. Since several issues remain unaddressed, a future extension is suggested for the developed Afaan Oromo NER framework as future works.

1. The framework developed in this research work is just a model. Any interested person can extend to develop a full-fledged Afaan Oromo framework that can be easily integrated into diverse applications.

2. The system developed in this research is just a prototype. Any interested person can do a project to develop a full-fledged Afaan Oromo NER that can be simply integrated into completely different Afaan Oromo NLP works like;

   Question Answering

   Grammar checkers

   Machine translation

   Search Engines (Semantic based)

   Indexing for Information Retrieval

   Dictionaries and etc.

BIBLIOGRAPHY

[1] K. K. Diksha Khurana, Aditya Koli and S. Singh., "Natural Language Processing: State of The Art, Current Trends and Challenges Department of Computer Science and Engineering," *Manav Rachna International University*, vol. 7, no. 200, 2017.

[2] J. Allen, "Natural language understanding." *California: Redwood, Benjamin/Cummings Publishing Company, Inc*, vol. 2, no. 200, 1996.

[3] R. S. Francois Ch., Rohit M., *Deep Learning for NLP*, 2017.

[4] Liddy., "Natural Language Processing." *In Encyclopedia of Library and Information Science*, vol. 2, 1998.

[5] A. Barkeessaa, "Semmoo: Bu'uura Barnoota Afaaniifi Afoola Oromoo," *Far East Trading PLC*, 2014.

[6] I. E. B. N. Laachfoubi, "Arabic named entity recognition using a deep learning approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, 2009.

[7] J. Zhou, GuoDong; Su, "Named Entity Recognition using HMM-based Chunk Tagger." *Proceedings of the 40th Annual Meeting of the ACL, Philadelphia*, vol. 7, 2002.

[8] V. Y. . S. Bethard, "A survey on recent advances in named entity recognition from deep learning models." *University of Arizona*, 2018.

[9] R. Grishman, "Information Extraction: Techniques and Challenges," *Computer Science Departmet New York University*, 1996.

[10] T. Mikiyas, "Amharic named entity recognition using a hybrid approach," *Master's thesis, AAU*, 2014.

[11] A. Besufikad, "A named entity recognition for amharic," *Master's thesis, AAU.*, 2013.

[12] M. Legesse, "Named Entity Recognition for Afaan Oromo." *Master's Thesis, School of Graduate Studies, Addis Ababa University.*, 2012.

[13] S. Abdi, "Afaan Oromo named entity recognition using a hybrid approach." *Computer Science, Addis Ababa University, Addis Ababa.*, 2015.

[14] W. X. Z. Huang and K. Yu, "Bidirectional lstm-CRF models for sequence tagging," *Baidu research*, 2015.

[15] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, 2016.

[16] A. Ekbal and S. Bandyopadhyay, "Named Entity Recognition using Support Vector Machine."

[17] G. Chowdhury, "Natural language processing." *Annual Review of Information Science and Technology,*, no. 3, 2003.

[18] R. Bose, "Natural Language Processing: Current State and Future Directions," *International Journal of the Computer, the Internet and Management,*, vol. 12, no. pp 1 – 11., January – April 2004.

[19] J. Cowie and W. Lehnert, "Information Extraction," *In Communications of the ACM*, 1996.

[20] U. A.Salini, "Named Entity Recognition Using Machine Learning Approaches." *Special Issue 11*, vol. Vol. 6, September 2017.

[21] G. . Wilks., " Information extraction: beyond document retrieval." *Journal of documentation*, 1998.

[22] e. Yonatan Aumann, "Visual Information Extraction.Knowledge and Information Systems archive," *Springer-Verlag New York, Inc. New York, NY, US.*, vol. Volume 10 ,, 2006.

[23] D. E. Appelt., "Introduction to information extraction," *AI Communications,*, 1999.

[24] K. Kaiser and S.Miksch., "Information Extraction," *Vienna University of Technology Institute of Software Technology Interactive Systems.*, 2005.

[25] D. J. . J. H. Martin, *Speech and language processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd Edition, Pearson Prentice Hall, New Jersey*, 2009.

[26] R. Sharnagat, "Named entity recognition: A literature survey." *Center For Indian Language Technology.*, 2014.

