JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

AMHARIC TEXT CHUNKER USING MACHINE-LEANING APPROACH

By

Birhan Hailu Getaneh

The Thesis Submitted to the School of Graduate Studies of Jimma University in Partial fulfillment of the Requirements for the Degree of Masters of Science in Information Technology

JUN, 2019

Jimma, Ethiopia

Declaration

I, undersigned, declare that this thesis entitled Amharic text chunker using machine-leaning approach has not been presented for a degree in this or any other universities, and that all sources of material used for the research have been duly acknowledged.

Declared by:

Name: Berhan Hailu

Signature: ________________

Date: Jun, 2019

This thesis has been submitted for examination with our approval as university advisors.

Dr. Gebeyehu Belay(Dr. of Eng.) Assoc. Professor [illegible] Jun,2019

Principal Advisor's

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

AMHARIC TEXT CHUNKER USING MACHINE-LEANING APPROACH

By

Birhan Hailu Getaneh

Signature of the Board of Examiners for Approval

Internal examiner	______________	______	______
	Name	Signature	Date
External examiner	______________	______	______
	Name	Signature	Date
Chairperson	______________	______	______
	Name	Signature	Date

Acknowledgement

Before Everything I would like to thank almighty GOD from my bottom of heart for his blessing and steering throughout my life including this work. Next to GOD, I would like to express my deep-felt special gratitude to my advisor, Dr. Gebeyehu Belay, from Bahir dar University for his great advice, encouragement, enduring patience and constant support from the beginning till end and this work was impossible without his patience and assistance and Mr. Geletaw Sehle my co-advisor for his great help.

I went thank Dr. Getachew Mamo and Mr. Tefere Kebebew for their help by sharing their experience on title selection and commenting the concept paper before the beginning of the work. I would also like to thank who was involved in manually preparing training and testing corpora for this thesis: Dr. Getachew Anteneh, from jimma university who give suggestion and constructive comment during corpora preparation from linguistic perspective.

Moreover, I would like also to thank Jimma University for its financial assistance. At the times of my stay, I erudite a lot of experiences that could be a change of course of my of life.

Finally, I would like to show my profound appreciation to all those who have provided me assistance and guidance including my families, best friends and classmate.

Table of Contents

List of tables

List of figures

Acronyms

ATC-Amharic Text Chunker

CRFs-Conditional Random Fields

HMM-Hidden Markov Model

MBL-Memory-Based Learning

MBT-Memory Based Tagger

ML-Machine Learning

MVDM-Modefied Value Difference Matric

NLP-Natural Language Processing

NP-Noun Phrase

PP-Preposition Phrase

TC-Text Chunking

TiMBL- Tilburg Memory-Based Learner

VP-Verb Phrase

IOB-Inside Outside Beginning

POS-Part Of Speech

NER-Named Entity Recognition

W-Word

WIC- Wolta Information Center

Abstract

Natural language processing has an important role in our daily life, by enabling computers to understand human languages. Text chunking is one of an essential task in NLP applications. The Information generated by this task can be helpful for many purposes, including automatic text summarizing, question answering, information extraction and so on. Text chunking or shallow parsing is a kind of NLP task, which is the process of grouping the input text or sentence into syntactically related non-overlapping part of words or chunks like Noun Phrase, Verb Phrase, Prepositional Phrase, Adjective Phrase and so on. Generally, the notion of TC is a words or chunks should be a member of one syntactic structure; chunks can't be a member of two or more syntactic structure.

The objective of this research work is to develop Amharic text chunker using machine learning approach, specifically by adopting conditional random fields and memory-based learning. To get the optimal feature set of the chunker; the researcher's conduct different experiment using different scenarios until a promising result obtained.

In this study different sentences are collected from Amharic grammar books,new articles,magazines and news of Walta Information Center (WIC) for the training and testing datasets. Unlike the data collected from WIC, the data collected from Amharic grammar books ,new articles,megazines are not tagged at all. Thus, these datasets were analyzed and tagged manually and used as a corpus for our model training and testing. But the entire datasets were chunk tagged manually for the training dataset and approved by linguistic professionals. For the identification of the boundary of the phrases IOB2 chunk specification is selected and used in this study.

Experiments have been conducting using the training and testing datasets using different scenarios to get better accuracy of the chunker. The experiments on Amharic text chunking scored the highest accuracy of 97.26% and 82.08% using CRFs and MBL respectively.

Chapter one: Introduction

1. Background

Now a day, Natural language processing (NLP) has an important role in our daily life, by enabling computers to understand human languages[1]. Text chunking (TC) is one of an essential task in NLP applications. The Information generated by this task can be helpful for many purposes, including automatic summarizing or question answering. It can be either shallow parsing or deep parsing [2].

