JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Afaan Oromo Text Chunking Using Conditional Random Fields and Support Vector machines

BY:

NEGESSA WAYESSA

THESIS SUBMITTED TO SCHOOL OF COMPUTING JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY

DECEMBER, 2019

JIMMA, ETHIOPIA

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

Afaan Oromo Text Chunking Using Conditional Random Fields and Support Vector Machines

BY:

NEGESSA WAYESSA

ADVISOR: GETACHEW MAMO (PhD)

CO-ADVISOR: ZERIHUN OLANA (MSc)

THESIS SUBMITTED TO SCHOOL OF COMPUTING JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY

DECEMBER, 2019

JIMMA, ETHIOPIA

APPROVAL SHEET

This independent research entitled as “Afaan Oromo Text chunking “has been read and approved as meeting the preliminary research requirement of school of computing in partial fulfillment for the award of the degree of masters in Information Technology.

Name	Signature	Date
1. Dr. Getachew Mamo(Advisor)	_____	_____
2. Mr. Zerihun Olana (Co-Advisor)	_____	_____
3. Dr. Kula Kekeba (External Examiner)	_____	_____
4. Mr. Teferi Kebebew (Internal Examiner)	_____	_____

DECLARATION

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

_____

Negessa Wayessa Juka

December 2019

DEDICATION

I dedicate this work to my dad Wayessa Juka Dabala and my mam Dammo Tolessa Fayisa

ACKNOWLEDGEMENTS

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

And also I would like to acknowledge my co-advisor Zerihun Olana for all his guidance at every step of the thesis work. Next, I would like to thanks Mr. Tefere Kebebew for his valuable assistance in providing his genuine and personal advice for reshaping my title during the title selection.

Besides, I would like to thank Mr. Dasale Tufa (MA in Afaan Oromo) for helping on data labeling and I would like to thank Mr.Sadik Abas for guiding on general knowledge of research and calling to me and asking my research status time to time. And also, I would like to thank Mr. Bayisa Busa and Miss Deivanai Guru, who contributed for this thesis work on correcting grammar and spelling of my research document. Last but not least, I would like to give my thanks to my classmates for their moral support, endless love and encouragement during my study.

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP- natural language processing

MEM- Maximum entropy model

HMM- Hidden Markov model

MBL- Memory based learning

SVM- Support vector machine

CRF- Conditional random field

POS- Part of speech tagging

IOB- Inside outside beginning

NP- Noun phrase

VP- Verb phrase

PP- Adpositional phrase

AdjP- Adjective phrase

AdvP- Adverb phrase

NN – common noun

PN –proper name

PPN –Personal pronoun

POP- Possessive Pronoun

VB-- Verb

JJ- Adjective

AD- Adverb

APC- Adpositions, pre-/postposition and conjunctions

DT- Determiner

CN- Counter number

PUNC- Punctuations

NEG - Negative word

ABSTRACT

Chunking is a natural language processing (NLP) application that is used to identify syntactically correlated parts of words in a text. It is the process of separates and segments the text into non-overlapping chunk. This separation and segmentation of text or sentence into chunk is recently considered as a major importance in NLP applications, especially in parsing, information retrieval, information extraction, named entity recognition and question answering system. The objective of this research was investigating the application of text chunking for Afaan Oromo language using Conditional random fields (CRFs) and Support vector machine (SVM). This was the first work to investigate chunking for Afaan Oromo language using these methods.

For experimentation, 725 sentences dataset (507 training and 218 testing) were used. During the experiment, seven different scenarios were considered based on the different combination of features such as word, part of speech tags of tokens and suffix. Since feature selection plays a crucial role in CRFs and SVM framework, experiments were carried out to find out most suitable features for Afaan Oromo tagging task. In the first scenario, these all features were considered. In the second scenario all features except suffix were considered. In the third scenario features except POS tag were considered. In the fourth scenario the features (two words from left and one word from right, POS tag, except suffix) were considered. In the fifth scenario the features (two words from left, POS tag, except any words from right and suffix) were considered. In the sixth scenario the features (one word from left and two words from right, POS tag, except suffix) were considered. In the seventh scenario the features (two words from right, POS tag, except any words from left and suffix) were considered.

The better result of CRFs and SVM methods were scenario-4 with (precision78%, recall 75%, F-score 76% and accuracy 84%); and (precision71%, recall 70%, F-score 70% and accuracy 80%) respectively achieved. This research indicated that a conditional random field (CRFs) was more applicable to Afaan Oromo text chunking than SVM.

Keywords: Afaan Oromo, Chunking, Shallow parsing, Conditional random fields, Support vector machine

CHAPTER ONE

INTRODUCTION

1.1 Background

Natural language processing (NLP) is a sub field of Artificial Intelligence which deals with the ability of computer systems to analyze and synthesize spoken and written languages as human beings. Natural language processing (NLP) is concerned with the progress of computational models of human language processing. It is an interdisciplinary research area at the border between linguistics and artificial intelligence aiming at developing computer programs by using natural language text or speech according to the linguistics rules [1]. NLP is making computers to perform useful tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech [2]. 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 [3].

The Natural language processing objectives is designing and building software/application that will analyze, understand, and generate human languages, so that eventually humans will be able to communicate with the machines using natural language like Afaan Oromo. Some of applications that make use of NLP to allow users interact with computer systems using natural languages are machine translation, information extraction and retrieval using natural language, text-to-speech synthesis, automatic written text recognition, grammar checking, named entity recognition, sentiment analysis, parsing, chunking (shallow parsing) and part-of-speech (POS) tagging.

The researchers of NLP use natural language technologies to develop computer applications for natural language. From this NLP investigation area or application, our work is mainly focuses on text chunking or shallow parsing.

Chunking or shallow parsing is a natural language processing application that attempts to provide process of dividing sentences into series of words that together constitute a grammatical unit (mostly either noun or verb, or postposition phrase). It is identifying groups of contiguous

words that depend on head of phrase in the sentences or texts. The term phrase is a group or combination of two or more words in the sentence.

Afaan Oromo is one of a major African language that is widely spoken in Ethiopia. The native speakers of the language are the Oromo people; they are the largest ethnic group in Ethiopia. It has around 40 million speakers, 34% of the total population of the country, native speakers and the most populous language of Ethiopia [9], [35], [36]. According to Tabor Wami [36] it is also the third most widely spoken language in Africa next to Arabic and Hausa languages. Specially, it is first widely spoken and used in most parts of Ethiopia and also some parts of other neighbor countries like Kenya, Tanzania, Djibouti, Sudan and Somalia [37]. We discussed further detail of Afaan Oromo language in chapter three. Though Afaan Oromo is one of the major languages in Ethiopia, and spoken by more than 34% of the total population of the country, research works on NLP like Text chunking are scarce. Thus, this research work is concerned with Afaan Oromo Text chunking.

Developing chunking or shallow parsing is challenging task especially for languages having low resource and complex linguistic structure like Afaan Oromo. Small amount of resources highly impacts the accuracy of the system. Complex structure of languages also needs design of important features and the best combination from these features. Afaan Oromo as one of low resourced and morphologically rich languages shares the above challenges in the designing NLP applications [13].

The natural language processing techniques to develop NLP applications like Text chunking can be broadly classified into rule based (linguistic approach), machine learning approaches and hybrid approach. Rule based approaches are based on handcrafted rules by language experts. Machine learning approaches are investigates how computers can learn (or improve their performance) based on data. A main research area is for computer programs to automatically learn to recognize complex patterns and make intelligent decisions based on data [19].

There are works that has been done under different languages for text chunking using different approaches like rule based, machine learning and hybrid. But adopting directly the approach or tool developed for different language is difficult for Afaan Oromo. Naturally, Natural languages have own unique grammar, syntax and feature of word. Previous some Afaan Oromo NLPs like information extraction, information retrieval, named entity recognition, POS tagging, sentiment analysis and anaphora were developed for solving the natural language processing area. These

applications were not identify syntactically correlated parts of words in a text. Especially, In Afaan Oromo language, some words comprises two words to represent single words are syntactically correlated parts of word. As example: ‘mana barumsaa’/School. This comprises words of Afaan Oromo is represent single word school in English. Therefore, applications used to identify syntactically correlated parts of words in a text were needed for Afaan Oromo language which was the objective of this research.

1.2 Statement of the problem

Chunking can be useful for developing parsing, information retrieval, information extraction, name entity recognition, machine translation, and question answering since a complete chunk (Noun, Verb, Adjective, Adverb or Postposition Phrase) is likely to be semantically relevant for the requested information [11].

The term parsing is a natural language processing technique that attempts to provide some understanding of the structure of a sentence into each word. Parsing is hierarchically categorized under syntax, and is used to generate valid parse tree for a sentence given grammar and lexical rules. Parsing uses recursive phrase structure grammars and arbitrary-depth trees. Parsing has problems with robustness, given the difficulty in getting broad coverage and in resolving ambiguity. It is also relatively inefficient: the time taken to parse a sentence grows with the cube of the length of the sentence, while the time taken to chunk a sentence only grows linearly. For these, use chunking text or shallow parsing is easier or more efficient [2]. Also, in the case of high ambiguity of the natural language to exact parsing of the text may become very complex. In these cases, chunking can be used as component to partially resolve these ambiguities. It is also widely used as an intermediate step to parsing for the purpose of improving the performance of the parser [7]. [12-13] designed Automatic parser for Afaan Oromo and they recommended developing chunking for Afaan Oromo is important to minimize time and effort.

Chunking the text is not only used for developing parsing; it can also be used for the development of information retrieval. In this application the chunking can be used to retrieve the data from the documents depending on the chunks rather than the words [6]. Chunking is not only used for developing parsing and information retrieval, but it is also used for generating each application, which was listed in the first paragraphs under statement problem topics. We discussed further detail of using chunk in the next titles under “Applications of result” subtopics.

Therefore, text chunker application which recognize the types of phrases in to noun phrase, adjectival phrase, adverbial phrase, Adpositional phrase and verb phrase are investigated. But, to the best knowledge of the researcher, text chunker has not been studied for Afaan Oromo yet. To this end, this study tries to answer the following research questions:

- What are the feature sets that will allow the CRFs and SVM to perform better result?
- From CRFs and SVM, which method fit text chunking system from Afaan Oromo text?

1.3 Objectives

1.3.1 General objective

The general objective of this study was to investigate Afaan Oromo Text chunking using conditional random fields and support vector machines

1.3.2 Specific objectives

The specific objectives of this research work were:-

- To prepare dataset for training and testing purposes.
- To identify feature set used for developing text chunking using CRFs and SVM.
- To build a prototype for text chunking from Afaan Oromo text.
- Evaluate the performance of the prototype model.
- To forward conclusion and recommendations based on result of this research.

1.4 Methodology

1.4.1 Literature review

To have conceptual understanding and identify the gap that is not covered by previous studies different materials, including journal articles, conference papers, books, and Jimma library repository have been reviewed.

1.4.2 Data collection and preparation

Afaan Oromo does not have publicly available annotated corpus text for NLP task like Text chunking. To investigate this research, we performed three tasks like: first, collected 725 sentences, then, POS tagged and lastly, chunk tagged.

1.4.3 Development tool

We used anaconda platform which includes a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems. For this study we used supervised machine learning algorithms of Scikit-learn and sklearn-crfsuite tool. From Scikit-learn, we applied Support vector machine algorithms (LinearSVC); and we applied sklearn-crfsuite (CRFs) algorithms.

1.4.4 Performance Analysis

Performance of the proposed system was analyzed using Precision, Recall, F-measure and accuracy test on various types of sufficiently large test samples. This analyzer metrics are the performance analysis from Scikit-learn and sklearn-crfsuite tool.

1.5 Scope and limitation of the study

This study was conducted to develop Afaan Oromo text chunking system. The prepared dataset was including both purpose based sentence (declarative, interrogative, imperative and exclamatory) and structure based sentences (simple, compound, complex and compound-complex) of Afaan Oromo. This dataset was manual annotated dataset. The case of used the manual dataset was no any annotated corpus of Afaan Oromo like Treebank and CoNLL2000 for English. The annotated dataset was consisted 725 sentences with 4689 tokens and 18 POS tag. Depends on resources like time and cost, our dataset was limit on small size which was 725 sentences.

1.6 Application of Results

Chunking is one of natural language processing application which useful in the different NLP applications. Some of NLP applications which use chunking are parsing, anaphora, information retrieval, question answering system, information extraction, named entity recognition, machine translation.

1.6.1 Parsing

Parsing is hierarchically categorized under syntax, and is used to generate valid parse tree for a sentence given grammar and lexical rules. Chunking or shallow parsing is the task of identifying and segmenting the text into syntactically correlated word groups and where parsing constructs

deeply nested structures for the result of chunking. It is considered as an intermediate step towards full parsing.

1.6.2 Information retrieval

Information retrieval is about returning the information that is relevant for a specific query or field of interest. In information retrieval, retrieving relevant data from the corpus or documents is depends on query. If this query is phrase level, the retrieved result is more relevant [7]. As an example: '*inni hoolaa adii bite'*. (He bought white sheep). [White sheep] or [*hoolaa adii*] is a phrase that is noun phrase. If we take this example to know about: 'what color of sheep bought by him?' For this question, the retrieve information by phrase query [white sheep] is rather relevant than the word query [sheep].

1.6.3 Anaphora

Anaphora means a word or phrase that refers to an entity that is mentioned previously and this word or phrase that it refers to is called its antecedent. Chunking is helpful for that antecedent that comprises two or more words [22]. For example, let us consider the name school in Afaan Oromo "mana barumsaa" which is composed of two words. if we take the output of POS tagger "mana " is tagged as NN and "Barumsaa" is also tagged as NN but both words follow each other and also belong to the same thing. For solve like this problem, chunking was need into [mana barumsaa]/NP.

1.6.4 Question answering system

A question answering is a task that aims to automatically give answers to questions described in natural language. It allows users to have exact answer rather than having list of potentially relevant documents. In the developing question answering system, query generation is a component of question answering system. In Afaan Oromo language, query is either single word or comprises two words. For two comprise words and syntactical correlated words, using chunk in the preprocessing was improve the quality of the system.

1.6.5 Information extraction

Information extraction is the way to extract information from a text in the form of text strings and processed text strings which are placed into slots labeled to indicate the kind of information that can fill them. To extract information, information extraction flow some procedure such as: first, the raw text of the document is split into sentences and each sentence is further subdivided into words which is tokenization. Next, each segmented sentence is tagged with part-of-speech

tags, which will prove very helpful in the next step, named entity detection. After named entity detected, transfer into relation detection, the sentence could chunked in the between named entity detection component and relation detection component.

1.6.6 Named Entity Recognition

Named entity recognition is a system which identifies the entity or the names of people, places, organizations, dates, amounts of money and other entity from text. The entity or the names of things represent single or more than one word in the text. Chunking is helpful for that name of thing comprises from more than one word.

1.6.7 Machine Translation

Machine translation is the translation of text from one human language into another human language by the use of a computer. In order to accomplish this it needs texts in one specific language as input and generates texts with a corresponding meaning in another language as output. The text or the words used as query were representing single or more than one word in the text. Chunking is helpful for that word comprises from more than one word.

1.7 Thesis Organization

The rest of this thesis is organized as follows. In chapter 2, literature review and related work is described. The chapter explains a different type of approaches to text chunking specifically it explains SVM and CRFs based approaches. Related works especially which is relevant to our work was discussed. Different types of chunk boundary identifications were also discussed in this chapter. In Chapter 3, Basic Afaan Oromo structure was discussed. Here we presented about the word class, phrases, clauses and sentences of Afaan Oromo. Chapter 4 presented Afaan Oromo Text chunking model using SVM and CRFs. Chapter 5 presented the experimental results of the proposed system along with its discussion. Finally, Chapter 6 concluded the thesis with the research findings and future works.