[27] D. Nadeau and S. Sekine, "A Survey of Named Entity Recognition and Classification," *New York University*, 2007.

[28] M. . Mamat., "Named Entity Recognition Approaches," *IJCSNS International Journal of Computer Science and Network Security,*, vol. VOL.8, no. No.2., 2008.

[29] Y. B. Y. LeCun and G. Hinton, "Deep learning," Nature," *vol. 521*, vol. vol. 521, no. no. 7553, 2015.

[30] J. H. Jing Li, Aixin Sun and C. Li., "A Survey on Deep Learning for Named Entity Recognition." *DRAFT IN PROGRESS*, vol. VOL. XX, no. NO. XX., 2018.

[31] J. L. Elman, "Finding structure in time." *Cognitive Science*, vol. 14:2, 1990.

[32] Y. Goldberg, "A primer on neural network models for natural language processing." *Journal of Artificial Intelligence Research,*, no. 57, 2015.

[33] D. Britz, "Recurrent neural networks tutorial," *part 1–introduction to rnns.*, vol. 32, no. 2, 2015.

[34] S. Hochreiter and Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, 1997.

[35] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013.

[36] M. Boden, "A guide to recurrent neural networks and backpropagation," *the Dallas project*, 2002.

[37] G. B. Gragg, "Oromo of Wollega: non-semetic languages of Ethiopia," *East Lansing, Michigan state university press*, 2006.

[38] A. Abebaw, "Census Report." *Ethiopia[U+201F]s population now 76 million", http://ethiopolitics.com/news,*, 2008.

[39] T. Gamta, "The Oromo language, and the Latin alphabet, Journal of Oromo Studies," *http://www.africa.upenn.edu/Hornet/Afaan*$_O$*romo*$_1$9777.*html*, *October*31, 2014.

[40] M.-H. Abubeker., "Part Of Speech Tagger For Afaan Oromo Language using Transformational Error Driven Learning (Tel) Approach," *Addis Ababa University School Of Gradute Studies Faculty Of Informatics Department Of Computer Science*, 2010.

[41] M. Getachew, "Part-Of-Speech Tagging for Afaan Oromo Language using Transformational Error Driven Learning Approach." *MSc Thesis. Addis Ababa University, Addis Ababa*, 2009.

[42] D. Megersa., "Automatic Sentence Parser For Oromo Language Using Supervised Learning Technique," *Addis Ababa University School Of Graduate Studies*, 2002.

[43] I. Nagi., "Caassefama Afaan Oromo," *Kuraz International, Addis Ababa.*, 2006.

[44] G. R., "FURTUU: Seerluga Afaan Oromoo(Oromo Grammar," 2009.

[45] B. McCaul and A. Sutherland, "Predictive text entry in immersive environments," *Proceedings of the IEEE Virtual Reality*, 2004.

[46] A. Moges, "Named entity recognition for the Amharic language." *Master's thesis, Addis Ababa University.*, 2010.

[47] M. Dagimawi, " Amharic named entity recognition using word embedding as a feature." *Master's thesis, AAU.*, 2017.

[48] D. Omer, "Amharic Named Entity Recognition System Using Statistical Method," *BDU*, 2017.

[49] B. M. S. S. K. K. Lample, G. and Dyer, "Neural architectures for named entity recognition," *ADDIS ABABA*, 2016.