Text chunking or shallow parsing is a kind of NLP task, which is the process of grouping the input text or sentence into syntactically related non-overlapping part of words or chunks like NP(Noun Phrase), VP (Verb Phrase), PP (Prepositional Phrase), AdjP (Adjective Phrase) and so on. Generally, the notion of TC is a word (chunk) should be a member of one syntactic structure(phrase), word(chunk) can't be a member of two or more syntactic structure(phrase).
For instance according to [1]: ትንሹ ልጅ ትንሽ እንጀራ በላ "The little boy ate little Injera"
Here the sentence can be segmented into three basic syntactic non-overlapping phrases: a noun phrase ትንሹ ልጅ "the little boy", another noun phrase ትንሽ እንጀራ "little Injera", a verb phrase በላ "ate".

As indicated by Nabil KHOUFI, Chafik ALOULOU and Lamia HADRICH BELGUITH [3], TC becomes an interesting alternative to full parsing since the main purpose of full parsing is a deep analysis of the syntactic structure of the sentence, these will decrease the performance of the system. However, the main purpose of TC is a higher-level analysis of a syntactic structure of a sentence, which is enough for most information retrieval(IR) and information extraction(IE) technology[4]. Due to thus reason integrating TC rather than full parser will increase the performance of IR and IE technologies.

The various studies indicate that many NLP and IR application needs chunking (shallow parsing or partial parsing) rather than full parsing. For instance, to develop fully fledged question answering system it needs Chunker as indicated by [4,5,6,7], Finding noun phrases and verb phrases may be enough for IR system. Commonly, TC can be integrated into information extraction, text mining, automatic summarization and so on.

Today different researchers investigated text-chunking for the various language of the world using a different methodology, for instance, Arabic[2], Chinese[8], Bengali[4], Amharic [9], Indonesian [10], Urdu [11]. As far as the researchers' knowledge concerned, only one work was done for Amharic. The work done by Abeba [9] is the first Amharic TC, their work didn't include all kind of Amharic sentence. They used HMM for chunking the text and then error-pruning rules to correct chunks which are incorrectly chunked by the HMM model. however as indicated by [8] and [4], HMM model has a label bias problem.

Even if the above works have been done in different languages on different aspects of text chunking, it is not possible to apply this text chunker to Amharic language context directly. The main reason is the Amharic language is highly inflectional, morphologically rich and has its own letters and grammatical rules.

As indicated above TC is one of the research areas under computational linguistics using different approaches. The basic approaches toward TC are [4]: rule-based and machine learning or Statistical approach. The rule-based approach is based on a handcrafted rule to identify the syntactic structure of a text. Its drawbacks are requirement of huge linguistic knowledge, very large number of rules to cover all the features of a language, highly language dependent so we can't adapt for another language easily, it consumes too much time to develop the rule**.** In machine learning approach there is no need of building chunking rules manually; so, the machine by itself generate a set of rules from the training data. This approach mainly categorized into two classes: supervised and unsupervised approaches. In supervised ML approach needs labeled training dataset to induce a set of rules from the labeled dataset to classify the testing dataset (unseen in training dataset). However, unsupervised ML approach doesn't need labeled training dataset. There is no teacher or supervisor providing supervision as to how individual instances should be handled. The main challenge to the unsupervised approach considers the features independently and are unable to capture the relationship between different features of a training data which results in lower accuracy[12]. Due to this, the reason we select supervised ML approach to conduct such an investigation toward Amharic TC.

Conditional Random Fields (CRFs) are one class of supervised ML, it was applied for different IR and NLP applications, where they are used for classification of sequential data [2].it were first introduced by [13] for the different sequential annotation tasks. Some of them were named entity recognition, POS tag, text chunking etc., where they get excellent results ([8],[4],[2]).

Memory-based learning (MBL) is also the other class of supervised ML. It is applicable for many areas of NLP tasks like POS tag, NER, morphological analysis, etc. the basic notion of MBL is learning from experience rather than deriving a new solution to each problem, it uses the existing knowledge or experience to answer new problem [14]. The fashion MBL solve new problem is based on the similarity matrix between new problem and existing solved problem in memory. The usual similarity measures are k-Nearest Neighbors (k-NN) and Modified Value Difference Metric (MVDM). MBL has been used for text chunker for Malayalam [15], they got an encoraging result. Due to the above reason, we are motivated to investigate TC for Amharic language using them (i.e. CRF based and memory-based TC).

The aim of this thesis is to investigate and develop a model for Amharic text chunker using supervised machine learning techniques i.e. CRFs and MBL. To develop the model firstly, we have manually tagged our training dataset using both POS tag and chunk tag. To create the chunk tag, we were used IOB chunk specification, and then we added the annotated dataset to CRFs and MBL to build the model. We have investigated a different feature of Amharic language to build the best model which chunk in a well-structured manner and better accuracy than the existing Amharic TC. Finally, we have evaluated our model using the test dataset.