CHAPTER TWO

LITERATURE REVIEW

This chapter is concerned with the review of literature. The chapter contains four sections. The first section is explains basic concept in chunking. In the second section, different techniques to the task of sentence chunking are reviewed. The third sections of this chapter discuss related work to our study. For related works research papers and articles are selected based on the criteria related topic or the same topic with our study and the language we develop for. The goal of this chapter was for understanding area of the research, clarifying the approaches in developed chunking and related works that was done for others languages.

2.1 Basic Concept of Chunking

As discussed in the first chapter, identifying phrases in sentences is useful for exploring different NLP applications. Sentence is divided into the phrases based on head of phrase in that sentence. Human being who is expert of a language is easy to identify the phrase in the sentences. Using rule of expert, there is also NLP concepts to identify phrases in sentences. The process of determining or identifying phrases in sentence using NLP concepts is known as chunking (shallow parsing).

Chunking or shallow parsing is a natural language processing application that is the way to identify syntactically correlated parts of words in a sentence. Chunking is the process of dividing the sentence into chunks. Chunks are the non-overlapping structure in a sentence. Chunking separates and segments a sentence into its sub constituents, such as noun phrase, verb phrase, adverb phrase, adjective phrase, adverb phrase and Adpositional phrases. Shallow parsing or chunking is the alternative for full parsing in developing NLP applications [40].

A chunker finds contiguous, non-overlapping spans of related tokens and groups them together into chunks. Chunkers often operate on tagged texts, and use the tags to make chunking decisions. The term tagged texts or part of speech tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech. After part of speech tagging is identified for word, then chunker chunks the tagged text into non-overlapping group that are noun phrase, verb phrase, adverb phrase or the others phrase.

Chunk a text into correlated group of words or non-overlapping group is based on head of phrase in the sentence. Head of phrase means phrase which explain the content or the information about the sentence. For example: ***inni gara mana deeme***. (He went to home). From this example, we learn the direction he moved ([***gara mana]*** (‘**to home’**)). This phrase shows the direction he went. The phrases which show direction is called Adpositional phrase in Afaan Oromo. So, the head of phrase in sentences is Adpositional phrase that is [***gara mana].***

In Afaan Oromo language, there are five types of phrases such as noun phrase, verb phrase, Adpositional phrase, adverbial phrase and adjectival phrase. Based on this component, we proposed five type of chunk in Afaan Oromo. These are chunk of noun phrase, verb phrase, Adpositional phrase, adverbial phrase and adjectival phrase. The component of phrases in Afaan Oromo is further detail in the next chapter under subtopic of “Afaan Oromo phrase (‘Gaalee’)”.

2.2 Approaches for Chunking

There are different approaches to chunk the sentences or text of natural language. It can be categorized into rule based approaches, machine learning approaches and hybrid.

2.2.1 Rule based approach

Rule based approach is a set of static rules which is created by language experts. Rule based approach uses regular expressions rule to match text within given pattern. The advantages of rule based approach are extremely simple to implement and it does not require training corpus. However, there is a shortage of rule based techniques in developing natural language processing applications. Such as: Rules are not sufficiently complete; they cannot predict the new data and exceptions that occur naturally in language (work for specifically applied structure and not predict the others) [7]. The other drawback of rule base is on generating new rule. Generating new rule is complex because naturally the human language is complex. The percentage of accuracy in rule based approach is less than machine learning approach if the dataset has large size [15].

2.2.2 Machine learning approaches

Machine learning investigates how computers can learn (or improve their performance) based on given data. Machine learning approaches learn based on statistical models to make predictions for new data depending on learning the given data. Machine learning approaches are Supervised, Unsupervised and Semi-supervised.

Supervised learning approach is the type of machine learning which learn from labeled train data and predict the new data based on the trained dataset whereas; the unsupervised learning process is unsupervised since the input examples are not class labeled. Semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled examples when learning a model. In one approach, labeled examples are used to learn class models and unlabeled examples are used to refine the boundaries between classes. From this machine learning algorithms we focus on supervised machine learning algorithms that is more applicable to text chunking area especially for under resourced language like Afaan Oromo. From those machines learning, we focused on supervised machine learning which is more relevant to developing text chunking applications.

Some of the supervised machines learning approaches which are related to our investigation that explored for different language are: Hidden Markov model, Maximum entropy model, Naïve Bayesian, Memory-Based Learning, neural networks, support vector machine and Conditional random fields [4].

Hidden Markov model (HMM) is a model for a statistical process that generates sequential data. In HMM, there are observed events and hidden events in developing model. Observed event is the token which we input and hidden event is labeled part. Observation is observed based on the hidden event or variable. The trainer trains such models based on data, and / or use a trained model to try and infer the hidden states, or just get the probabilities of different observation sequences. The states or event processed in hidden Markov model are discrete number. The probability distribution of future states must depend only on the present state and be completely independent of past states. The states before the current state have no impact on the future except via the current state. For instance, if it predicts tomorrow weather you could examine today weather but you weren't allowed to look at yesterday weather.

Maximum entropy model (MEM) is also supervised machine learning algorithm used in sequential classification like HMMs. But, MEM solves the problem of multiple feature representation and long term dependency issue that occurs in HMM. It has increased the recall and greater precision than Hidden Markov Model [16].

Memory-Based Learning (MBL) is supervised learning approach that classifying data based on their similarity to data that they have seen earlier. Memory-based learning algorithm is constructs a classifier for a task by storing a set of examples [6].

A neural network is also an algorithm that designed for both supervised and unsupervised data. Neural network is process information in a similar way of human brain. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. Neural network is contains cyclic connections, which enable it to learn the temporal dynamics of sequential information. In the training process, neural network use most data to improve best accuracy of system [18].

Naïve Bayesian is one of the generative models which are the types of supervised machine learning. Naïve Bayesian is mostly used in text classification of natural language processing. It belongs to the group of graphical models which is used to model conditional independence between random variables. The Bayesian model is based on the use of Bayes law of probability which uses the inverse of conditional probability [39].

Next, we discussed briefly the supervised machine learning algorithms that we have used in our experiment which are support vector machine (SVM) and Conditional random fields (CRF).

2.2.2.1 Support vector machine (SVM)

A support vector machine (SVM) is a type of supervised machine learning classification algorithm. SVM was first introduced by Vladimir Vapkin and his colleagues [46]. According to [46] SVM are well-known for their good generalization performance and have been applied to many recognition problems, including chunking, named entity recognition, parsing and text categorization to mention some. [46] Also stated that there are theoretical and empirical results that indicate the good performance of SVMs' ability to generalize in a high dimensional feature space without over-fitting the training data than others distance or similarity based learning algorithms such as k-nearest neighbor (KNN) or decision tree.

SVM is also differed from the other classification algorithms in the way that it chooses the decision boundary that maximizes the distance from the nearest data points of all the classes. An SVM doesn't merely find a decision boundary; it finds the most optimal decision boundary. The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. The nearest points from the decision boundary that maximize the distance between the decision boundary and the points are called support vectors as seen in below figure. The decision boundary in case of support vector machines is called the maximum margin classifier, or the maximum margin hyper plane [47].

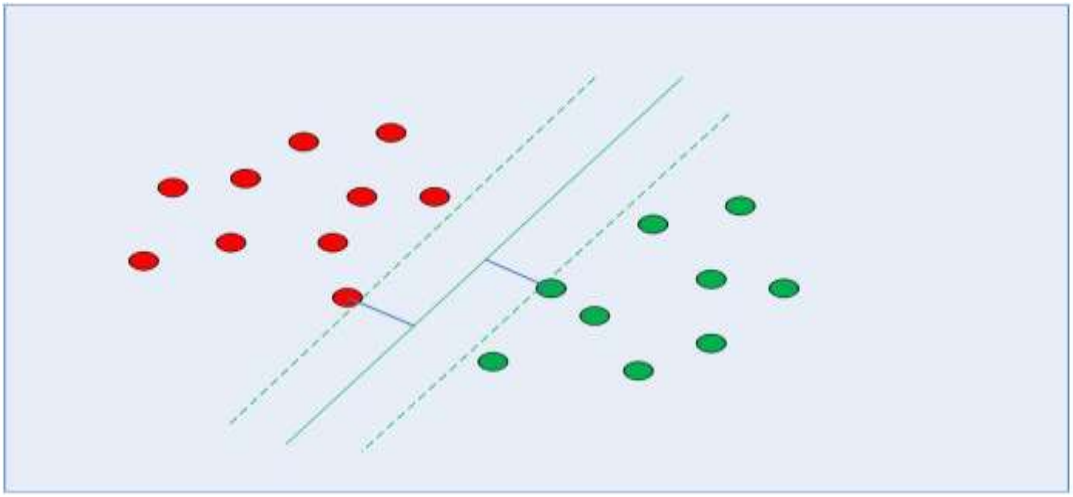

Figure 1: Decision boundaries with support vectors

Support vectors are the data points, which are closest to the hyper plane. These points will define the separating line better by calculating margins. Support vectors data points are more relevant to the construction of the classifier. The term hyper plane is a decision plane which separates between a set of objects having different class memberships. The others term is margin. Margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin [46].

In their basic form shown in Figure 2.1, SVM construct the hyper plane in input space that correctly separates the example data into two classes. This hyper plane can be used to make the prediction of class for unseen data. The hyper planes always exist for the linearly separable data [46]. The aim of hyper planes in the training phrase of the SVM is to plot the vectors in n-dimensional hyperspace and draw a hyper plane as evenly as possible to separate points from the two categories.

The formula description of support vector machine in separating points using hyper planes has described by [47]. As an example: in the classification when xi as a feature and yi is a label class either 1 or 0 (Whether yi is positive or negative indicates the category for the vector i). Assume, this hyper plane has normal vector w. then the hyper plane can be written as the points x satisfying:

$$w.x-b=0 \qquad (1)$$

Where $\frac{b}{||w||}$ is the offset of the hyper plane from the origin along w. This hyper plane is chosen so as to maximize the margin between the points representing the two categories. Imagine two hyper planes lying at the 'border' of two regions in each of which there are only points of either category. These two hyper planes are perpendicular to w and cut through the outermost training data points in their respective regions. Two such planes can be seen illustrated as dashed lines in above figure. Wanting to maximize the margin between the points representing the two categories is the same thing as wanting to keep these two hyper planes as far apart as possible The training data points which end up on the dashed lines in above figure are called support vectors, hence the name Support Vector Machine. The hyper planes can be described by the equations.

$$w.x-b=1 \qquad (2)$$

$$w.x-b=-1 \qquad (3)$$

The distance between the two is $\frac{2}{||w||}$. Since the SVM wants to maximize the margin, we need to minimize ||w||. It also does not want to extend the margin indefinitely, since it does not want training data points within the margin. Thus, the following constraints are added to the problem:

w.xi-b≥1 for xi in the first category, and

w.xi-b≥-1 for xi in the second category. This can be rewritten as the optimization problem of minimizing ||w|| subject to

(w.xi-b)yi≥1, (1≤i≤n)

If one replaces ||w|| with $\frac{||w||}{2}$, one can use Lagrange Multipliers to rewrite this optimization problem into the following quadratic optimization problem:

$$\underset{w,b}{min} \quad \underset{\alpha i \geq 0}{max} \left\{ \frac{||w||^2}{2} - \sum_{i=1}^{n} \alpha i \big((w.xi - b) yi - 1 \big) \right\}, \alpha i \geq 0 \qquad (4)$$

Where, the α_i is Lagrange multipliers. Data sets which are possible to divide in two are called linearly separable. Depending on how the data is arranged, this may not be possible. It is, however, possible to use an alternative model involving a soft margin. The soft margin model

allows for a minimal number of mislabeled examples. This is done by introducing slack variables i for each training data vector xi. The function to be minimized, ||w||2 /2 are modified by adding a term representing the slack variables. This can be done in several ways, but a common way is to introduce a linear function, so that the problem is to minimize:

$$\frac{||w||}{2} + C \sum_{i=1}^{n} \in i \qquad (5)$$

For some constant C, to this minimization, the following modified constraint is added:

$$(\mathbf{w}.\mathbf{x_i} - b)y_i \geq 1 \in i, (1 \leq i \leq n) \qquad (6)$$

By using Lagrange Multipliers as before, the problem can be rewritten as:

$$\min_{w,£,b} \max_{\alpha,\beta} \left\{ \frac{||w||^2}{2} + C \sum_{i=1}^{n} £i - \sum_{i=1}^{n} (\alpha i(yi\,(w.xi - b) - 1 + £i) - \beta i£i) \right\} \qquad (7)$$

For a, b >=0. To get rid of the slack variables, one can also rewrite this problem into its dual form:

$$\max_{\alpha} \left\{ \sum_{i=1}^{n} \alpha i - \frac{1}{2} \sum_{i,j}^{n} (\alpha i \alpha j yi yj \boldsymbol{xi}.\boldsymbol{xj} \right\} \qquad (8)$$

Subject to constraint

$0 \leq \alpha_i \leq C, (1 \leq i \leq n)$

And

$$\sum_{i=1}^{n} \alpha i yi = 0, (1 \leq i \leq n)$$

The above formal description on SVM applies only to those problems which are linearly separable. In a non-linear classifier, the input vectors xi are transformed as to lie in an infinitely dimensional Hilbert Space where it is always possible to linearly separate the two data categories.

Hence SVM is a binary classifier. The SVM described in the above formula explained by [47] is used for binary classification and which classify data in two categories. But Text chunking is a

multi-class classification problem since in natural language there are more than two label categories. For multi class problem, support vector machine can solve multi classification problem using different method. Some method in SVM that used to solve multi classification problem are: one-against-one and one-against-rest [48]. In the one-against-one approach, instead of trying to distinguish one class from all the others, they seek to distinguish one class from another one. In the one-against-rest binarization of problem, SVM is trained for each class in order to distinguish that class and the rest. When we used this both method of SVM, one-against-rest method are more suitable for practical than one-against-one. The one-against-all method has the added advantage of being already available in sklearn (LinearSVC library), so it should probably be your default choice.

2.2.2.2 Conditional Random Fields

Conditional random fields (CRF) is machine learning approach that uses discriminative learning for automatically learning the rules from the annotated training data and produces model which is applied to the test data specially for sequence classification [44]. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting structured data [31]. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences.

CRFs is probabilistic model for computing the probability (i.e $P\left(\underset{y}{\rightarrow} \middle| \underset{x}{\rightarrow}\right)$ of a possible output sequence $\underset{y}{\rightarrow}$ = (y1…..yn) $\sum$ y, given the input sequence $\underset{x}{\rightarrow}$ = (x1…..xn) $\sum$ x that is called observation. Y is the set of all possible output category sequence and x is the set of observation sequence. Under this section a special form of CRFs which is linear chain CRFs is discussed in detail that is used in our case. Linear chain CRFs is a special form of CRFs, which is structured as a linear chain that models the output variables as a sequence. Linear-chain CRFs can be formulated as [49]:

$$P\underset{\lambda}{\rightarrow}(\vec{y} \mid \vec{x}) = \frac{1}{z\overrightarrow{\lambda}(\vec{x})} . \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i \, f_i\left(y_{i-1}, y_j, \vec{x}, j\right)\right) \quad (9)$$

Where, x is a data sequence that is observation, y is a category sequence of data, n indicated the length of sentence, m indicated number of feature templates, λi represent the weights assigned to the different features in the training phase and $Z\underset{\lambda}{\rightarrow}(\vec{x})$ is a normalization factor that make the probability in the range [0,1], which can be expressed as [49]:

$$Z\underset{\lambda}{\rightarrow}(\vec{x}) = \sum_{\vec{y} \in y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{i-1,}, y_i, \vec{x}, j\right) \qquad (10)$$

The next sub sections will respectively discuss how a linear chain CRF model is used in the developing our model.