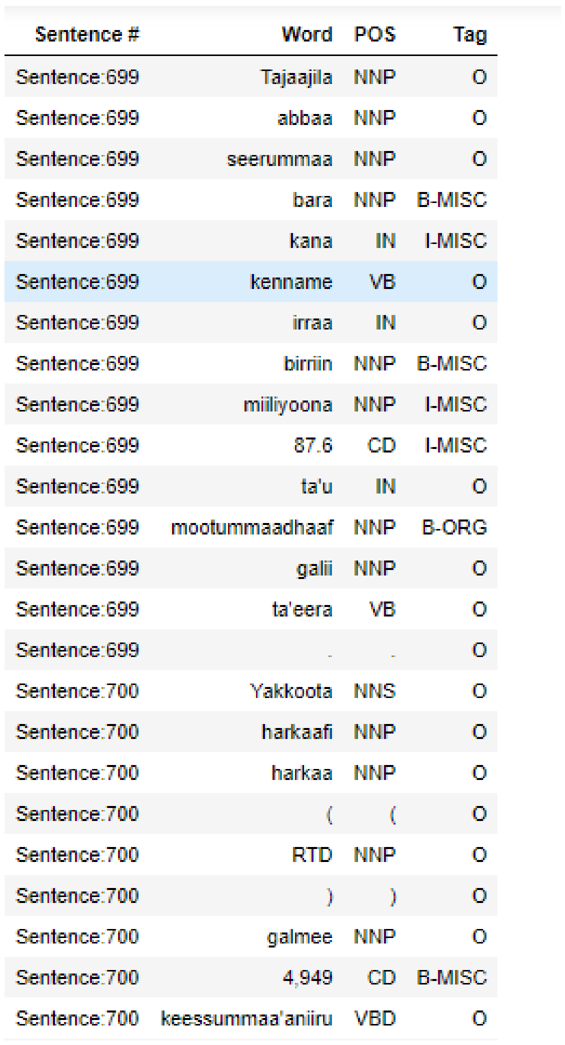
[50] S.-Q. Y. Hui-Kang Yi, 2Jiu-Ming Huang, "A Chinese Named Entity Recognition System with Neural Networks," *ITM Web of Conferences 12*, 2017.

6.6 *Appendix A: Name of Days and Months in Afaan Oromo*

Days:

Afaan Oromoo	English
Wiixata/Dafinoo	Monday
Facaasaa	Tuesday
Roobii	Wednesday
Kamisa	Thursday
Jimaata	Friday
Sambata	Saturday
Dilbata	Sunday