1.2 Statement of the problem

The motivation that initiated the researcher to conduct this research toward Amharic text chunker is the advantageous and application areas of text chunker in different NLP and IR tasks. for instance, in information retrieval task dividing the given search keyword text into different syntactically correlated part of chunk can increase the performance of the searching time and also in phrase level machine translation chunker can be one component which helps the translation process by dividing the stream of text into chunk(phrase).

Amharic ranked as the second by the number of the speaker under Semitic language in the world next to Arabic[16]. Despite having large speaker population, the language has little computational linguistic resources. So, conducting a research on such kind of language is advisable to overcome the current shortage of computational linguistic resources. Amharic is one of the under-resourced languages. Hence, there is no well-designed full-fledged Amharic text chunker that could contribute towards designing NLP. So that conducting Amharic ATC with best performance will have a contribution to advance research in NLP for Amharic language.

Text chunking is one of the important tools in NLP. It serves as an intermediate component for different higher-level NLP and IR applications like machine translation, question answering, and information extraction and so on. Currently, many researchers have been developed different NLP application related with the Amharic language; for instance, Amharic question answering [6]; the researchers indicates that to develop full-fledged question answering, chunker is one component for the system. And also as indicated in [17], text chunker increases the performance of Amharic Anaphora Resolution. In addition to the aforementioned application, chunker has many application indicated by other languages, for instance, according to[4],[9],[11],[3] finding noun phrases and verb phrases may be enough for IR System, since the main aims of TC is segmenting the input sentence into phrases that give us information about time, places, objects, etc., are more significant than the complete analysis of a sentence (full parsing). Generally, TC used as sub-component for different areas of NLP and IR applications like Question-answering, information extraction, text mining, machine translation, spoken language understanding, automatic summarization, and so on.

Even if chunker has a lot of advantage for higher-level NLP and IR application, Amharic language has not benefited from it due to lack of good text chunker. Hence, the development of the text chunker for Amharic sentences will have a vital role in the implementation of higher-level NLP and IR applications with better performance.

As described earlier, only one attempt was done by Abeba [1] and further improvement regarding Amharic text chunker has never been tried after the first ATC attempt. As the researcher knowledge concerned in the Nationwide also there is no attempts for developing a text chunker of other Ethiopian languages .The first attempt to ward amahric text chunker[1] used Hidden Markov Model (HMM) to develop Amharic base phrase chunker and bottom-up approach with transformation algorithm to transform the chunker to the parser. However,the size of the corpus is small and also as indicated by [4], [8] the model they use (i.e. HMM) has label bias problem. The researchers minimized the error incorrectely chunked by HMM model using error pruning rule, but major errors that are not pruned by rules are the tag sequence conflict; by taking these limitations into consideration we are motivated to further investigate the Amharic TC. So, in the proposed system the researcher tried to reconsider features that could increase the accuracy of Amharic text chunker.

Therefore, the research questions are defined and articulated as follows:

1. How to optimize the accuracy of an automatic Amharic text Chunker, and what contexts need to be reconsidering for its best achievement?
2. Which of the proposed approach is the best performer in chunking Amharic text either MBL or CRFs?
3. How to optimize the influential or (determinant) factor (s) to improve the performance of the proposed systems? How it is important than others?

1.3 The Objective of the Study

1.3.1 General Objective

The general objective of this research work is to investigate text chunking for Amharic using conditional random fileds and memory based learning.

1.3.2 Specific objectives

In order to achieve the main objective and answers to the research questions, the following specific tasks are done.

- To study the basic syntactic structure of Amharic sentence (i.e. phrase structure, word category, sentence construction etc.).

- To design a generic model for MBL and CRFs based text chunker for Amharic.
- To develop a prototype of the chunker that accepts Amharic text as an input and produces chunked text as an output.
- To conduct experiments to evaluate the performance of the proposed system.
- To evalut and compare the result of MBL and CRFs based models

1.4 Scope and limitation of the study

The study explores and develop Amharic text chunker based on MBL and based on CRFs for the Amharic language. After an investigation, the researchers select the best Amharic text chunker from the two (i.e. MBL and CRF based text chunker). Transforming the proposed text chunker into the full parser and integrating the developed chunker into higher level NLP and IR application is out of the scope of this thesis work ;In addition to it we exclude imperative and interrogative sentence from our investigation due to time constraint. The time to accomplish these thesis works was scheduled for around four months to finish up on june,2018, but due to some cases, the schedule post ponded to few months to finalize the work on the first week of Jun,2019.

1.5 Methodology

This research conducted in order to investigate an automatic text chunker for the Amharic language. Design science research methodology is followed for this study. To accomplish this and to meet the prescribed objectives the following subtasks were carried out in detail.