2.2.2.2.1 Feature extraction function

The basic part developing natural language processing application using machine learning is identifying a feature extraction. Identifying features is the most important process for machine-learning approaches because a feature design greatly affects the labeling accuracy [40]. The feature extraction is the components of CRFs. In our special case of Linear-chain CRFs, the general form of a feature extraction function is $\mathbf{f_i\,(y_{j-1},\, y_j,\, \underset{x}{\rightarrow},\, j)}$, which looks at a pair of adjacent states $\mathbf{y_{j-1},\, y_j,}$ the whole input sequence $\underset{x}{\rightarrow}$, and where we are in the sequence (j). These are arbitrary functions that produce a real value. As an example: we can define a simple feature function which produces binary values

$$\mathbf{fi}\left(\mathbf{yj-1, yj,} \underset{x}{\rightarrow}, \mathbf{j}\right) = \sum_{\mathbf{0}}^{\mathbf{1}}, \mathbf{yj} = \text{Noun } and\ xj = Jimma, otherwize\ 0$$

This is to mean, "If the y^{th} word is *jimma* and having a tag Noun, then $\mathbf{f_i}$ is one otherwise zero. The usage of this feature depends on its corresponding weight $\lambda 1$. If $\lambda 1$>0, whether f1 is active, it increases the probability of tag sequence $y_1{:}n(\underset{y}{\rightarrow})$. This is another way of saying the CRFs model should prefer the tag Noun for the word jimma. Otherwise $\lambda 1$<0, the CRF model will try to avoid the tag Noun for jimma. In this case the value of $\lambda 1$ will be learned from the corpus. [49] Addressed two problems of Linear chain CRFS:

Problem I: given observation x and CRF M: How to find the most probably fitting label sequence $\underset{y}{\rightarrow}$? This problem is the most common application of a conditional random field to find a label sequence for an observation.

Problem II: given label sequences Y and observation sequences X: How to find parameters of a CRF M to maximize P ($\underset{y}{\rightarrow} | \underset{x}{\rightarrow}$, M)? This problem is question of how to train to adjust the parameters of M which are especially the feature weights λi .

2.2.2.2.2 CRFs Model Training

For all types of CRFs, the maximum-likelihood method can be applied for parameter estimation. That means, training the model is done by maximizing the log-likelihood $\mathcal{L}$ on the training data $\mathcal{T}$:

$$\acute{L}(T) = \sum_{(\vec{x},\vec{y})\in T} \log P(\vec{y}\,|\vec{x}) = \sum_{(\vec{x},\vec{y})\in T} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right)}{\sum_{\vec{y'}\in y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)} \right) \right] \quad (11)$$

To avoid over fitting the likelihood is penalized with the term, $-\sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma^2}$

The parameter σ^2 models the trade-off between fitting exactly the observed feature frequencies and the squared norm of the weight vector. The smaller the values are, the smaller the weights are forced to be, so that the chance that few high weights dominate is reduced. For the purpose of avoiding over fitting, the likelihood function L($\mathcal{T}$) can be reorganized as:

$$\acute{L}(T) = \sum_{(\vec{x},\vec{y})\in T} \left[\log \left(\frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, \vec{x}, j)\right)}{\sum_{\vec{y'}\in y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y'_{j-1}, y'_j, \vec{x}, j\right)\right)} \right) \right] - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2} \quad (12)$$

$$= \sum_{(\vec{x},\vec{y})\in T} \left[\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{y-1,}\, y_j, \vec{x}\,, j) \right) - \log \left(\sum_{y'\in T} exp \left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y'_{j-1}, y'_j, \vec{x}\,, j) \right) \right) \right] - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

$$= \frac{\sum_{(\vec{x},\vec{y})\in T} \sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\left(y_{j-1}, y_j, \vec{x}, j\right)}{A} -$$

$$-\frac{\sum_{(\vec{x},\vec{y})\in T} log\frac{\left(\sum_{y'\in y} exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)\right)}{Z\overrightarrow{\lambda}(\vec{x})}}{B}-\frac{\sum_{i=1}^{m}\frac{\lambda_i^2}{2\sigma^2}}{C} \quad (13)$$

The partial derivations of $L(\mathcal{T})$ by the weights λ_k are computed separately for the parts A, B, and C. The derivation for part A is given by:

$$\frac{\partial}{\partial\lambda_k}\sum_{(\vec{x},\vec{y})\in T}\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1},y_j,\vec{x},j\right)=\sum_{(\vec{x},\vec{y})\in T}\sum_{j=1}^{n}f_k\left(y_{j-1},y_j,j\right) \quad (14)$$

The derivation for part B which corresponds to the normalization is given by:

$$\frac{\partial}{\partial\lambda_k}\sum_{(\vec{x},\vec{y})\in T} logZ_{\overrightarrow{\lambda}}(\vec{x})=\sum_{(\vec{x},\vec{y})\in T}\frac{1}{Z_{\overrightarrow{\lambda}(\vec{x})}}\frac{\partial Z_{\overrightarrow{\lambda}(\vec{x})}}{\partial\lambda_k}$$

$$=\sum_{(\vec{x},\vec{y})\in T}\frac{1}{Z_{\overrightarrow{\lambda}(\vec{x})}}\sum_{\overrightarrow{y'}\in y}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)$$

$$\sum_{j=1} f_i\left(y'_{j-1},y'_j,\vec{x},j\right)$$

$$=\sum_{(\vec{x},\vec{y})\in T}\sum_{\overrightarrow{y'}\in y}\frac{\frac{1}{Z_{\overrightarrow{\lambda}(\vec{x})}}exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y'_{j-1},y'_j,\vec{x},j\right)\right)}{P_{\overrightarrow{\lambda}}(\vec{y}|\vec{x})\ see\ equation\ (9)}$$

$$=\sum_{(\vec{x},\vec{y})\in T}\sum_{\overrightarrow{y'}\in y}P_{\overrightarrow{\lambda}}\left(\overrightarrow{y'}|\vec{x}\right)\sum_{j=1}^{n}f_k\left(y'_{j-1},y'_j,\vec{x},j\right) \quad (15)$$

Part C, the derivation of the penalty term, is given by:

$$\frac{\partial}{\partial\lambda_k}\left(-\sum_{i=1}^{m}\frac{2\lambda_k}{2\sigma^2}\right)=-\frac{2\lambda_k}{2\sigma^2}=-\frac{\lambda_k}{\sigma^2} \quad (16)$$

Equation 14, the derivation of part A, is the expected value under the empirical distribution of a feature f_i:

$$\check{\mathrm{E}}(f_i=\sum_{(\vec{x},\vec{y})\in T}\sum_{j=1}^{n}f_i\left(y_{j-1},y_j,\vec{x},j\right) \quad (17)$$

Accordingly, equation 15, the derivation of part B, is the expectation under the model distribution:

$$E(f_i) = \sum_{(\vec{x},\vec{y})\in T} \sum_{\overrightarrow{y'}\in y} P\overrightarrow{\lambda}\left(\overrightarrow{y'}\middle|\vec{x}\right) \sum_{j=1}^{n} f_i\left(y'_{j-1}, y'_{j}, \vec{x}, j\right) \quad (18)$$

The partial derivations of $\mathcal{L}(T)$ can also be interpreted as:

$$\frac{\partial_l(T)}{\partial \lambda_k} = \tilde{E}(f_k) - E(f_k) - \frac{\lambda_k}{\sigma^2} \quad (19)$$

To get the maximum by the approximation of the first derivation,

$$\tilde{E}(\boldsymbol{f_k}) - \boldsymbol{E}(\boldsymbol{f_k}) - \frac{\lambda_k}{\sigma^2} = \mathbf{0} \quad (20)$$

From this it is possible to calculate each weighting value of λk for each features f_k

Computing $\tilde{\mathbf{E}}(\mathbf{f_i})$ is easily done by counting how often each feature occurs in the training data. Computing $\mathbf{E}(\mathbf{f_i})$ directly is impractical because of the high number of possible tag sequences |Y|. In a CRF, sequences of output variables lead to enormous combinatorial complexity. Thus, a dynamic programming approach is applied, known as the Forward-Backward algorithm which is beyond the scope of this study to explain [49].

2.3 Related works

2.3.1 Text Chunking for English

A number of researchers have conducted research on English language text chunking. We present some of the successful systems that have been made on the text chunking of English language as follows:

Kuang-hua Chen and Hsin-His Chen [41] investigated chunking for English language using statistical approach. They used Susanne corpus which is a modified but shrunk version of Brown corpus is used as training dataset to train the system. They achieved 98% accuracy.

Chakraborty et al. [42] designed a rule based chunker for English language by framing handcrafted morphological rules. This chunker has been tested on 50 English text documents and they achieved 84% of accuracy. The complexity of rule based chunker is that the morphological or linguistic rules are language dependents and requires language experts.

Kudo and Matsumoto [17] investigated Chunking with Support Vector Machines for English language. To develop English text chunking using support vector machine, their first task is

identify feature set. Their feature set is includes: words, part-of-speech tags of context and combinations of part-of-speech tags to determine the chunk tag of the current word on the window size of context is 5 which means from left two words to right two words within current word itself. They also applied weighted voting of 8 SVM based system to achieve higher accuracy. They derived weighting strategy from theoretical basis if the SVM for the weighted voting systems. They have used three annotated corpora for their experiments. The base NP standard dataset and base NP large data set which consists sections (15-18) and sections (02-21) of WSJ part of the Penn Treebank for the training data and section 20 and section 00 for the testing data, which are used for the noun phrase identification, respectively. The chunking data set that is used for all types of phrase identification also consists of sections (15-18) of the WSJ part of Penn Treebank for the training data and section 20 for the test data. As they reported their approach achieves 94.15% precision and 94.29% recall for base NP-S data set, 95.62% precision and 95.93% recall for base NP-L data set and 93.89% precision and 93.92% recall and F-score 94.22 for the chunking dataset.

Feifei Zhai et al. [18] experiment on neural models for sequence chunking. They were used CONLL 200 dataset which contained 8,936 training and 893 testing sentences. 10% of the training data as discard for validation test. Lastly, they obtained 95.86% F-score.

2.3.2 Text Chunking for Hindi language

Sneha Asopa et al. [14] designed a rule based chunker for Hindi language. They developed model using rule based on 500 sentences. They obtained accuracy of model 74.16%. Lastly, they recommended the accuracy of machine learning is better than rule based approach to compare the results of [15].

Akshay Singh et al. [15] investigated a chunker for Hindi language using HMM approach. To design chunker for Hindi language, they were divided task into two subtasks: identifying the chunk boundaries and labeling the chunks with their syntactic categories. The first sub problem is takes a text in which words are tagged with part of speech tags as its input and marks the chunk boundaries in its output. Then, label chunk with syntax categories. They used 2,00,000 words dataset for developing the model and achieved 91.7% of accuracy.

Koeling [16] developed chunking model using MEM for Hindi language. They were used POS tagged dataset. To predict chunk tag, the algorithm check only three words from left and two

words from right side were being used. Their dataset contain 2,00,000 words. They achieved recall 91.86% and precision 92.08%.

2.3.3 Text Chunking for Chinese

Guang-Lu S. Et al. [39] Developed Chinese chunking based on Naive Bayes model and semantic features. The researchers used Chinese semantic dictionaries which called Hownet dictionary to extract the features. Hownet dictionary covers 67,440 words defined by 2112 different semantic which are regarded as a semantic category. Through the relation of words senses, they can get the most close word pair between each two words. Because each word perhaps has many definitions. In the feature extraction, researchers used the lexical and POS information of the five words that means include from current word left two words, right two words and current word itself within POS tag. The others a feature is suffixes which have two characters. Dataset of Chinese chunking was automatically extracted from the Chinese Penn Treebank which consisted of 3,822 sentences with 74,587 chunks and 92,729 word tokens. They used 90% of dataset of Chinese chunking for training and 10% for testing. They measured the performance of Chinese chunking in terms of the precision, recall, and F-score by applying different approach that are HMM, Naive Bayes model and Naive Bayes model with semantic features. They achieved Precision of 89.07, Recall of 90.82, F-score of 89.94 by applying HMM. They achieved Precision of 92.15, Recall of 90.90, F-score of 91.52 by applying Naive Bayes model. They achieved Precision of 93.03, Recall of 92.57, F-score of 92.80 by applying Naive Bayes model with semantic features. Naïve Bayes model with semantic features achieved the best results on the CPTB chunking data set.

Guang-Lu S. Et al.,[42] Developed Chinese Chunking Based on Maximum Entropy Markov Models. They used CPTB chunking data set and MSRA chunking data set which was dataset of Peking University corpus. The CPTB chunking data set consisted of 3,822 sentences with 74,587 chunks and 92,729 word tokens were 90% was for training and 10% for testing. MSRA chunking data set was also consisted of 18,239 sentences with 243,868 chunks and 473,179 word tokens. The vocabulary size was 34,793. Forty-two types of POS tags and forty-three types of chunk tags occurred in this the data set. For experiment, they adopted algorithms in CoNLL-2000.

For feature extraction process, they used lexical and POS information of the current word, the left context consisting of two words, and the right context consisting of two words are regarded as histories. In addition, they also used the affix information of the current word and the chunk tags of the previous word are atomic features. They also analyzed feature impact on chunking

result by the same data set as well as by the same algorithms. For the CPTB chunking data set, the achieved result MEMM POS features without lexical features was 88.35 F-score and MEMM Lexical and POS features 92.68. For the MSRA chunking data set, the achieved result MEMM POS features without lexical features was 85.38 F-score and MEMM Lexical and POS features 91.02. Chunking performance achieved by applying HMM to the CPTB data set was 89.94 F-score and MSRA data set was 88.53 F-score. For chunking boundary identification, they applied BIES annotated format which means B begins of chunk, I inside of chunk, E end of chunk and s single word of chunk.

For the purpose of comparing the performance of different models, they applied chunking models on both the CPTB chunking data set and MSRA chunking data set. The experiments on the CPTB data set showed that the new model achieved an F-score of 92.68%, which was better than the F-scores of HMM and MEMM in Chinese chunking. The experiments on the MSRA data set showed that the new model had an F-score of 91.02%, which was also better than the F-scores of HMM and MEMM. The reasons for the improvement have been analyzed through error analysis. They have also discussed the effects of different feature types and different sizes of training data sets on the performance of MEMM.

Fang Xu and Chengqing Zong [20] designed A Hybrid Approach to Chinese Base Noun Phrase Chunking. This hybrid is combination of Support Vector Machine (SVM) and Conditional Random Field model (CRF). Using the comparison between SVM and CRF they can check most of those errors then apply linguistic rules to minimize the error again. In this system, the results from CRF are better than that from SVM and the error-pruning performs the best. In the experiment, they used Yamcha and CRF++ tool to treat the testing data. They compared the original results from the two Chunkers, which used exactly the same format. They used conditional probability to detect the wrong IOB tags obtained and choose the most suitable output. They used Penn Chinese Treebank words which has size 13 MB, including about 500,000 Chinese words. Trained dataset was 300,000 Chinese words and the left was test dataset. The overall results achieved after comparison of SVM and CRF within grammar rule was 89.27% precision on the base NP chunking.

[43] Designed Chinese Base-Phrases Chunking. They have used a hybrid model to combine Memory-Based Learning (MBL) method and disambiguation proposal based on lexical information and grammar rules populated from a large corpus for 9 types of Chinese base

phrases chunking. This corpus contains 7606 sentences which are split into 6846 training sentences and 760 held out for testing. In the feature extraction process, they used feature window size (-2, +2). In the implementation, the used the software package provided by Tilburg University (TiMBL developed tool) within linguistic rules. They achieved accuracy (F-measure) of 93.4%.

2.3.4 Shallow Parser for Kannada Sentences

In 2017, [45] they were designed Shallow Parser for Kannada Sentences using Machine Learning Approach. From machine learning approach, they used conditional random fields. In the developing the chunker system for Kannada language, they divided dataset into training part and testing part. The training dataset was 6,000 (approximately 80,000 words) sentences have been taken from EMILLE (Enabling Minority Language Engineering) corpus and manually identified chunk boundaries and marked chunk labels for each word in the corpus. The second dataset was testing dataset which annotated (POS tagged) text. Those tested dataset were novels and stories category (from EMILLE corpus) dataset, containing 2,732 sentences (9,000 words) and 3,971 sentences (40,000 words) respectively. For feature extraction process, they used lexical and POS information of the current word, the left context consisting of two words, and the right context consisting of two words are regarded as histories. In addition, they also used the affix information of the current word and the chunk tags of the previous word are atomic features. For chunk boundary identification, they used BOI format which means B begin of chunk, I inside of chunk and O outside of any chunk. They achieved an accuracy of 92.77% and 93.28% is achieved on novels and stories dataset respectively.