6.8 *Appendix C: Sample corpus for AONER*



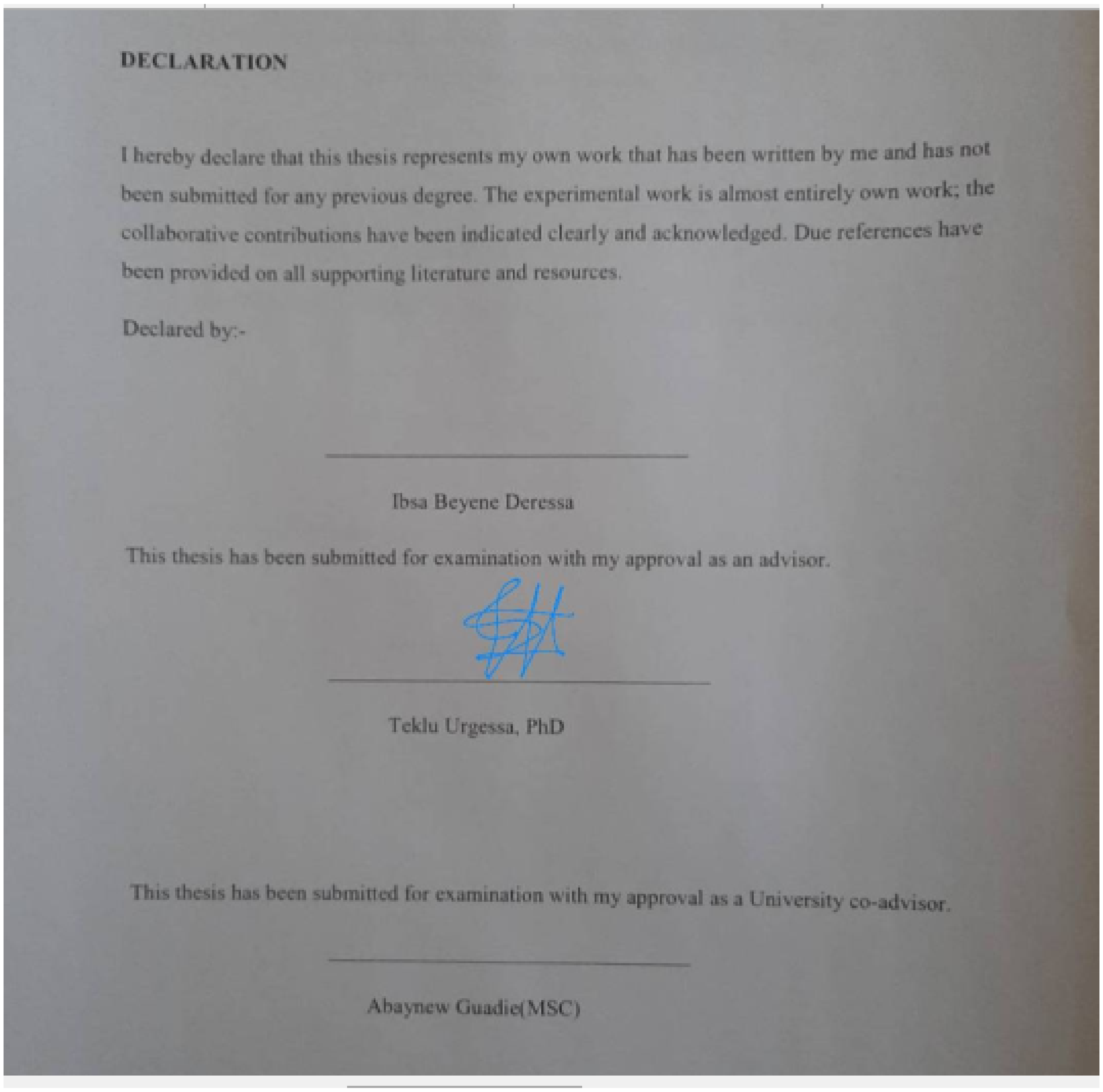
Sentence #	Word	POS	Tag
Sentence:699	Tajaajila	NNP	O
Sentence:699	abbaa	NNP	O
Sentence:699	seerummaa	NNP	O
Sentence:699	bara	NNP	B-MISC
Sentence:699	kana	IN	I-MISC
Sentence:699	kenname	VB	O
Sentence:699	irraa	IN	O
Sentence:699	birriin	NNP	B-MISC
Sentence:699	miiliyoona	NNP	I-MISC
Sentence:699	87.6	CD	I-MISC
Sentence:699	ta'u	IN	O
Sentence:699	mootummaadhaaf	NNP	B-ORG
Sentence:699	galii	NNP	O
Sentence:699	ta'eera	VB	O
Sentence:699	.	.	O
Sentence:700	Yakkoota	NNS	O
Sentence:700	harkaafi	NNP	O
Sentence:700	harkaa	NNP	O
Sentence:700	(	(	O
Sentence:700	RTD	NNP	O
Sentence:700	)	)	O
Sentence:700	galmee	NNP	O
Sentence:700	4,949	CD	B-MISC
Sentence:700	keessummaa'aniiru	VBD	O

Months:

Afaan Oromoo	English
Fulbaana	September
Onkoloolessa	October
Sadaasa	November
Mudde	December
Amajjii	January
Guraandaha	February
Bitootessa	March
Ebla	April
Caamsaa	May
Waxabajji	June
Adoolessa	July
Hagayya	August

6.7 *Appendix B: Some Numerals in Afaan Oromo*

Afaan Oromoo	English
Tokko	One
Lama	Two
Sadii	Three
Afur	Four
Shan	Five
Jaha	Six
Torba	Seven
Saddeet	Eight
Sagal	Nine
Kudhan	Ten
Digdama	Twenty
Soddoma	Thirty
Afurtama	Forty
Shantama	Fifty
Jaatama	Sixty
Torbaatama	Seventy
Saddeettama	Eighty
Sagaltama	Ninety
Dhibba	Hundred
Kuma	Thousand
Kuma Kudhan	Ten Thousand
Kuma Dhibba	Hundred Thousand
Miliyoona	Million
Biliyoona	Billion



DECLARATION

I hereby declare that this thesis represents my own work that has been written by me and has not been submitted for any previous degree. The experimental work is almost entirely own work; the collaborative contributions have been indicated clearly and acknowledged. Due references have been provided on all supporting literature and resources.

Declared by:-

____

Ibsa Beyene Deressa

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

[illegible]

____

Teklu Urgessa, PhD

This thesis has been submitted for examination with my approval as a University co-advisor.

____

Abaynew Guadie(MSC)