1.5.1 Data collection

We have used the Walta Information Center (WIC) corpora, which is prepared by the Ethiopian Languages Research Center of Addis Ababa University. in addition to it, we collect from Amharic grammar book, news article, and magazine.The Walta Information Center (WIC) corpora is annotated by POS tag ,however the data collected from Amharic grammar book,magazines and news article are not annotated by POS tag, we have annotated them manually due to lack of publicly available automatic part of speech tagger for Amharic. After finishing the annotation of POS tag, the researchers manually annotate chunk tag to the whole dataset and get comment and suggestion from Amharic linguistic expert. Finally, after finishing annotation the whole dataset with a syntactic structure (i.e. POS tag and chunk tag) the dataset categorized into a training dataset and test dataset to start the experiment.

1.5.2 Implementation Tool

As described earlier we are going to implement two popular supervised ML algorithm (i.e. CRFs and MBL). To achieve these objectives, we have adopted the package CRFsuite and Tilburg Memory-Based Learner (TiMBL) for CRFs and MBL respectively. Initially CRFs was introduced by [13], it has a CRFsuite package, which is open source; freely available conditional random filed implementation package, we used it to develop CRFs model. Tilburg Memory-Based Learner (TiMBL) is the package which includes different memory-based learning algorithms [18]. Memory-based Tagger (MBT) is one of the packages in TiMBL which used for POS tag and text chunking since Amharic TC is a kind of text chunking so we have implemented it.

1.5.3 Testing and Experimental procedure

Twelve experiments are conducted in this thesis work. The First four experiments are done using CRFs and the rest eight experiment are done using MBL. We have arranged the dataset to test the text chunker prototype. The accuracy of the system has been measured to evaluate the performance of the system. After evaluating, the best accuracy, which chunks text in good structure, from the two experiment selected for Amharic text chunker.
The accuracy of the model measured by the following mathematical formula

$$\text{Accuracy} = \frac{number\ \ of\ correctly\ predicted\ chunk\ tag}{total\ number\ of\ chunks}$$

1.6 Significant of the study

As indicated by[1,3,4,9,19] some of NLP and IR higher level application are beneficiary from text chunking, for instance Text summarization, Speech Synthesis, and Recognition, Information Retrieval (IR), Information extraction (IE), Machine Translation (MT), Parsing, Index Term Generation, Named Entity Recognition (NER) are some of them .

Some of the above chunking application explained by [19], are described below:-

- **Information extraction:** the chunker divides the sentence into chunks of interrelated data. Noun phrase and verb phrase are chunked and can be used in information extraction systems. IE focuses on discovering the names of people and events they participate in, from a document.
- **Named entity identification:** In this system, the main aim is to identify the particular words in the document. Such as people, places and other nouns in the sentence.
- **Search:** searching for a particular noun or verb can be done. As the sentence is chunked in pieces, search becomes an easy task and the whole chunk can be represented as the search result
- ***Machine translation*:** machine translation is the process of translating one language into another language. Chunking is useful in this task as the chunks are converted into another language.
- **Information retrieval:** indexing noun phrase

1.7 Organization of the thesis

This thesis is strcutured into six chapters.The next sub sequent chapter is structured as, in Chapter two,it is dedicated to a literature review. It discusses text chunking approaches, text chunking specification and reviews of related work. In chapter three, the characteristics of the Amharic writing system that are applicable to the research area are discussed to represent and process the Amharic texts in electronic format. In chapter four, dataset preparation and modeling procedure are discussed. The experimental settings, the process of the experimentation and the findings are presented in chapter five. Finally, in chapter six general conclusions and recommendations are made based on observations and results from the experiment.

Chapter two: Survey of related literatures

2.1 Introduction

This chapter deals with the state of the art works related to text chunking for the various language of the world. In order to complete this study, literature reviews are necessary to analyze and understand the previous researches that have been done in text chunking. In recent years, there have been many works done in Natural Language Processing (NLP). Text chunking is one of the most important NLP tools. As far as the researchers' knowledge concerned only one attempt were done for the Amharic language regarding text chunker, however a number of studies have been carried out in different foreign languages on different aspects of text chunking on various approaches. In the following sub-section, we are going to discuss in detail about text chunking approaches, related work which has been done before in the area of text chunking.

2.2 Text chunking approaches

Text chunking is one of the research areas under computational linguistics using different approaches. The basic approaches toward text chunking are [4]: rule based , machine learning or Statistical and hybrid approach.

2.2.1 Rule based approach

Rule based text chunker also known as Knowledge-Based or linguistic-based approach, which is the easiest approach of text chunker. It is based on generated handcrafted linguistics rule to build a text chunker, which needs deep linguistic knowledge about the target language to develop the rule.