2.3.5 Text Chunking for Amharic language

Abeba I. [19] implemented a hybrid approach to Amharic base phrase chunking and parsing. The hybrid approach of [19] was combination of HMM within error-pruning rule. They constructed hand crafted rule used to correct chunk phrases incorrectly chunked by the HMM. Feature set of this research POS tagging information of word. They organized 31 POS tags and 13 chunk tags to training chunking and parsing system. They were used IOB2 tag set to identify the chunked boundary. In the experiment, they used 320 sentence dataset which 288 training and 32 testing dataset and they used to evaluate 10-fold cross validation technique. This training dataset is contains feature set word, POS tag and chunk tag and testing dataset is containing word and POS tag. In this research first, HMM model apply on training dataset and testing dataset and lastly, rule based applied on output of HMM model to minimize the error of the result. They achieved

accuracy of 85.31% before applying the rule for error correction and an average accuracy of 93.75% after applying rules

2.3.6 Text chunking for Afaan Oromo

As far as the knowledge of the researchers concerns there were no system or work had been proposed in the past for Text chunking Afaan Oromo language. For, Afaan Oromo our thesis is the first for chunking Afaan Oromo text and it became the initial or starting point for other researcher who was interested on these areas. The summary of papers, methodology, dataset used, features, language and accuracy obtained are shown in the below table.

Papers	Methods	Dataset	Features	Language	Performance %
Kuang-hua Chen and Hsin-His Chen,1993 [41]	Statistical	Brown corpus	linguistic rules	English	Accuracy:98
Chakraborty, 2016 [42]	Rule-based	50 documents	linguistic rules	English	Accuracy :84
Kudo and Matsumoto, 2001 [17]	SVM	Penn Treebank	Lexical+ POS tag	English	F-score:94.22
Sneha Asopa et al. 2016 [14]	Rule based	500 sentences	linguistic rules	Hindi	Accuracy :74.16
Akshay Singh et al. 2009 [15]	HMM	2,00,000 words	lexical	Hindi	Accuracy :91.7
Koeling 2000 [16]	MEM	2,00,000 words	Lexical	Hindi	Recall:91.86 precision: 92.08
Guang-Lu S. Et al. 2010 [39]	Naive Bayes + semantic feature	Hownet dictionary 67,440 words	Lexical+ POS	Chinese	precision :93.03, Recall:92.57, F-score: 92.80
Guang-Lu S. Et al., 2016 [42]	MEMM	CPTB and MSRA	Lexical+ POS	Chinese	F:92.6 CPTB F:91.02 MSRA
Prathibha RJ, Padma MC, 2017 [45]	CRF	EMILLE (6,000 sentences)	Lexical+ POS	Kannada	Accuracy:92.77 novels and 93.28 in stories
Abeba I. 2013 [19]	HMM+ rule	320 sentence	Lexical + POS	Amharic	Accuracy: 93.75

Table 1: Summary of Related work

Text chunking systems were summarized by using different approach in the above table. The rule based approach is good result in very small data size and it is dependent on linguistic experts. Machine learning approach is better performance than rule based approach. However, in

Machine learning approaches, a pre-tagged or an annotated text is required to train the system. The accuracy of Machine learning chunker directly depends on the size of training dataset. As the size of training data increases, the accuracy also increases. The calculated result of machine learning is also different.

In the above section, we also surveyed on different approaches of chunking and relevant related work of our study. From discussed approaches, CRFs and SVM achieved better performance than others machine learning in the reviewed papers. Therefore, in this research we applied conditional random fields and SVM machine learning algorithms for Afaan Oromo Text chunking.

CHAPTER THREE

OVER VIEW OF AFAAN OROMO LANGUAGE

3.1 INTRODUCTION

Ethiopia is one of the multilingual countries which located in horn of Africa. It constitutes more than 83 ethnic groups [34] with diversified linguistic backgrounds. The country comprises the Afro-Asiatic super family (Cushitic, Semitic, Omotic and Nilotic). Afaan Oromo belongs to an East Cushitic language family of the Afro-Asiatic language super family. It is the most widely spoken language in Ethiopia. It has around 40 million speakers, 34% of the total population of the country, native speakers and the most populous language of Ethiopia [9], [35], [36]. According to Tabor Wami [36] it is also the third most widely spoken language in Africa next to Arabic and Hausa languages. Specially, it is widely spoken and used in most parts of Ethiopia and some parts of other neighbor countries like Kenya, Tanzania, Djibouti, Sudan and Somalia [37].

Currently, Afaan Oromo is an official language of the Oromia state (which is the largest regional state among the current Federal States in Ethiopia). With regard to the writing system, *Qubee* has been adopted and became an official script of Afaan Oromo since 1991 in Ethiopia. The writing system of Afaan Oromo language is straightforward which is designed based on the Latin script. Thus, letters in the English language are also in Oromo except the way it is written. The letters are combining together and give meaning full words in according to the language grammar. Then, words are sitting together by using spaces between each other's and produce others structure of language like text. Afaan Oromo text is written from left to right and spaces between words use as demarcation [9]. Naturally, natural languages are collection of lower and higher hierarchy in structures of text. The lower level structures combining with each other and build the top level structures. According to Addunyaaa Barkeessaa [24] hierarchy of Afaan Oromo structure are listed in the below format.

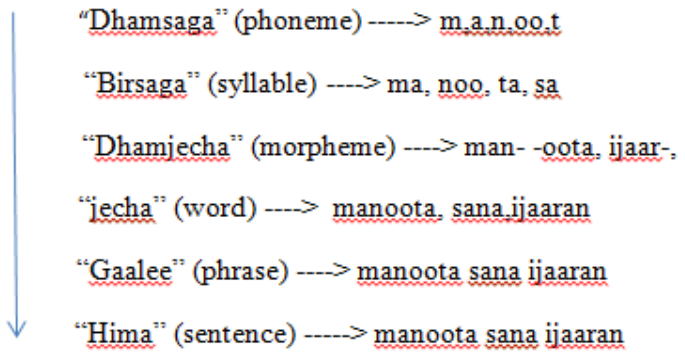

Figure 2: hierarchy of Afaan Oromo structure

In this research, we investigate text chunking of Afaan Oromo using a machine learning techniques. Text chunking is dividing sentences into non-overlapping phrases in such a way that syntactically related words are grouped in the same phrase. Hence, the phrases are a group of word that is appears in sentences structures. As an example: ***"Tolaan Hoolaa adii bite".*** [Tola bought white sheep.] This sentence is constructed from different words such as **Tolaan**, **Hoolaa**, **adii** and **bite** within punctuation mark which is period "**.**". From this sentences we extract two phrases depend on Afaan Oromo grammar such as [**Tolaan**] and [**hoolaa adii**]. The first phrase is construct from noun word "Tolaa" and suffix "-n". The second phrase is also constructed from two words that are head words and modifiers such as "hoolaa"/sheep (head words) and "adii". "adii"/white is an adjective which modifies the noun word sheep/'hoolaa'. According to Afaan Oromo grammar the modifications come after the modified word to describe the modified words [10] [24]. Here the word white is an adjective that comes for describe the word sheep. In Afaan Oromo language, when noun come before adjective and adjective come after a noun, they produce noun phrases.

The others classes in sentences of Afaan Oromo are punctuation marks which appear at the end of word order. Punctuation marks used in both Afaan Oromo and English languages are the same and used for the same purpose except apostrophe mark. Apostrophe mark (') in English shows possession, but in Afaan Oromo it is used in writing to represent a glitch sound known as ***hudhaa***. It plays an important role in Afaan Oromo reading and writing system. For example, it is used to write a word in which most of the time two vowels appear together like "***Sa'a***". The sentences of Afaan Oromo are ended by period or full stop, question mark and exclamation mark.

Since our work involves in text chunking or extraction of phrases in sentences into non-overlapping groups. This phrase is constructed from different word classes and part of sentences. The following sections of this chapter explore word classes, phrases, clause and sentences of Afaan Oromo.

3.2 Afaan Oromo Word Classes ('Kutaa jecha')

Word class is the basic part of natural language structure. It combines together and gives meaningful information for language. The combination of these words based on grammar of that language. The sentences of Afaan Oromo construct from word in the order of subject-object-verb (SOV). For example: *Kumsaan mana bite*. "Kumsa bought house." This sentence construct from three words which is subject (kumsaa), object (mana) and verb (bite). However, the words order in sentence of Afaan Oromo is not same as an English language. The word order of English language is subject-verb-object. In this sentence, the subject (kumsa), verb (bought) and object (house). Here, we understand the order of word classes in the sentence is depends on grammar of that language.

Afaan Oromo has different words based on their meanings and grammatical. In the old Afaan Oromo books, Afaan Oromo words classes are categorized into eight different types; namely Noun, Verb, Adjective, Adverb, pre-/postpositions, Pronoun, Conjunction and Interjection [12]. This categorization like English language criteria. However, Afaan Oromo word classes categorized into groups not based on English or other language criteria, categorized based on concept of Afaan Oromo is very important. Therefore, Afaan Oromo word classes categorized into five based on content and form of Afaan Oromo language such as noun, adjective, verb, adverb, pre-/postpositions [21] [26]. The following sections discussed the word classes of Afaan Oromo which were played in the construction of phrase structure.

3.2.1 Noun word class

Afaan Oromo nouns are words that used to identify things like place, people, animal and etc. it is also the word that represent person, place, material, abstract of something, etc. This noun word classes were categorized into five. Such as: proper noun, common noun, collective noun, material noun and abstract noun.

Proper noun, it is nouns which representing or calling one place or identify something form each other's. For example: Jimma, Ambo, Chala, and Kumsa. According to grammar of Afaan Oromo

(‘Seerluga Afaan Oromoo’) to show plural of noun, not add ***“-oota”*** suffix on the proper nouns like the others nouns. However, to show plural idea, we can add ***“-faa”*** word or suffix on some proper noun. As an example:

- Finfinnee + **oota**=Finfinnoota. (**Impossible**)
- Abdiisaan dhufeera (Abdisa has come.) (Single person, Abdisa)
- Abdiisaan**faa** dhufaniiru (Abdisa and his group come).

Common nouns of Afaan Oromo were representing something which has common behaviors. It refers to general, unspecific categories of entities. For example:Saree (dog), Nama (person), Hoolaa (sheep), Farda (horse), Laga (lake), Haadha (mother), etc.The common noun of Afaan Oromo can add suffix like (“-oota, -ota, -lee, -wwan, -oli, -een) to shows the plural noun. For example:

Saree ----- saroota (dog --- dogs)

Hoolaa ---- hoolota (sheep ---- sheep)

Farada - fardooota/fardeen (horse --- horses)

Sa’a--- sawwan (cow—cows)

Collective nouns are another part of nouns that refer to a group of something in particular. For example: Bineensa (wild animal), Uummata (people), Beeylada (domestic animal) and Horii (cattle)

Material nouns of Afaan Oromo are nouns that assign for different materials. For example: Barcuma/chair, Siree/bed, Damma/hone, Okkotee/jar, Dhadhaa/butter, etc.

Abstract nouns are nouns that focus on ideas, qualities or conditions like love, hate, power, and time. For example: Fayyaa/health, Hiyyummaa/poor, Bilisummaa/freedom, angoo/power and etc. Abstract noun of Afaan oromo are not add suffix to show plural. For example: Hiyyummaa -- Hiyyummoota (***impossible***)

According to Addunyaaa Barkeessaa [38] Noun word classes can be derived from non-noun or other noun words by affixing nominal affixes. Those derived nouns are created in three different ways. The first one is noun derived from adjectives. As an example: Bal’aa (adjective) + -ina = Bal’ina (noun). The second is noun derived from base noun. As an example: Guyyaa (noun) + -

saa= Guyyaasaa (noun). The third is personal pronouns which are included under noun classes. See personal pronouns of Afaan Oromo in the following table.

Person	Plural/singular	Case		Saalaa [gender]
		Matima [subject]	Anima [object]	
First	Singular	Ani [I]	Ana [me]	general
	Plural	Nuti [we]	Nuti [us]	
Second	Singular	Sii [you]	Sii [you]	General
		Sii [you]	Sii [you]	
Third	Singular	Inni [h]/ishee[she]	Isa [him]/isaan[they]	Male/female
	Plural	Isaan [they]	Isiin [their]	General

Table 2: Personal pronoun in Afaan Oromo

3.2.2 Adjective word class

An adjective is a word used to modify or describe a noun or pronoun. Adjectives of Afaan Oromo always come after nouns or pronouns which they modify. As an example: Tolaan mana **guddaa** bite. (Tola bought **big** home). The bold word (**guddaa**)/big modifies the noun come before it that 'mana'/home. Adjective word classes of Afaan Oromo are different types. Such as : adjective of quantity, adjective of cardinal numbers, possessive adjectives, interrogative adjectives, descriptive adjective, descriptive adjective of color and descriptive adjective of quality. As an example for the above types of adjectives:

- Tolaan qarshii ***shan*** qaba. (Tola has **five** birr.)—>(adjective of cardinal numbers)
- Kitaaba ***koon*** ergiseef. (I lend my book.)—> (possessive adjectives).

Afaan Oromo Adjective can show masculine and feminine. Masculine adjective are used with masculine noun, feminine adjective modify feminine nouns. All adjective can be made masculine or feminine by attaching the appropriate suffix. Masculine suffixes for adjectives are: -aa, -aawaa, -acha, and -eessa. Feminine suffixes are: -oo, -tuu, -ooftuu, and -eettii. Standard morphology rules apply when attaching suffixes [21].

Example:

Afaan Oromo		English Meaning
Masculine	**Feminine**	
Bareedaa	Bareedduu	Beautiful
Xinnaa/xiqqaa	Xinnoo/xiqqoo	Small
Gurraacha	Gurraattii	Black
Hiyyeessa	Hiyyeettii	Poor
mi'aawaa	mi'ooftuu	Sweet
Godeessa	Godeettii	Skinny

3.2.3 Verb word class

The verb is a word that describes the subject's action or state within a sentence. In the sentence structure of Afaan Oromo, Verb is comes at end of sentence. It is independent word in sentence like noun word class. Some verbs of Afaan Oromo are show masculine and feminine**.** Example:

Tolaan sanga diimaa guddaa bi***t***e. [Masculine]

Tolaashiin sanga diimaa guddaa bi***tt***e. [Feminine]

In all above sentences the word 'bi***t***e ', and 'bi***tt***e ' are verb class of Afaan Oromo and they are represent masculine and feminine respectively. The single character ***t*** is show the subject of this sentence is male name and the double character ***tt*** is show the subject this sentence is female name. Afaan Oromo word class is categorized into main verbs and auxiliary verbs. Some main verbs of Afaan Oromo are: Kottu, Deemi, dubbisi, jaallatte, jaallate and etc. Auxiliary verbs of Afaan Oromo are Jira, jirti, jiru, ture, qaba, qabda, qabna, qabdu, tahe, ta'e, ta'uu, barbaada and etc [25].

3.2.4 Adverb word class

Adverbs of Afaan Oromo are the word that are come before verb and describes or modifies a verb, an adjective or another adverb, but never a noun. Adverbs indicate manner, time, place, cause, or degree and answers questions such as _akkamitti' [how], _yoom' [when], _essaa'

[where]. The primitive adverbs are very few in number and some of them are: _haamayyu‘ [yet], _daddafii‘ [quickly], ‘garmalee’ [almost], ‘daran’[more], ‘akkamalee’[fast] , ‘baay’ee’[many] and etc. as an example in sentence:

Isheen **akkamalee** fiigdi. (She is running ***fast***)

In this example, the adverb ‘**akkamalee** [suddenly] precedes the verb ‘fiigdi [running] that it modifies. But, any word that comes before a verb is not necessarily an adverbs. Sometime main verb or the other word class comes before verb. As an example: “inni **deema jira**”. He is ongoing. Here **deema** and **jira** are a verb that comes repeatedly.

3.2.5 Adpositional word class

Adpositional is a word placed before and after noun and pronoun to form a phrase modifying another word in the sentence. The main properties of Adpositional are: they never use affixes and they don‘t assist to form other words. Some common prepositions and postpositions of Afaan Oromo are:

bira— [beside, with, around]

booda — [after]

cinaa‘ — [beside, near, next to]

‘dur, dura‘ — [before]

‘duuba‘ — [behind, back of]

‘itti' — [to, at, in]

gara — [towards]

‘eega‘, _erga‘ — [since, from, after]

haga‘, hanga‘ — [until]

‘waa'ee‘ — [about, in regard to]

As an example:

Tolaan ***gara*** jimmaa deeme –[Tola went ***to*** jimma]

hanga torban dhufu – [*until* next week]

Boonsaa***rraa*** – [**from** Bonsa]

Conjunction, pronoun and interjections are others word classes in others language. In Afaan Oromo language conjunction, pronoun and interjections word classes is not visible as word class by itself. Pronouns are used in place of noun and categorized under noun; this was discussed above noun word class topics. Interjections and conjunctions are used for joining one word with other, it is not necessary to assign different word class due to this reason the form or structures of conjunction and interjections are categorized under one of these because they are used to give functions for other words. When conjunction or interjections shows as a function of adpositional, they categorized under adpositional, etc. this implies they cannot form its phrase structure [26]. Therefore, In Afaan Oromo language the word class like noun word class, verb word class, adverb word class, adjective word class and adpositional word class are form to noun phrase, verb phrase, adverb phrase, adjective phrase and adpositional phrase respectively.

3.3 Afaan Oromo phrase (‘Gaalee’)

A phrase is a group of words that stand together as a single unit, typically as part of a clause or a sentence. This group word classes are stands as one part to explain one idea. The combined word classes in phrases are wide idea than each of word at different time. From group of words in phrases, one word class help has a head of phrases. Phrases are lack of subject-predicate organization of a clause. But, phrases play big role in the construction of sentences. As an example: inni ***mana guddaa*** bite. “He bought **big house**”. Here the phrase ***mana guddaa*** “**big house**” construct from the word ***mana*** and ***guddaa***. The type of ***mana*** is noun and ***guddaa*** is an adjective which come after noun house and describe the house. This big word is describing the types of house that means identify the size of house is big or small/middle. When we omitted the word big/guddaa from this sentence, the new sentence is also meaningful, but it cannot identify the type of house like the first sentence. The head of this phrase is noun word class that is house/mana word and the adjective (big) stands after noun for modification. Phrases are not transfer full message. For instance: ‘big house’. What is big house? It was constructed or bought or others? To transfer full message or meaning message, phrase is combining within the others phrases or word classes like: **inni mana guddaa bite**. “He bought big house.” This structure has meaning full message and knows as sentence level structure in a language.

In Afaan Oromo, phrases are categorized into five based on head of phrase. Such as noun phrase, verb phrase, adjective phrase, adverb phrase and adpositional phrase [10][5].

3.3. 1 Noun phrase (NP)

A noun phrase is a phrase which has a noun as head and others from word classes. Depend on Afaan Oromo grammar there are different rules to develop noun phrases from the same word classes and from different word classes. The following point discusses the rules how the noun phrases construct from different Afaan Oromo word classes [10][5].

- Construct noun phrase (NP) from noun and noun. Noun phrase is constructing from the same word classes noun and noun.

  Example:

  Nama/NN + Booranaa/NN= Nama Booranaa "Borana Man"

  Mana/NN + Jireenyaa/NN = Mana jireenyaa "living house"

  Aadaa + Oromoo+Booranaa= Aadaa Oromoo Booranaa "Culture of Borana Oromo"

- Construct noun phrase (NP) from noun and adjective. Noun phrase is constructing from different word classes noun and adjective.

  Example:

  Hoolaa/NN + adii/Adj = hoolaa adii "white sheep"

  Mana/NN + guddaa/Adj = mana guddaa "big home"

- Construct noun phrase (NP) from noun and pronoun. Noun phrase is constructing from different word classes noun and pronoun.

  Example:

  Mana/NN + isaanii/PN = mana isaanii "they home"

  Haadha/NN + ishii/PN = haadha ishii "her mother"

- Construct noun phrase (NP) from noun and demonstrative. Noun phrase is constructing from different word classes noun and demonstrative.

  Example:

  Mana/NN + sana/DT = Mana sana "that home"

  Farda/NN + kana/DT = Farda kana "that horse"

- Construct noun phrase (NP) from noun and numbers. Noun phrase is constructing from different word classes noun and numbers.

  Nama/NN + afur/CN = nama afur "four person"

  Barattoota/NN+ dhibaa/CN + shan/CN= Barattoota dhibaa shan "five hundred students"

In Afaan Oromo grammar, noun phrase is also construct from single proper noun word class that has suffix like '***n'*** and pronoun by itself it is a noun phrase [10]. As an example:

Abdiin mana guddaa ijaare. [Abdi build big house].

inni amma dhufte. [Currently, he is coming]

In the first sentences, "***Abdiin***" is constructing from personal name ***Abdii*** and suffix –***n***. therefore, according to Afaan Oromo grammar, there are two phrases in this sentence. [Abdiin/PNS]/NP and [mana/NN guddaa/Adj]/NP. In the second sentences, "***inni"/he'*** is personal pronoun that is gives by itself the message of phrase. Therefore, we can extract [inni/PPN] NP and [amma/AD dhufte/VB]/VP depend on grammar this language.

3.3.2 Adjective phrase (AdjP)

An adjective phrase is a phrase which has an adjective as head from others word classes. The following point discusses how the adjective phrases construct from different word classes.

- Construct Adjective phrase (AdjP) from adjective and adjective.
  Gurraacha/Adj + dheeraa/Adj= gurraacha dheeraa "black long"
  Magaala/Adj + xiqqaa/ Adj = magaala xiqqaa "brown small"
- Construct Adjective phrase (AdjP) from adjective and determines or two adjective and determines. Here determines Afaan Oromo are come after adjective.
  Example:
  Gurraacha/Adj sana/DT "this black"
  Magaala/Adj dheeraa/Adj kana/DT "this small handsome "
- Construct Adjective phrase (AjP) from adjective and numbers.
  Gurraacha/Adj + lama/CN= "two black"
  Adii/Adj+ sadii/CN = "three white"
- Construct adjective phrase from adjective plus numbers and determines. This determines come after the others.
  Example:
  Guggurraacha/Adj arfan/CN sunneen/DT = 'this four blacks'

3.3.3 Verb phrase (VP)

In the verb phrase of Afaan Oromo, the head of phrase is verb and the other word classes are composed under a head is called modification or description of a verb. Verb occurs at last of sentences of Afaan Oromo.

Example: Leensaan xalayaa ***barreessite***. "Lense ***write*** the letter." "***barreessite***" is a verb class of Afaan Oromo and present at the end of sentence. This sentence has two phrases noun phrase "**Lensaan**"/Lense's and verb phrase "xalayaa barreessite"/write letter. Here the head of word is writing and letter is the thing which the write action happens on. The following point also discusses how the verb phrases construct from different word classes.

- Construct verb phrase from adverb and verb. The adverb is come before verb to describe the verb phrase of Afaan Oromo. Example:

  Margaan[fiigichaan dhufe]. "Merga come running". 'Fiigichaan/running is the adverb that comes before head word (verb) 'fiigichaan/running to describe how this person was coming?
- Construct verb phrase from verb and verb.

  Example: Ishiin [barachuu qabdi]." She must learn". 'Barachuu/learn' is main verbs and 'qabdi/must' is auxiliary verbs in this sentences.
- Verb phrases of Afaan Oromo are also construct from adverb + auxiliary verbs + verb (head)

  Inni as turee deeme. [as/Adv turee/AU deeme/VB]
- Verb phrases of Afaan Oromo are construct noun + verb + verb

  Boonsaan Asallaa bulee gale. [Asallaa/NN bulee/VB gale/VB]
- Verb phrases of Afaan Oromo are construct objective + adverb + verb (head)

  Birrituun sa'a daddafte elmite. [Sa'a/NN daddafte/VB elmite/VB].
- Verb phrases of Afaan Oromo are construct objective + verb + verb (head)

  Leenci gaafarsa caccabsee nyaate. [gaafarsa/NN caccabsee/VB nyaate/VB].
- Verb phrases of Afaan Oromo are also construct adjective + verb (head).

  Adaamaan jimmaarra daran bal'aadha. [Daran/Adj bal'aadha/VB]

3.3.4 Adverb phrase (AdP)

Adverb phrase of Afaan Oromo construct from repetition of the same adverb word class and unit of different adverb structure. Therefore, the following rule is the way to construct adverb phrase in the Afaan Oromo.

- Adverb phrase construct from the same adverb.
  Ayyanni irreechaa [wagga waggaan] kabajama

Adverbs phrases are combined at begin of sentences, between subject and object and between object and verb. Example:

Darbee darbee bokkaan gammoojjii rooba.

Bokkaan ***darbee darbee*** gammoojjii rooba.

Bokkaan gammoojjii ***darbee darbee*** rooba.

- Adverb phrase construct from different adverb.
  Tolaan [***halkaanii guyyaa***] hojjeta.
- Adverb phrase construct from adverb of time + verb
  Caaltuun [***bara darbee***] heerumte.
- Adverb phrase construct from adjective+ Adpositional (‘-tti’). The adjectival that take postpreposition ‘-tti’ is adjective which show the time, not place or position.
  Nuti kitaaba caasima Afaan Oromoo [***dhiyoottii***] maxxansina.

3.3.5 Adpositional phrase (PP)

Adpositional phrase is a phrase whose head is Adpositional. Adpositional phrases are constructing from noun and Adpositional or Adpositional and noun. The Adpositional place is come before noun and may be come after noun in sentence of Afaan Oromo.

For example:

[Muka jala] “under tree”. Here the adpositional’ jala/under’ comes after noun ‘Muka/tree’.

Caaltuun [gara manaa] deemte. “ Chaltu went to home.” But, in this sentence the adpositional’gara/to’ come before the noun ‘mana/home’. The order of this adpositional and noun are depend on context of Afaan Oromo sentences.

In the above example, the notation we used was described as the following: NN: noun, VB: verb, AD: adverb, CN: cardinal number, PNS: Personal pronoun, Adj: adjective and etc.

3.4 Afaan Oromo Clause ('Ciroo Afaan Oromoo')

Clauses are a group of words or the combinations phrases that have subject and verb in their structure/constructions. So in order to say a clause there must be a minimum of one verb. In the phrase structure, phrases has not own subject and predict or verb. But, clauses have both subject and predicate. In addition, clauses are also different from sentences, sentences are transferring full message and clauses are may be transferring full message or not. Depend on the above criteria clauses of Afaan Oromo are categorized into dependent and independent clauses. Dependent clauses have subject-verb, but it not transfer full message. Independent clauses are transferring message like sentences or it is equal to sentences [5]. Example:

Barumsi **waan itti cimuuf**, "education is difficult to," …… (Dependent clause)

Caalaan ***barataa cimmaadha***. "Chala is diligent student" …..(Independent clause)

The first underlined word classes are shows as the phrases in dependent and the second underlined is phrase in independent clauses of Afaan Oromo.

3.5 Afaan Oromo Sentence ('Hima Afaan Oromoo')

Sentences are the structures that construct from collection of word classes and transfer meaning full message as well as at minimum it contain subject-predicate part. Subject means in Afaan Oromo "Matima' and predicate (kutima) is contain objective (Antima) and verb (gochima).

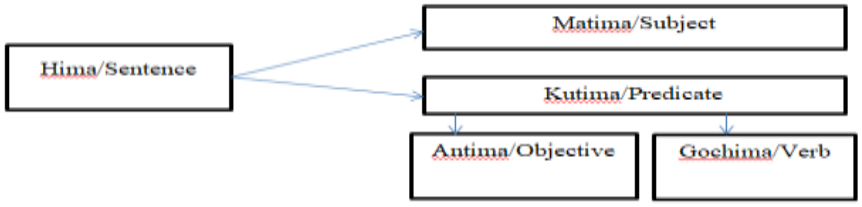

Figure 3: Afaan Oromo Sentence structure

"Inni manaa ijaare". He construct home. "inni" is subject," mana ijaare" is predicate (mana/objective and ijaare/verb).

Afaan Oromo Sentences are categorized into two based on grammatical structure. Such as: purpose based sentences and structural based sentences.

1. **Purpose based sentences:** Purpose or function based sentence of Afaan Oromo are categorized into declarative, interrogative, imperative and exclamatory sentences.
   a. **Interrogative sentence:** Interrogative sentences are sentences that can form a question. The question can be the one that ask the known thing to be sure or the one that asks the unknown one. It explains about question about the subject of sentence and task of verb. Example: Margaan yoom dhufe? ("When Merga come?")

   Afaan Oromo interrogative sentences consist interrogative pronouns which are maal "what", eenyu "who", maaliif "why", eessa "where", meeqa "how", kam "which", yoom "when"

   b. **Declarative sentence:** Declarative sentence is a type of sentence which used to describe happiness of something, changeable of something, the occurrence of something. Example: Barnoota barachuun gaarii dha. ("Learning education is good")

   This sentence is describing about learning education. If we learn education, it is good.

   c. **Imperative sentence:** Imperative sentence is a sentence which is used for pass instructions or command. Example: Yeroo barnoota kabaji. "Keep education time"
   d. **Exclamatory sentence:** Exclamatory sentence is a sentence which is used express happiness and worries. Example: dhifaama! ("Sorry!")
2. **Structural based sentence:** Afaan Oromo sentences are also categorized into simple, compound, complex and compound-complex sentences depend on grammar structure ('caasaa seerlugaa').
   a. **Simple sentence**

A simple sentence is a sentence with one main clause or it is a sentence which consist only a single verb in its structure no matter how many subjects are there. This simple sentence is transfer only one fully message. Simple sentence can construct from noun phrase and verb phrase. Example:

Tolaan gara mana kitaabaa deeme. ("Tola went to library")

Tolaa, Margaa fi Iftuun gara mana kitaabaa deeman. ("Tola, Merga and Iftu went to library")

Here in the first sentences there is only one verb '***deeme'***/went. The message of this sentence is only one. Tola went to library only. He was not gone to neither of library. The second sentence is also containing only one verb '**deeman'**/went (it is plural form of '***deeme'***). The subject of second sentence is more than one. Three of them went library.

b. Compound sentence

Compound sentence is a sentence that contains two or more than two verb and transfer two or more than two message. Compound sentence is also contains two or more than two independent clause. It construct from two or more than two simple sentences or independent clauses. Example:

Fayyisaan jimmaa deeme, Tolaan immoo Finfinnee deeme. "Fayissa went to jimma. But, Tola went to Finfine".

The first part of this sentence is 'Fayissa went to jimma'. It is simple sentence and it contains single verb 'deeme' as well as it transfers one message (Where Fayissa is go?). The left part is also independent clause or simple sentence like the first one.

Each main clause of a compound sentence has its own subject and predicate. These clauses usually combined by: coordinating conjunctions, semi-colon and adding markers like {-e} on verb to generate long sound. For example look the following three independent clauses:

"Namni gaaridha"(the man is good);

"Namni mana ijaare gaaridha"(the man who built house is good);

"Namni barumsa barate" (the man learned an education)

These are simple sentences, when we combine them to one another they form the following compound sentence:"Namni mana ijaareefi namni barumsa barate gaariidha"(the man who built house and learned an education is good)

c. Complex sentence

A complex sentence is an independent clause joined by one or more dependent clauses. A dependent clause either lacks a subject or a verb or has both a subject and a verb that does not express a complete thought. In the complex sentence, independent clause was concluded dependent clauses ideas depend on subordinator on the dependent clauses. Example:

Yoo Finfinnee deemteef, meeshaa naa bita. “If you went to Finfine, you bought material for me.” This sentence is formed from independent clause and dependent clause. The dependent clause contains subordinator (yoo/if) and independent part is ‘meeshaa naa bitta’.

d. Compound complex sentence

A compound-complex sentence has one and more independent clauses and two and more dependent clause. Example:

“Yommu dhaqes, yommuu gales, natti goree na gaafatee darbe”(he asked me when he had gone and come). In this sentence there are two dependent clauses and one independent clause. “Yommu dhaqes” and “yommuu gales” are dependent clause. “natti goree na gaafatee darbe” is independent clause.