The advantage of rule-based approach is: it is easy to implement and no need of training data preparation. However, there are also drawbacks such as requirement of huge linguistic knowledge, very large number of rules to cover all the features of a language, highly language dependent so we can't adapt for other language easily, it consumes too much time to develop the rule and difficulty to extend or scale-up the system.

2.2.2 Machine learning approach

In machine learning approach there is no need of building chunking rules manually. So, the machine by itself generates a set of rules from the training data. Developing a system with machine learning approach is relatively faster than the rule-based approach but it requires that a sufficient volume of training data to be available. Using these approaches there are two learning paradigms: supervised and unsupervised learning approach. Supervised approach learns by example whereas the unsupervised approach learns by pattern.

2.2.2.1 Supervised Approach

Supervised learning uses training data to automatically induce chunking rules. Thereby, almost no knowledge about the domain is necessary, but it needs training data to be annotated according to the underlying structure of information like POS tag and chunk tag. It uses probability or statistics in analyzing the syntactic structure of text. ML approach that use supervised learning paradigm includes support vector machine (SVM), Hidden Markov Model (HMM), conditional random fields (CRFs), memory-based learning and so on. Among different alternative approaches, the selection of the approaches deepened on the problem we went to tackle. In this work, we select conditional random fields(CRFs) and memory-based learning(MBL) to develop text chunker for Amharic, partly motivated by the limitation of the previous attempt using hybrid approach (i.e. HMM plus rule-based)[1].Both conditional random fields and memory-based learning have promising features in analyzing NLP tasks like Named entity recognition, POS tagger, semantic labeling, morphological analysis, text chunker due to these we are inspired to investigate using them.

2.2.2.1.1 Conditional random fields (CRFs)

CRFs are a kind of discriminative undirected probability graph model, it usually applied for segmenting and labeling of sequential data. It often used in a different area of NLP task, Specifically, named entity recognition (NER), information extraction (IE), POS tagging, chunking, and their result was promising. It has rich feature set which solves fundamental drawback of maximum entropy Markov models (MEMMs),hidden Markov model(HMM) and other discriminative Markov models based on directed graphical models, which can be biased towards states with few successor states[20].

As described above CRFs are probabilistic model, it computes the probability of P(X|Y) of possible output $Y=\{Y_1\ldots\ldots\ldots Y_n\}\varepsilon Y^n$ for the given input $X=\{X_1\ldots\ldots.X_n\}\varepsilon X^n$ [21].In case of text chunking Y is related with sequence of chunk tag and X is related with a sequence of word. Linear chain CRFs is a special form of CRFs, which is structured as a linear chain that models the output variables as a sequence. It can be shown using the following mathematical formula [21] .

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \frac{1}{Z_{\vec{\lambda}}(\vec{X})}.exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i\left(y_{j-1}, y_j, \overrightarrow{X}, j\right)\right) \quad (1)$$

From the above equation 1, n indicates the length of the sentences, m indicates the number of feature templates, j indicates the position of the input sequence, λ_i indicates the weights assigned to the different features in the training phase, $Z_{\vec{\lambda}}(\vec{X})$ is normalization factor that make the probability in the range [0,1], which can be expressed as[21] :

$$Z_{\vec{\lambda}}(\vec{X}) = \sum_{\vec{y}\in Y} exp\left(\sum_{j=1}^{n}\sum_{i=0}^{m}\lambda_i f_i\left(y_{j-1}, y_j, \vec{X}, j\right)\right) \quad (2)$$

Feature Functions

Selecting the best feature related to the domain play a vital role for the implementation of CRFs with good system performance. Feature specifies which information is considered from the labeled corpus during the training stage.In our case we use the word itself, POS tag, IOB chunk specification and by combining each other to build our feature set. we adopt a special form of CRFs (i.e. linear chain CRFs), the feature function for linear chain CRFs can be expressed as F_i

(Y_{j-1}, Y_j, X, j) where, Y_j-1, Y_j are adjacent class label and $\vec{X}$ is sequence of input words. we have described more in detail about the feature set we use on chapter four.

2.2.2.1.2 Memory-based learning in NLP

Memory-based language processing (MBLP) is a machine-learning method based on the notion that examples can be re-used directly in processing NLP problems. It consists two components: the learning component and the performance or processing component. During training phase, the learning component of MBLP stores examples in memory without abstraction, selection, or restructuring due to these it sometimes called as lazy learning. In the performance or processing component of an MBLP system the stored examples are used as a basis for mapping input to output; input instances are classified by assigning them an output label. During the time of classification, previously unseen test instance is presented to the system. The class of this instance is determined on the basis of an extrapolation from the most similar example(s) in memory[14]. The similarity between the new instance X and all examples Y in memory is computed using a distance metric $\Delta(X;Y)$.The general architecture of memory-based learning system adopted from [22] is shown below.

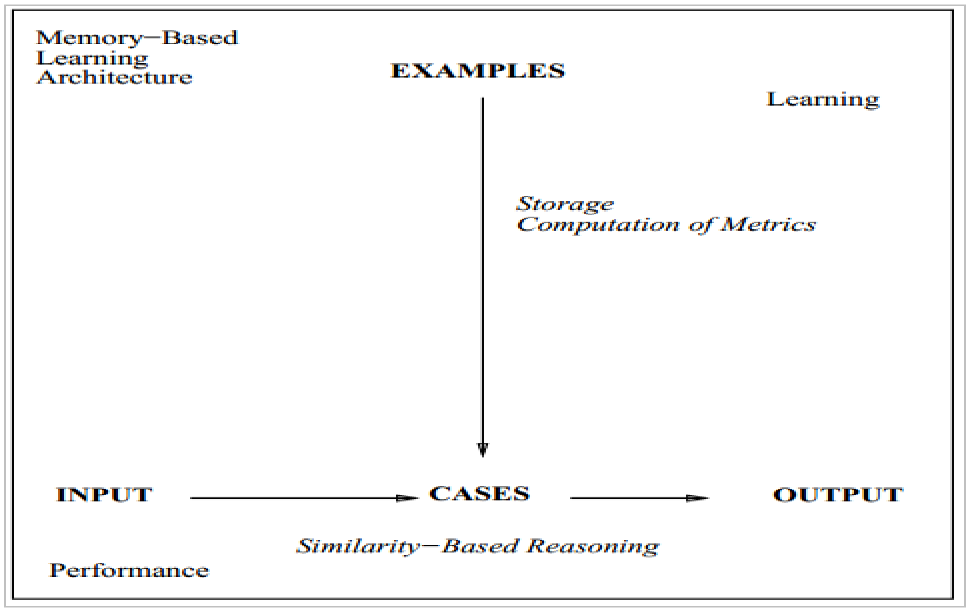

Figure 2. 1 General architecture of an MBL system(Adopted from [22]**)**

Memory-based learning system also known as example-based learning, lazy learning, instance-based learning, It is symbolic machine learning method which inherits the advantage of both probabilistic and knowledge-based method, when it adopted to NLP task [14]. Since many NLP problems have constraints of a large number of examples and many attributes with different relevance, memory-based learning uses more complex data structure and different speedup optimization which inherited from the k-NN [23];So it produces excellent result even in the presence of huge feature space.

As described in chapter one, memory-based learning implemented using free open source software package called Tilburg Memory-based Learner (TiMBL)[1]. This software package is developed and maintained by Walter Daelemans, Antal van den Bosch and their colleagues at Tilburg University and the University of Antwerp. To implement Tilburg Memory-based Learner (TiMBL) different alternative were possible: it can run as command line classifier or as server and the other is using C++ API that can integrate memory-based learning functionality into other programs like python. An illustration about memory-based learning system is shown below in figure 2.2. The encoder "γ" maps an input from the input space “X” into a set of addresses and the decoder "β" maps the set of activated memory locations into an output in the output space “Y”. The look-up table for MBL systems can be organized as hash tables, trees or full-search tables [24].

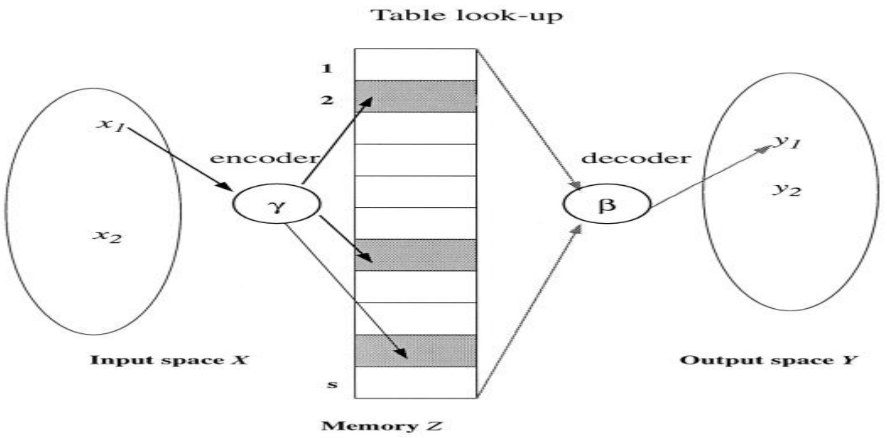

Figure 2. 2 An Example of Memory-Based Learning System(adopted from[24]**)**

As describe earlier Tilburg Memory-based Learner (TiMBL) is the software package which implements different memory-based algorithms using various similarity measure matrix. Some