As general, in this chapter we discussed over view of Afaan Oromo structure in term of related to our case. Specially, we discussed the word classes of Afaan Oromo which are helps to construct phrases. Also we explained about phrases as well as clause and sentences which are containing phrases as parts. In the next chapter we discussed the way to prepared dataset by using this Afaan Oromo structure which was discussed in this chapter.

CHAPTER FOUR

AFAAN OROMO TEXT CHUNKING MODEL

4.1 Introduction

Based on the description of the previous chapters, Text chunking involves the identification and divides sentences into correlated group of words such as noun phrases, verb phrases, adjective phrases, adverb phrases and adpositional phrases. Moreover, in the taxonomy of computational linguistics, Text chunking falls within the category of sequential labeling or segmenting like POS tagging, word segmentation, Named entity recognition and information extraction. The machine learning models which have sequence modeling framework classification introduced to the field of natural language processing (NLP) are CRFs, SVM, HMM, MEMM, naves Bayes and decision tree. Conditional random fields (CRFs) are a statistical sequence modeling framework first introduced to the field of natural language processing (NLP) to overcome weakness of HMM and MEMM. CRFs have all the advantages of HMM and MEMMs without limitation of HMM and MEMMs. SVM is also the best machine learning algorithms that solve the natural language processing application like named entity recognition, information extraction, part of speech tagging, text chunking and others NLPs application.

In this chapter, the design and implementation of CRFs and SVM based Text chunking system for Afaan Oromo was discussed. This chapter was organized as follows: the first section talks about data collection and preparation. Second section describes over view of the chunking boundary identification; third section was architecture of Afaan Oromo text chunking.

4.2 Data collection and Data preparation

For this research, we prepared sentences manually by reading grammar book of Afaan Oromo and the others sentences were prepared by Afaan Oromo teachers in Bule Hora University (BHU) and Afaan Oromo instructors in ABFM academy of Jimma. In addition to expert of language, the dataset that prepared by researchers were also checked by Afaan Oromo Teachers in BHU.

This prepared includes both purpose based sentences and structural based sentences of Afaan Oromo. In the Afaan Oromo language, the structure and content of declarative and imperative is

not differing from others structural based sentences of Afaan Oromo. It only identified when we read those sentences. The sentences like interrogative and exclamatory sentences of purpose based are identified by punctuation mark from others. The left sentences like simple, compound, complex and double complex have criteria to count the statistics of each in dataset. In the reality of Afaan Oromo grammar, the simple sentences have single verb and transfer only single message. The compound sentences are including two or more than two verbs in sentences. The complex sentences contains single independent clause and one or more than one dependent clauses. The compound complex sentences are also including one or more independent clauses and two or more dependent clauses. Totally, the prepared sentences in this research were 725 sentences which include both purpose based sentences and structural based sentences of Afaan Oromo language.

The sentences that were collected by researcher and data collectors were not tagged. After collected data, the next steps of researcher were prepared of dataset in the form machine learning can learn from. This preparation dataset was assigning part of speech tagging for each tokens and identifying phrase boundary in sentences.

Part of Speech Tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context i.e., its relationship with adjacent and related words in a phrase or sentence. In sentences, all words can be labeled with their Part-of-Speech tag. These tags denote the grammatical function of the word in the sentence. Some simple, but well-known part of speech tags are for instance nouns, verbs, adjectives, adverbs and determiners. Part-of-Speech tagging makes sentences easier to parse or chunking by a computer, and is therefore a preprocessing step frequently used in text-processing systems.

[25], [32] and [33] designed part of speech tagging (POS) for Afaan Oromo language in different time and by using different approaches. However, we could not get the tagged dataset from those researchers to use in this research. Due to un-availability of POS tagged dataset from those researchers, for our work we prepared POS tagged dataset manually. In the process of annotated dataset, we used English tag set [2]. Totally, tag set used in this research were 18.

No	Tag	Description	Example
1	NN	Common noun	Nama, Muka, jimmaa
2	NNP	A tag for all types of plural nouns that are not joined with other categories in sentence (not show location and direction)	Namoota,biyyoota, Namni,sangichi,
3	NNPS	Common noun plus pre/postposition (which is show location and direction)	Inni **Booranattii** gale.
4	PN	proper nouns	Dammee,Caaltuu, Oromiyaa
5	PNS	Proper nouns that are joined with other postposition in sentence (not show location and direction)	Tolaan, caaltuun
6	PPN	Personal pronoun	Ana/ani, nuyi/nuti, sii/ati, isin, ishee
7	POP	Possessive Pronoun	Koo, kee, keenyaa
8	VB	Verbs	Kottu, Deemi, dubbisi, qaba
9	JJ	Adjectives	Bareedduu,diimaa, magaala,
10	AD	Adverb	Kaleessa,edana, yoomiyyuu,sirritti
11	APC	Adpositions, pre-/postposition and conjunctions	Ni, irraa, garuu, fi, akka,yoom,ykn, bira, gara,
12	DT	Determiners	Kun, kana, kanneen, sana
13	CN	Cardinal numbers.	Lama, kudhan, 2012
14	PUNC	Punctuations	. ? ! ‘ : [{ ( ) =-__
15	NEG	Negative word	Hin,miti
16	PPNS	Personal pronoun + postposition	Nurraa
17	ADPP	Adverb plus postposition “-ttii”	Tulluun qawwee dhukaasee [*fagootti*] bineensa ajjeese
18	PNPP	Proper name + postposition (which is show location and direction)	Abdiirra

Table 3: POS tag set used in this research

4.3 Chunk boundary format in Afaan Oromo

Afaan Oromo language phrases are categorized into five based on context of language structure such as noun phrase, adjective phrase, verb phrase, adpositional phrase and adverb phrase. Based on this, we designed five type of text chunking for Afaan Oromo like chunking of noun,

adjective, verb, adpositional and adverb. Outlining of handcrafted linguistic rules is not a trivial task. However, we have manually outline almost all linguistic rules and used as reference to identify the boundaries of chunker. We have prepared at 30 linguistic rules that are used to identify the boundaries of chunker for Afaan Oromo language in this research and they are listed below.

Noun chunking is a chunk of correlated group of words and noun is the head this chunk. We have got at 12 linguistic rules that are used to identify the boundaries of noun chunker for Afaan Oromo language. Such as:

1. Noun and noun
2. Noun and adjective
3. Noun and Possessive Pronoun
4. Noun and number
5. Noun and determiners
6. Noun, adjective and determiners
7. Proper nouns with (‘-n’)
8. Personal pronoun
9. Noun, Possessive Pronoun and determiners
10. Noun, adjective and number
11. Noun and noun phrase
12. Noun phrase and determiners

Verb chunking is the verb group includes the main verb and auxiliary verbs. We have prepared 5 linguistic rules that used to identify the boundaries of verb chunk. Such as:

1. Adverb and verb
2. Verb and verb
3. Adverb, verb and verb
4. Noun, verb and verb
5. Noun, adverb and verb

Adjective Chunk: Adjective chunk consists of all adjectives including predicative together with noun chunk. However, adjectives appearing before a noun will be grouped together with the

noun chunk. Therefore, we designed 4 linguistic rules that used to identify the boundaries of adjective chunk. Such as:

1. Adjective and adjective
2. Adjective and determiners or adjective, adjective and determiners
3. Adjective and number
4. Adjective, number and determiners

adpositional chunk is chunking a group of word when pre-/postposition acts as head. We prepared 5 linguistic rules that used to identify the boundaries of postposition chunk. Such as:

1. Noun and adpositions, pre-/postposition and conjunctions
2. Noun phrase and adpositions, pre-/postposition and conjunctions
3. adpositions, pre-/postposition and conjunctions and noun
4. adpositions, pre-/postposition and conjunctions and noun phrase
5. adpositions, pre-/postposition and conjunctions and postposition phrase or postposition and adpositions, pre-/postposition and conjunctions

Adverb chunk: a chunk that includes all adverbial phrases and adverb act as a head. We organized 4 linguistic rules that used to identify the boundaries of adverb chunk. Such as:

1. adverb and adverb
2. from different adverb
3. adverb with (‘-tti’)
4. adjective and postposition with (‘-tti’)

Over all, for this research, we prepared 30 linguistic rules to determine the chunk boundaries in Afaan Oromo sentences and we used when we prepared dataset.

After the phrase boundary length was identified, the next step was labeling the chunk. The POS tag set helps to assign the label on chunk. We used the IOB notation model in which every word is to be tagged. Every chunk included each of this notation models, for instance noun chunk contains B-NP and I-NP and others are also like this. The number of IOB label which we used in this designed were 11 that including O notation which is outside of any chunk. As an example: “**Dallaan loonii kaleessa ijaarame.”** This is Afaan Oromo simple sentence which was annotated in the table form for training.

Tokens	POS	Chunk_tag
Dallaan	NNP	B-NP
Loonii	NNP	I-NP
Kaleessa	AD	B-VP
Ijaarame	VB	I-VP
.	PUNC	O

Table 4: Sample chunk annotated Afaan Oromo sentences for training model

4.4 Architecture of the system

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

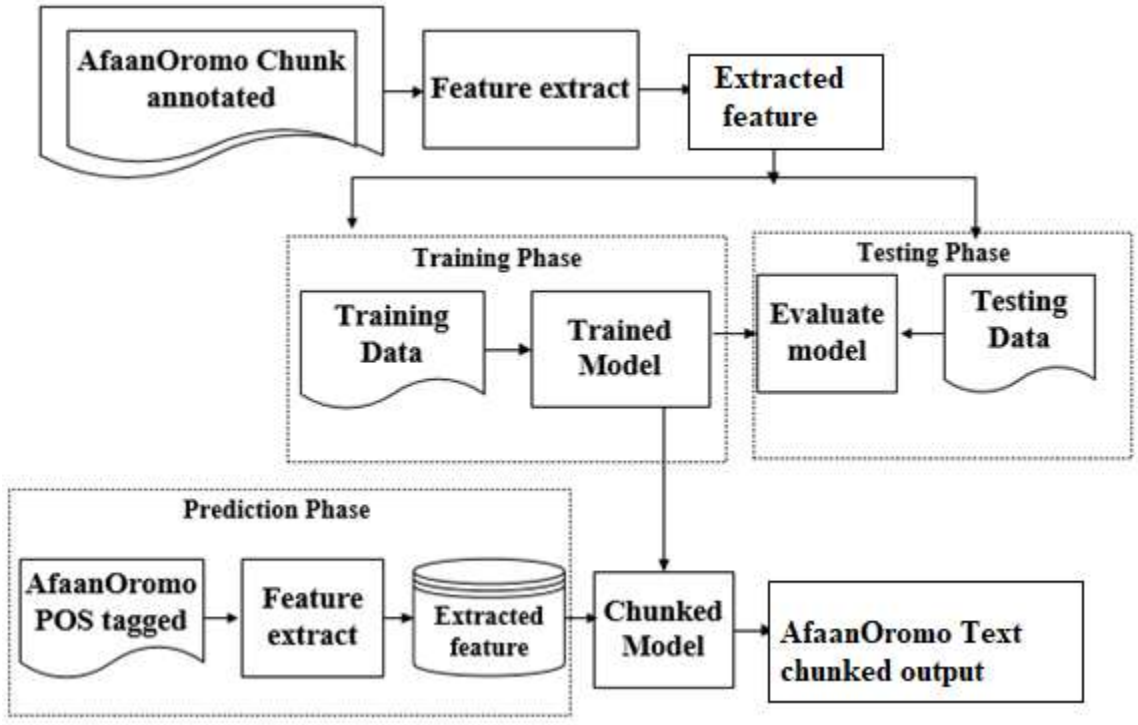

Figure 4: Proposed Architecture for Afaan Oromo Text chunking

4.4.1 Afaan Oromo chunk annotated

Afaan Oromo chunk annotated text was prepared in the form of three excel columns that means the first columns is tokens, second columns is POS tag and third columns is chunk tag or labels classes.

4.4.2 Feature Extraction

The basic part developing natural language processing application using machine learning is identifying a feature set. Identifying features is the most important process for machine-learning approaches because a feature design greatly affects the labeling accuracy [40]. Features indicate which information is used from the annotated corpus during the training stage.

In this research we chose to use the word itself with POS tag annotated to build our features. These features express the characteristic of the word at position current word by using information from the context of words. Due to POS tag of previous words and next words influence the chunk tag of current word we used the context window size is start from previous to previous word until next to next word of current words. We also used suffixes has feature sets. The term suffix is a sequence of last few characters of a word. The suffix information is helpful to track text chunking because words in the text chunk share some common suffix. The training features window size used in the proposed chunker were summarized in the below table.

Feature	Definition
Wi-2	Previous to previous word
Wi-1	Previous word
Wi	Current word
Wi+1	Next word
Wi+2	Next to next word
Pi-2	POS of previous to previous word
Pi-1	POS of Previous word
POSi	POS of current word
POSi+1	POS of next word
POSi+2	POS of next to next word
Si-2/1	Suffixes (-2/-1) characters lengths

Table 5: Feature extraction window size in the proposed system

After feature was extracted, we have used classifies extracted feature into 70% train and the left was for test sets.

4.4.3 Training phase

The training dataset in Machine Learning is the dataset that is used to train the model for performing various actions. This is training dataset is the ongoing development process models

learn with various algorithm to train the machine to work automatically. Shortly, the sample of data used to fit the model is known as training dataset. This training phase is for building the model. For this research, we used the conditional random field (CRF)(sklern-crfsuite) and SVM (LinearSVC) library to develop trained model. Trained model was output which we get after fitting training target of x with training target of y when target of x is the feature which contains words with POS and target of y is label part that is chunk classes.

4.4.4 Testing phase

The dataset which is not in training dataset that is used to evaluate the training model is called testing dataset. In our case, after trained model was stored, we used stored trained model to evaluate the testing data.

4.4.5 Prediction phase

Prediction phase is the process to input new data in the form of training feature and display the categorized data into labels classes. In the text chunking the input data is a text which tagged with part of speech. After trained model was stored and testing feature was evaluated, we can predict new inputted using this model. Prediction in this model is predicted when the inputted data was pass in the trained model.

Sample input for chunker model and Output obtained from the developed chunker model is given below:

Input: [('Tolaan', 'PNS'), ('gara', 'APC'), ('manaa', 'NN'), ('deemu', 'VB'), ('qaba', 'VB'), ('.', 'PUNC')]

Predicted Output: [(('Tolaan', 'PNS'), 'B-NP'), (('gara', 'APC'), 'B-PP'), (('manaa', 'NN'), 'I-PP'), (('deemu', 'VB'), 'B-VP'), (('qaba', 'VB'), 'I-VP')]

CHAPTER FIVE

EXPERIMENTS AND RESULTS

In this chapter we discussed development tools in this research and experimental results of text chunking using SVM and CRFs.

5.1 Development tools

The development tools used in the process of this research are anaconda platform for python programing and jupyter notebook (for editor interface), Scikit learn and sklearn-crfsuite machine learning library.

Anaconda is data science and machine learning platform for the Python and R programming languages. It is designed to make the process of creating and distributing projects simple, stable and reproducible across systems and is available on Linux and Windows. For this experiment, we used anaconda which is a Python based platform that contain major data science packages including pandas, scikit-learn and NumPy. Pandas are a software library written for the Python programming language for data manipulation and analysis. Pandas Data Frames make manipulating your data easy, from selecting or replacing columns and indices to reshaping your data. The other library is NumPy which is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices).