[1] *The software and instructions to install Available at http://ilk.uvt.nl/timbl*

of the algorithm incorporated in TiMBL are: IGTree, IBl, IBl-IG, TRIBl, TRIBL2, IB2, C4.5 and the similarity measure matrix are Overlap, Modified value difference, Jeffrey divergence Dot-product and cosine matrix. In addition to the algorithm and the similarity measure matrix there is also feature weighting mechanism incorporated like Gain Ratio, Information Gain, Chi-squared, shared variance are some of the weighting techniques. All implemented algorithms have in common that they store some representation of the training set explicitly in memory. During testing, new cases are classified by extrapolation from the most similar stored cases. The main differences among the algorithms incorporated in TiMBL lie in: The definition of similarity, the way the instances are stored in memory, and the way the search through memory is conducted [22].In this research we implement IB1 and IGTree algorithm using overlap and Modified value difference similarity measures.

Some of the advantage of memory-based learning over the other machine learning approach are: its incremental training nature with no cost or little cost, in that the addition of new experience immediately affects processing without any need of re-computation of knowledge structures. The other is Forgetting training examples is a common trait of many machine learning algorithms but memory-based learning is best to never forget training examples [25] and the last is its language independent nature so it is possible to apply for any language. due to this feature of memory-based learning we are motivated to investigate for Amharic text chunker using them.

The main disadvantage of memory-based learning compared to most rival approaches is its slow classification speed. Its worst-case complexity of classification is O(nf), where n is the number of memorized examples, and f is the number of features; each new example needs to be compared against all of the memorized examples, each time involving a comparison of all features [25].

2.2.2.2 Unsupervised Approach

Unsupervised Learning systems reduce the burden of the user which spent in annotating training data. It requires only a statement of the required information. No chunking patterns are given in advance by the user. In this learning an annotated corpus is not used to improve the system's level of performance. The main challenge is to realize the user's need into a set of the chunking patterns. The systems are based on bootstrapping methods, expanding an initial small set of chunking patterns.

Like rule-based approach machine learning also has its own advantage and disadvantage. The advantage of machine learning approach is that it can be transferred to a different domain easily as long as specific texts and a person who can annotate them are available. But sometimes those texts are problematic or expensive to obtain or there is a lack of useful documents on which an algorithm can learn, and manual (or even machine-aided) annotation on the scale needed to provide reasonable levels of performance may be expensive.

2.2.3 Hybrid approach

In hybrid approach the combination of machine learning and rule-based approach were implemented. So, it inherits the advantage of both rule-based and machine-learning approach. During implementation of this kind of approach different alternative are possible. In some case, firstly rule based approach were implemented then for handling some exceptional case machine learning approach were implemented. for instance the paper [23] use hybrid approach for Korean text chunking, initially, they implemented rule-based approach for chunking Korean text and for exceptional case to the rule they were adopted memory-based machine learning approach to handle exceptions. In the other way, firstly machine-leaning approach implemented then for exceptional case to the machine-learning approach error pruning rule were applied to correct the error which is incorrectly chunked by machine-learning approach. For instance the paper [9] use hybrid method for Amharic base phrase chunking. Initially they implement HMM model for chunking Amharic text and for some error which is incorrectely chunked by HMM they use error pruning rule to correct the error.

2.3 Text Chunking specification

There are different types of chunk tags and chunk boundary identification techniques. The chunk tag can be noun phrase, verb phrase, adverb phrase etc. as of the target natural language construction rule. In order to identify the boundaries of each chunk in sentences, there are seven boundary types. These are IOB1, IOB2, IOE1, IOE2, IO, [and] [26]. The first four formats are complete chunk representation which can identify the beginning and ending of phrases. All use I tag for words that are inside a phrase and an O tag for words that are outside a phrase. They differ in their treatment of chunk-initial and chunk-final words.

- IOB1- the first word inside a phrase immediately following another phrase receives a B tag.
- IOB2- all phrases- initial words receive a B tag
- IOE1- the final word inside a phrase immediately preceding another same phrase type receives an E tag
- IOE2- all phrases- final words receive an E tag

Whereas, the last three are partial chunk representation

- IO- words inside a phrase receive an I tag, others receive an O tag
- [all phrase-initial words receive [tag other words receive tag
-] all phrase-final words receive] tag and other words receive Tag

2.4 Related works

As described earlier, Text chunker for the different language of the world has been done using different approaches. Here we are going to discuss the tool, techniques and experimental results of different scholar works related to text chunker.

2.4.1 Text chunker for the Amharic language

As indicated on chapter one only one work has been done for Amharic text chunker by Abeba Ibrahim [1]. The main objective of the research was to develop Amharic base phrase text chunker and transforming the developed TC into the full parser. The researchers were applied Hidden Markov Model (HMM) to develop the text chunker, in addition to HMM the researchers develop some rule to correct chunk which is incorrectly chunked by HMM. To transform the chunker to full parser, bottom-up approach with transformation algorithm was applied. For phrases boundary identification, the author used Inside Outside Beginning (IOB) chunk specification method.