Scikit-learn is a Python machine learning library which includes a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems. It focuses on bringing machine learning to non-specialists by providing a general-purpose high-level language. Scikit learn is a machine learning library for the Python programming language.

For this research, we used support vector machine learning algorithms from sklearn tool. Support vector machine (SVM) has many classes library that is capable of performing binary classification and multi classification. Some of them are: SVC, NuSVC and linearSVC library. SVC and NuSVC are similar methods and use kernel parameters. On the other hand, linearSVC is another implementation of Support Vector Classification for the case of a linear kernel, but linearSVC does not accept keyword kernel.

Text chunking has multi-classification concept that means text chunking category are more than two categories, in our case there are 11 categories. In the multi classification, SVC and NuSVC implement the "one-against-one" approach for multi-class classification. If K is the number of classes, then K(K-1)/2 classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision function shape option allows transforming the results of the "one-against-one" classifiers to a decision function of shape (n_sample, K).

LinearSVC automatically uses the one-against-all strategy by default. You can also specify it explicitly by setting the multi_class parameter to ovr (one-vs-the-rest). Theoretically, the term 'One-against-one 'is an approach used to classify one class from others classes. In this approach, instead of trying to distinguish one class from all the others, we seek to distinguish one class from another one. As a result, we train one classifier per pair of classes, which leads to K(K-1)/2 classifiers for K classes. Each classifier is trained on a subset of the data and produces its own decision boundary. For this experiment, we used LinearSVC library of SVM which contains default approach for multi-class classification used in sklearn.

Sklearn-crfsuite is a thin conditional random field suite or it is python-crfsuite wrapper which provides interface similar to scikit-learn in Anaconda platform by python programing. Sklearn-crfsuite is a library which has implementation of CRFs for labeling and segmenting sequence data.

5.2 Performance evaluation

Scikit learn and sklearn-crfsuite machine learning has standard evaluation metrics by itself. From evaluation metrics, we used classification report measurements of performance system. A Classification report is used to measure the quality of predictions from a classification algorithm by counting how many predictions are True and how many are False.

The report shows the main classification metrics precision, recall, f1-score and accuracy on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes.

$$Precision = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{F1 Score} = 2 * (\frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}) \qquad \text{accuracy} = \frac{TP+TN}{all}$$

Where,

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

5.3 Experimental Results

Our experiment was conducted by used Conditional random fields and Support vector machine.

Sklearn-crfsuite is implementation of CRFs for labeling and segmenting sequence data. LinearSVM is also implementation of Support vector machine especially help for binary classification and multi classification. Here, in our case, SVM is used for text chunking with multi-classification concepts that have two or more classes.

Experiments were carried out to find out most suitable features the chunking task. The main features for the text chunking task have been identified based on the different possible combination of available word and POS tag context. The features also include suffix for all current words. The term suffix is a sequence of last few characters of a word. In this study, the researchers considered different combination of feature set which were discussed in chapter 4.

To do this research, the annotated dataset consisted of 725 sentences with 4689 tokens. 70% of this dataset (507) were used for training and the left were testing (218). To do the experiment, 7 different scenarios were considered. Those scenarios were used in different feature extractions and in the same dataset which discards 30% of training dataset for test. For evaluating the performance of the system, we use standard measures: Precision, Recall, F-measure and accuracy. The obtained performance results and the features used in both methods are shown. The notation in the scenario tables were described in the feature extraction table of chapter 4.

In the first scenario, we used all features which includes suffixes until one and two lengths (-2/-1) characters, two words from left and two words from right of the current word within corresponding POS tag of each tokens as a feature.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	$S_{i-2}\ S_{i-1}\ W_{i-2}\ W_{i-1}\ W_i\ W_{i+1}\ W_{i+2}$ $POS_{i-2}\ POS_{i-1}\ POS_i\ \ POS_{i+1}\ POS_{i+2}$	0.877 secs	73	72	72	81
SVM		0.511 secs	71	70	70	79

Table 6: Performance of the Afaan Oromo text chunking system in scenario 1

In the 2nd scenario, we used all features except suffix which includes two words from left and two words from right of the current word within corresponding POS tag.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	$W_{i-2}\ W_{i-1}\ W_i\ W_{i+1}\ W_{i+2}$ $POS_{i-2}\ POS_{i-1}\ POS_i\ \ POS_{i+1}\ POS_{i+2}$	0.6 secs	75	74	75	83
SVM		0.245 secs	72	69	70	79

Table 7: Performance of the Afaan Oromo text chunking system in scenario 2

In the 3rd scenario, all features which have the same length in scenario-2 except POS tags were used.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	$S_{i-2}\ S_{i-1}\ W_{i-2}\ W_{i-1}\ W_i\ W_{i+1}\ W_{i+2}$	0.52 secs	60	56	57	71
SVM		0.202 secs	59	52	54	68

Table 8: Performance of the Afaan Oromo text chunking system in scenario 3

In the 4th scenario, we used two words from left and one word from right of the current word within corresponding POS tag of each tokens as a feature, except suffixes.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	$W_{i-2}\ W_{i-1}\ W_i\ W_{i+1}$ $POS_{i-2}\ POS_{i-1}\ POS_i\ \ POS_{i+1}$	0.474 secs	78	75	76	84
SVM		0.158 secs	71	70	70	80

Table 9: Performance of the Afaan Oromo text chunking system in scenario 4

In the 5th scenario, we used two words from left with their corresponding POS tag of each tokens as a feature. In this experiment, we have not used any words from next of current word and suffixes.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	W_{i-2} W_{i-1} W_i	0.476 secs	74	70	71	83
SVM	POS_{i-2} POS_{i-1} POS_i	0.21 secs	63	61	60	74

Table 10: Performance of the Afaan Oromo text chunking system in scenario 5

In the 6th scenario, one word from left and two words from right of the current word within corresponding POS tag of each tokens as a feature and suffixes feature set were not considered.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	W_{i-1} W_i W_{i+1} W_{i+2} $POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	0.635 secs	74	72	73	84
SVM		0.169 secs	70	68	69	78

Table 11: Performance of the Afaan Oromo text chunking system in scenario 6

In the 7th scenario, we used two words from right of the current word within corresponding POS tag of each tokens as a feature. In this experiment, any words from left current word and suffixes were not included.

Methods	Features used	Time taken	P (%)	R (%)	F(%)	A (%)
CRF	W_i W_{i+1} W_{i+2} POS_i POS_{i+1} POS_{i+2}	0.702 secs	72	66	68	80
SVM		0.176 secs	62	59	59	72

Table 12: Performance of the Afaan Oromo text chunking system in scenario 7

Experimental result summary

The above experiments show that different combination of features give different performance result. Table 13 shows the summarized of all the results.

No	Features used in different Scenarios	Measurements (%)							
		CRFs Methods				SVM Methods			
		P	R	F	A	P	R	F	A
Scenario-1	S_{i-2} S_{i-1} W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2} POS_{i-2} POS_{i-1} POS_i POS_{i+1} POS_{i+2}	73	72	72	81	71	70	70	79
Scenario-2	W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2} POS_{i-2} POS_{i-1} POS_i POS_{i+1} POS_{i+2}	75	74	75	83	72	69	70	79
Scenario-3	S_{i-2} S_{i-1} W_{i-2} W_{i-1} W_i W_{i+1} W_{i+2}	60	56	57	71	59	52	54	68
Scenario-4	W_{i-2} W_{i-1} W_i W_{i+1} POS_{i-2} POS_{i-1} POS_i POS_{i+1}	78	75	76	84	71	70	70	80
Scenario-5	W_{i-2} W_{i-1} W_i POS_{i-2} POS_{i-1} POS_i	74	70	71	83	63	61	60	74
Scenario-6	W_{i-1} W_i W_{i+1} W_{i+2} POS_{i-1} POS_i POS_{i+1} POS_{i+2}	74	72	73	84	70	68	69	78
Scenario-7	W_i W_{i+1} W_{i+2} POS_i POS_{i+1} POS_{i+2}	72	66	68	80	62	59	59	72

Table 13: Performance of the Afaan Oromo text chunking system in scenario summary

5.3.1 Discussion result of CRFs and SVM

Table 13 includes the summary result of both CRFs and SVM in all scenarios. We used the same dataset as well as the same features set for both CRFs and SVM. In the next, first we discussed CRFs results and then we discussed on SVM results. Lastly, we compared the result of CRFs and SVM results in Afaan Oromo text chunking.

The maximum average of CRFs results are scenario-4 with (precision78%, recall 75%, F-score 76% and accuracy 84%). scenario-6 with (precision74%, recall 72%, F-score 73% and accuracy 84%). The minimum result was in scenario-3(precision 60%, recall 56%, F-score 57% and accuracy 71%). We saw increased window size of tokens and POS tag are important feature for Afaan Oromo text chunking performance. Increasing window size was depends on length of sentences and depends on data set size. If the majority sentences in dataset are longer like double complex sentences of Afaan Oromo, increased widow size achieves better performance of the

system was possible. In this research, the majority of dataset was not long that means simple sentences plus exclamatory and interrogative sentences are almost in number and they are short lengths. The size of dataset used in this experiment was also very small. For these case, we used window size of feature until -2/+2. For this small dataset which contains majority of short length, increased window size of feature sets more and more were more immutable on the results.

Due to the majority of this dataset lengths are short length, removed feature set in the next to next from current word was achieve better result in this experiment. In another meaning, in the next to next from current word was no more impact on the result because of majority sentences of dataset are short length. In the short length of Afaan Oromo sentences, the word which more influence the current word are previous word of current word and next word of current word rather than previous to previous word from current word and next to next word from current word. When we compared the previous to previous word from current word with next to next word from current word, the previous to previous word from current word some important than next to next word from current word (we got this information from scenario-4).

The other is using suffix as a feature set in CRFs. We also understood when we used CRFs the suffixes of Afaan Oromo has impact on text chunking System. In Afaan Oromo language, some suffixes which attached on words change the word classes. As an example: "**Buna**/Coffee and **Bunatti**"/in coffee. The first word is a noun word class and the second word is noun which has a suffix that describe as postposition. So, currently, "Bunatti" is a postposition word class. When we removed the last characters that was ('-tti') from Bunatti, this word was the same as the first word class 'Buna'. This has impact on the original word (Bunatti) and part of speech tagging of that word. Because, after '-tti' removed from 'Bunatti', it was 'Buna' which noun word class. For this case, the dataset before feature extraction and after feature extraction are contradicted to each other on same word. We got this information from accuracy scenario-1 (with suffix) to compared scenario-2 (without suffix) for Afaan Oromo language.

In this experiment, the time taken in each scenario of CRFs was different. The maximum time taken was stated in scenario-1 which we used all features set and the lowest time taken was stated in scenario-4. Here we saw, increased feature set size was stated the longest time in the loading training dataset.

Our second approach in this experiment was support vector machine (SVM). In support vector machine, increased window size of tokens and POS tag as feature achieved best performance.

Using suffixes of current word has not effect on the achieved best performance of the system. In developing NLP application, extract features format supported by different machine learning algorithms were different. Machine learning like decision tree, SVM supported DictVectorizer feature. The term DictVectorizer can be used to convert feature arrays represented as lists of standard Python dict objects to the NumPy/SciPy representation used by scikit-learn estimators. DictVectorizer is also a useful representation transformation for training sequence classifiers in Natural Language Processing models that typically work by extracting feature windows around a particular word of interest. This DictVectorizer term transform extracted feature into numerical values that maintain the notion of ordering.

In this experiment, the better result of SVM achieved in scenario-4 then scenario-1 and scenario-2 respectively. From this result, we saw used suffixes as a feature was not effect on performance of the system. Due to this, the SVM chooses the decision boundary that maximizes the distance from the nearest data points of all the classes to predict the better categories. So, SVM was considered on decision boundary, but it was not considered suffix availability.

During this experiment, the time taken in each scenario of SVM was also different. The maximum time taken was stated in scenario-1 which we used all feature set and the lowest time taken was stated in scenario-4. Here we saw, increased feature set size was stated the longest time in the loading training dataset like in CRFs.

5.3.2 Comparison result of CRFs and SVM

CRFs have many promising result than SVM by using the same feature extraction window size and the same dataset which discards 30% of training data for test. The maximum result of CRFs was achieved in the fourth scenario. In this scenario, we used two words from left and one word from right of the current word within corresponding POS tag of each tokens as a feature and except suffixes of words. The achieved result in this fourth scenario was precision78%, recall 75%, F-score 76% and accuracy 84%. The maximum result of SVM was also achieved in the fourth scenario. In the fourth scenario, we used two words from left and one word from right of the current word within corresponding POS tag of each tokens as a feature, except suffixes. The achieved result in this fourth scenario was precision71%, recall 70%, F-score 70% and accuracy 80%. We summarized the compared better result of each methods according to the below table.

Methods	Scenarios	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)
CRFs	Scenario-4	78	75	76	84
SVM	Scenario-4	71	70	70	80

Table 14: Comparison better result of CRFs and SVM

To answer the first research question, the feature set used in the fourth experiments scenario is used to answer the first research question. In the fourth scenario, feature set used by CRFs and SVM are two words from left and one word from right of the current word within corresponding POS tag of each tokens as a feature were achieved better result. To answer the second research question, CRFs method fit text chunking system for Afaan Oromo language in this dataset.

Finally, from this comparison result, we understand, CRFs was more applicable for Afaan Oromo text chunking than SVM in this dataset. The lower performance could be due to the concept SVM is text classification and it is chooses the decision boundary that maximizes the distance from the nearest data points and the CRFs approach is sequence labeling classification algorithm. Text chunking is a sequential labeling problem which has concept of CRFs.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Chunking is the process of identify syntactically correlated parts of words in a sentence. Chunking can be useful in parsing, anaphora, information retrieval, Question answering system, information extraction, named entity recognition, machine translation. This research was conducted for Afaan Oromo text chunking using Conditional random fields (CRFs) and Support Vector Machine (SVM). To do this research, 725 sentences dataset (507 training and 218 testing) were used. This sentence was collected manually for this work. After collecting sentences, we were labeling POS tag and chunk tag.

During the experiment, seven different scenarios were considered based on the different combination of features such as word, part of speech tags of tokens and suffix. Since feature selection plays a crucial role in CRF and SVM framework, experiments were carried out to find out most suitable features for Afaan Oromo tagging task. In the first scenario all features were considered. In the second scenario all features except suffix were considered. In the third scenario, the features (two words from left and right, suffix, except POS tag) were considered. In the fourth scenario, the features (two words from left and one word from right, POS tag, except suffix) were considered. In the fifth scenario, the features (two words from left, POS tag, except any words from right and suffix) were considered. In the sixth scenario, the features (one word from left and two words from right, POS tag, except suffix) were considered. In the seventh scenario, the features (two words from right, POS tag, except any words from left and suffix) were considered.

The better result of CRFs and SVM methods are scenario-4 with (precision78%, recall 75%, F-score 76% and accuracy 84%); and (precision71%, recall 70%, F-score 70% and accuracy 80%) respectively achieved. This research indicated that a conditional random field (CRFs) is more applicable to Afaan Oromo text chunking than SVM. The feature set achieved better in CRFs methods were two words from left and one word from right and POS tag. This experimental result shows that the performance of developed chunker is significantly good for first time of Afaan Oromo text.

6.2 Challenge of text chunking for Afaan Oromo

During investigating text chunking for Afaan Oromo, many challenges were encountered by the researchers. Some of them are listed below:

1. The primary challenge was preparing chunking dataset of Afaan Oromo. Afaan Oromo hasn't had as such well resources that could have helped for exploring. For this research, we prepared manual from different sentences of Afaan Oromo. The maximum sentences in the prepared dataset were short length. When we collect sentences of Afaan Oromo from different source, almost, the available sentences were short length. In the short length sentences almost the phrase was noun and verb phrases. This impact on the performance of the others phrases.
2. The second challenge was getting POS tagged dataset from previous researchers on Afaan Oromo POS tagging. Even if [25], [32] and [33] investigated Afaan Oromo POS tagging; however, there was no chance of getting POS tagged data set from those researcher. POS tagging is basic step in investigating text chunking. That means chunking must incorporated POS tagger dataset. So, for our experiment, we prepared POS tagging data manually. Labeling POS tag and chunk tag dataset was expensive.

6.3 Contribution of the work

- ✓ 725 sentences dataset within POS tagged and chunk category were prepared.
- ✓ Text chunking model was investigated with CRFs and SVM for Afaan Oromo language.
- ✓ The CRFs approach result shows better accuracy for Afaan Oromo Text chunking than Support vector machine approaches.
- ✓ Important features in Afaan Oromo Text chunking were identified in both CRFs and SVM.

6.4 Future works

The study has shown that Afaan Oromo Text chunking can be done automatically using conditional random field algorithm and Support vector machine learning algorithm. listed below are immediate recommendations and future works:

- ➢ With regard to dataset, Afaan Oromo language didn't have dataset which is manually tagged with POS and chunk categories. For this study, small amount of corpus is tagged manually but this corpus is insufficient and very small. So, in the future huge amount of dataset for Afaan Oromo chunking has to be prepared and the system has to be trained on

that to improve its performance. Also, if sentence in dataset are equal in a numbers, it improved the performance of the system.

- Conducting this study with hybrid approach is recommended since it might give a better performance.
- Investigate this research also with deep learning techniques. Deep learning is advanced techniques but it use huge amount of dataset when we compare within others machine learning. So, in the future huge amount of dataset for Afaan Oromo chunking has to be prepared and the system has to be trained by deep learning that to improve its performance.
- Reproduce this work for others Ethiopian languages.

REFERENCE

[1] Peter Jackson and Isabelle Moulinier, natural language processing for online applications: text retrieval, extraction and categorization, 1984

[2] Daniel Jurafsky & James H. Martin. Speech and Language Processing: An introduction to natural language processing,computational linguistics, and speech recognition. 2006

[3] Karen Sparck Jones,Natural Language Processing, a historical review, University of Cambrige, Cambrige, October 20

[4] Biplav Sarma, Anup Kumar," A Comprehensive Survey of Noun Phrase Chunking in Natural Languages" International Journal of Engineering Research & Technology (IJERT) Vol. 4 Issue 04, April-2015

[5] Lammii Kabbabaa "Kitaaba deeggarsa barnoota Afaan Oromoo kutaalee 7ffaa fi 8ffaatiif qophaa'e" 2019

[6] Daelemans, W., S. Buchholz, and J. Veenstra. 1999. Memory-based shallow parsing. In proceedings of CoNLL, Bergen, N. submitted.

[7] S. P. Abney. Parsing by chunks (pp. 257-278). Springer Netherlands. 1992

[8] Soheila Kian, Tara Akhavan, Mehrnoush Shamsfard. Developing A Persian Chunker Using a Hybrid Approach. Proceedings of the International Multi conference on ISBN 978-83-60810-224 Computer Science and Information Technology, pp. 227 – 233

[9] Census Report (2007) "Ethiopia's population now 92 million"

[10] GuyaT (2003) CaasLuga Afaan Oromoo: Jildii-1, Gumii Qormaata Afaan Oromottin Komishinii "Aadaa fi Turizimii Oromiyaa",Finfinnee.

[11] D. Jurafsky & J. H.Martin. Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition. Upper Saddle River, NJ: Prentice Hall. 2000.

[12] Gaddisa H . Automatic syntactic parser for Afaan Oromo complex sentence using context free grammar.Addis ababa university, Unpublished, October 2016.

[13] Diriba M, An Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique, Addis ababa university, Unpublished, june 2002.

[14] Asopa, Sneha., Asopa, Pooja. (2016). Iti Mathur, and Nisheeth Joshi. Rule based chunker for Hindi. In: 2nd International Conference on Contemporary Computing and Informatics, p. 242–245, March 2016.

[15] Akshay Singh, S M Bendre, and Rajeev Sangal. HMM based chunker for Hindi. In: 2009 Proceedings of the Second International Joint Conference on Natural Language Processing, October 2005

[16] R. Koeling (2000). Chunking with maximum entropy models. In Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7 (pp. 139-141) . Association for Comput ational Linguistics.

[17] T. Kudo & Y. Matsumoto (2001). Chunking with support vector machines. In Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies (pp. 1-8). Association for Computational Linguistics.

[18] Feifei Zhai et al. Neural Models for Sequence Chunking. IBM Watson. 1101 Kitchawan Road, Yorktown Heights, NY 10598

[19] Abeba Ibrahim. A Hybrid Approach to Amharic Base Phrase Chunking and parsing (Unpublished Master‟s thesis). Addis Ababa University, Addis Ababa, Ethiopia. 2013.

[20] F. Xu, C. Zong, & J. Zhao,"A Hybrid Approach to Chinese Base Noun Phrase Chunking". In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (Vol. 22-23, p. 87-93). Sydney, 2006

[21] Tesfaye G, Afaan Oromo Parser using Hybrid Approach, Jimma university, Unpublished, june 2017.

[22] Misganu T. and Ramesh B, "A Novel Anaphora Resolution Model for Afaan Oromo Language Texts." In: International Journal of Engineering Research & Technology (IJERT), 2019

[23] U. Jain and J. Kaur, "A Review on Text Chunker for Punjabi Language" vol.4, no.7 Vol. 4, no. 7, July 2015

[24] Addunyaaa Barkeessaa, “Semmoo: Buʻuura Barnoota Afaaniifi Afoola Oromoo”,Finfinne, Ormiyaa, Far East Trading PLC. 2014

[25] Abraham T, Degen H, et al. “Automatic Part-of-speech Tagging for Oromo Language Using Maximum Entropy Markov Model (MEMM)”, ournal of Information & Computational Science, 2014

[26] Addunyaaa Barkeessaa. “Sanyii: Jechaafi Caasaa Isaa(Word and its structure)”. Finfinne, Ormiyaa. 2013.

[27] Mahdi, Hamid M., Oromo Dictionary, English-Oromo, Sagalee Oromo Publishing co. Atlanta, Georgia 1995

[28] Ton Leus. Et al., Borena Dictionary, W.S.D. Grafisch Centram Schijndel, Holland, 1995

[29] Baye Yimam, “The Phrase Structure of Ethiopian Oromo”. The Degree of Ph. D in Linguistics, School of Oriental and African University of London, London, 1986

[30] Hamiid M., English-Oromo Dictionary, Sagalee Oromoo Publishing Inc, Atlanta, 1996

[31] J. Lafferty, A. McCallum, F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proceedings of the Eighteenth International Conference on Machine Learning (lCML).

[32] Getachew Mamo, Parts of Speech Tagging for Afaan Oromo, International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence

[33] Abraham Tesso. Et al.., Automatic Part-of-speech Tagging for Oromo Language Using Maximum Entropy Markov Model (MEMM), journal of Information & Computational Science, 2014

[34] Abdi Sani, Afaan Oromo named entity recognition using hybrid approach, Master’s Thesis, Addis Ababa University, 2015

[35] Tilahun Gamta, The Oromo language and the latin alphabet, Journal of Oromo Studies, 1992.http://www.africa.upenn.edu/Hornet/Afaan_Oromo_19777.html last visited on Friday, October 31, 2014.

[36] Tabor Wami's, new book titled, Yewugena Dirsetoch ena Yetarik Ewunetawoch,2004.

[37] Workineh Tesemaa, “Designing a Rule Based Disambiguator for Afan Oromo Words”. In: American Journal of Computer Science and Information Technology, 2017

[38] B. Addunyaa, NATOO: Yaadrimee caasluga Afaan Oromoo. Addis Abebe, 2012.

[39] Guang-Lu S. Et al.., “Chinese Chunking Based on Naive Bayes Model”. In: Research Institute of Information Technology, Tsinghua University, Beijing, China, 2010.

[40] Nabil KHOUFI. Et al.., “Chunking Arabic Texts Using Conditional Random Fields”

[41] Kuang hua Chen., Hsin-Hsi Chen. (1993). A probabilistic chunker. In: Proceedings of ROCLING-93, p. 99–117.

[42] Chakraborty, Neelotpal., Malakar, Samir., Sarkar, Ram., Nasipuri, Mita. (2016). A rule based approach for noun phrase extraction from English text document. In: Seventh International Conference on CNC-2016, p. 13–26.

[42] Guang-Lu S. Et al.., “Chinese Chunking Based on Maximum Entropy Markov Models”. In: The Association for Computational Linguistics and Chinese Language Processing on June 2006 pp. 115-136.

[43] Yuqi Z.,Qiang Z,.” Chinese Base-Phrases Chunking” State Key Laboratory of Intelligent Technology and Systems Department of Computer Science and Technology Tsinghua University, Beijing China

[44] Vivekananda S and Vivekananda G, “Bengali Noun Phrase Chunking Based on Conditional Random Fields”, In: International Conference on Business and Information Management (ICBIM). 2014

[45] Prathibha RJ, Padma MC, “Shallow Parser for Kannada Sentences using Machine Learning Approach”, n: International Journal of Computational Linguistics Research Volume 8 Number 4 December 2017.

[46] Asif Ekbal and Sivaji Bandyopadhyay,2010, Named Entity Recognition using Support Vector Machine: A Language Independent Approach: International Journal of Electrical, Computer, and Systems Engineering 4:2 2010.

[47] Joel Mickelin,2013, Named Entity Recognition with Support Vector Machines: MSc Thesis, Royal Institute of Technology School of Computer Science and Communication, Stockholm, Sweden.

[48] Chih-Wei Hsu and Chih-Jen Lin, “A Comparison of Methods for Multi-class SupportVector Machines”

[49] Roman Klinger, et.al. Classical ProbabilisticModels and Conditional Random Fields. Algorithm Engineering Report, TR07-2-013, ISSN 1864-4503. Dortmund University of Technology, Department of Computer Science, Germany, 2007

Appendices

Appendix 1: POS tagged used in this study

No	Tag	Description	Example
1	NN	Common noun	Nama, Muka, jimmaa
2	NNP	A tag for all types of plural nouns that are not joined with other categories in sentence (not show location and direction)	Namoota,biyyoota, Namni,sangichi,
3	NNPS	Common noun plus pre/postposition (which is show location and direction)	Inni **Booranattii** gale.
4	PN	proper nouns	Dammee,Caaltuu, Oromiyaa
5	PNS	Proper nouns that are joined with other postposition in sentence (not show location and direction)	Tolaan, caaltuun
6	PPN	Personal pronoun	Ana/ani, nuyi/nuti, sii/ati, isin, ishee
7	POP	Possessive Pronoun	Koo, kee, keenyaa
8	VB	Verbs	Kottu, Deemi, dubbisi, qaba
9	JJ	Adjectives	Bareedduu,diimaa, magaala,
10	AD	Adverb	Kaleessa,edana, yoomiyyuu,sirritti
11	APC	Adpositions, pre-/postposition and conjunctions	Ni, irraa, garuu, fi, akka,yoom,ykn, bira, gara,
12	DT	Determiners	Kun, kana, kanneen, sana
13	CN	Cardinal numbers.	Lama, kudhan, 2012
14	PUNC	Punctuations	. ? ! ‘ : [{ ( ) =-__
15	NEG	Negative word	Hin,miti
16	PPNS	Personal pronoun + postposition	Nurraa
17	ADPP	Adverb plus postposition “-ttii”	Tulluun qawwee dhukaasee [*fagootti*] bineensa ajjeese

Appendix 2: chunk tag used in this study

No	Chunk tag	Description
1	B-NP	Begin of noun phrase
2	I-NP	Inside of noun phrase
3	B-VP	Begin of verb phrase
4	I-VP	Inside of verb phrase
5	B-AdjP	Begin of adjective phrase
6	I-AdjP	Inside of adjective phrase
7	B-PP	Begin of postposition phrase
8	I-PP	Inside of postposition phrase
9	B-AdvP	Begin of adverb phrase
10	I-AdvP	Inside of adverb phrase
11	O	Outside of adverb phrase

Appendix 3: Sample chunk tag text for the training data set

Inni	PPN	B-NP
sa'a	NN	B-NP
Aannanii	NNP	I-NP
Bitate	VB	O
.	PUNC	O
Dallaan	NNP	B-NP
Loonii	NNP	I-NP
Kaleessa	AD	B-VP
Ijaarame	VB	I-VP
.	PUNC	O
Tolaan	PNS	B-NP
Nama	NN	B-NP
Ambooti	NPP	I-NP
.	PUNC	O
Gammachuun	PNS	B-NP
Suuta	AD	B-VP
Deema	VB	I-VP
Darbe	VB	I-VP
.	PUNC	O
Caaltuun	PNS	B-NP
Kaleessa	AD	B-VP
Daftee	AD	I-VP
Deebite	VB	I-VP
.		O
Bariisoon	PNS	B-NP
Garmalee	AD	B-VP
Soorame	VB	I-VP
.	PUNC	O

Appendix 4: Feature extracted:

Listed sentences:

```
[['Namooni', 'NNP', 'B-NP'],
 ['guguddaa', 'JJ', 'B-AdjP'],
 ['saddeeti', 'CN', 'I-AdjP'],
 ['dhufan', 'VB', 'O'],
 ['.', 'PUNC', 'O']]
```

Feature extracted file of the above listed sentences in all features set in CRFs:

```
 [{'bias': 1.0,
  'current_token.lower()': 'namooni',
  'suffix-2': 'ni',
  'suffix-1': 'i',
  'current_token.isdigit()': False,
  'current_postag': 'NNP',
  'First_token': True,
  'next_1_token_lower': 'guguddaa',
  'next_2_token_lower': 'saddeeti',
  'next_1_pos': 'JJ',
  'next_2_pos': 'CN'},
 {'bias': 1.0,
  'current_token.lower()': 'guguddaa',
  'suffix-2': 'aa',
  'suffix-1': 'a',
  'current_token.isdigit()': False,
  'current_postag': 'JJ',
  'prew_1_token_lower': 'namooni',
  'prew_1_pos': 'NNP',
  'next_1_token_lower': 'saddeeti',
  'next_2_token_lower': 'dhufan',
  'next_1_pos': 'CN',
  'next_2_pos': 'VB'},
 {'bias': 1.0,
  'current_token.lower()': 'saddeeti',
  'suffix-2': 'ti',
  'suffix-1': 'i',
  'current_token.isdigit()': False,
  'current_postag': 'CN',
  'First_token': True,
  'prew_2_token_lower': 'namooni',
  'prew_1_token_lower': 'guguddaa',
  'prew_2_pos': 'NNP',
  'prew_1_pos': 'JJ',
  'next_1_token_lower': 'dhufan',
```

'next_2_token_lower': '.',
'next_1_pos': 'VB',
'next_2_pos': 'PUNC'},
{'bias': 1.0,
'current_token.lower()': 'dhufan',
'suffix-2': 'an',
'suffix-1': 'n',
'current_token.isdigit()': False,
'current_postag': 'VB',
'First_token': True,
'prew_2_token_lower': 'guguddaa',
'prew_1_token_lower': 'saddeeti',
'prew_2_pos': 'JJ',
'prew_1_pos': 'CN',
'next_1_token_lower': '.',
'next_1_pos': 'PUNC'},
{'bias': 1.0,
'current_token.lower()': '.',
'suffix-2': '.',
'suffix-1': '.',
'current_token.isdigit()': False,
'current_postag': 'PUNC',
'First_token': True,
'prew_2_token_lower': 'saddeeti',
'prew_1_token_lower': 'dhufan',
'prew_2_pos': 'CN',
'prew_1_pos': 'VB',
'last_token': True}]

Appendix 5: Sample predicted chunk tagged output

Input file:

[('Tolaan', 'PNS'), ('gara', 'APC'), ('manaa', 'NN'), ('deeme', 'VB'), ('rafe', 'VB'), ('.', 'PUNC')]

Predicted output:

[(('Tolaan', 'PNS'), 'B-NP'),
(('gara', 'APC'), 'B-PP'),
(('manaa', 'NN'), 'I-PP'),
(('deeme', 'VB'), 'B-VP'),
(('rafe', 'VB'), 'I-VP')]