In this research [1], 320 diverse kinds of sample sentences were collected from Amharic grammar books and news of Walta Information Center (WIC) for the training and testing dataset. Among 320 sentences, 288 sentences were used to train the system while the rest 32 sentences were used to test the system. Since the sentences collected from Amharic grammar books are not tagged, they were analyzed and tagged manually. The entire data set was also chunk tagged manually for the training set. However, tag sequence confilict were the main problem which is faced and finally they recommended to investigate further for future scholars. In addition to this, the size of the corpus is small. as indicated by [4], [8] the model they use (i.e. HMM) has label bias problem and also they were minimized the error mis-chunked by HMM model using error pruning rule, however major errors that are not pruned by rules are the tag sequence conflict.

2.4.2 Text chunker for Korean language

S.-B. and B.-T. Z. Park [23], the main objective of the research was to investigate Korean text chunker using hybrid approaches (i.e. rule based and memory based learning).Initially, they implement rule based method to chunk Korean text and for exceptional case to the rule they adopt memory based learning to handle the exceptional case to the rule. The information for predicting the chunk tag of a word were lexicons, POS tag and chunk tag of the surrounding word. Specifically, to determine the lexicons and POS tag three words of left and three word of right were used while for chunk tag prediction two words of left were analyzed. The performance of the chunker were using only rules gives 97.99% of accuracy and 91.87 f-score. However, when applied hybrid method the F-score was 94.21 which is improvement of 2.34 over the rule based only.

2.4.3 Text chunker for Indian language

K. Sarkar and V. Gayen [4], the paper discuss about Bengali noun phrase chunking using conditional random fields (CRFs).To apply CRFs selection of best feature related with the domain plays vital role for getting good result; to achieve these the researchers tried different features of Bengali language like preceding word and next to current words, POS tag of preceding word and POS tag of next to current word, word suffix, word length, etc. finally after extensive simulation based on variety features of the language using ICON 2013 dataset they get inspiring F-score results of 95.92,which improve the base line F-score results of 90.10.

W. Ali, M. K. Malik, S. Hussain, S. Shahid, and A. Ali [11],the main objective of the paper was to investigate and develop noun phrase chunker for Urdu language. To achieve this objective, they applied HMM model for automatic NP chunker of Urdu text. To enhance the accuracy of the system the researcher conducted four experiments using different features of the language. The feature they were used IOB tagset, POS tag, extended tagset and scanning direction both left to right and right to left directions. Finally, an accuracy of 97.61% was scored, which is the best accuracy from the four experiment, where the researchers get this accuracy by the feature using extended tagset and right to left scanning.

C. T. Rekha Raj and P. C. Reghu Raj [15], The main goal of the paper was to develop a text chunker for Malayalam by adopting Memory Based Learning (MBL).The authors used the package called Tilburg Memory-Based Learner (TiMBL),specifically for their experiment they use Memory Based Tagger (MBT) which is one component of TiMBL. The MBT was train using the feature words and POS tag of word, chunk Tag with IOB chunk specification. Finally, the model scores an accuracy of 97.14%.

S. Nm, R. K, and K. Shiva Kumar [27], They propose POS tagger and shallow parser(text chunking) for Kannada language using conditional random fields (CRFs). like other language the feature they used for chunking were POS tag, word and IOB chunk notation. Totally 3,000 sentences of Kannada corpus were used for the experiment, form these 2,500 sentences were used for training the model and the rest 5,00 were used for testing the model. They got for both POS tagging and chunking an accuracy of 96.86%.

2.4.4 Text chunker for Arabic language

N. Khoufi, C. Aloulou, and L. H. Belguith [2], The paper discus about chunking Arabic text by adopting conditional random fields(CRFs).They were used Penn Arabic Treebank as a dataset for their experiment and as feature they were used two preceding word and two next word from the current word, two preceding POS tag and two next POS tag of word from the current word, and IOB chunk boundary identification. They demonstrate using 12,624 sentences from these

10,100 sentences (80%) were used for model building and the rest 2,524 sentences (20%) is used for testing the model. their model scores an overall accuracy of 96.54%.

2.4.5 Text chunker for Chines language

Y. Zhao and T. Zhao [8],The objective of the paper was to enhance Chinese functional chunk parsing using clause boundary information. Initially, they develop a rule-based scheme to automatically identify the boundary of the clause by using the punctuation and predicate information, then, they were added this information as new feature to their model (i.e. CRFs model). In addition to clause boundary they also added POS tag as new feature to CRFs model. Finally, after conducting their experiment the result shows that clause boundary information is so effective to improve Chinese functional chunk parsing and also it solves the problem unable to parse long functional chunks.

2.5 Summary

In this chapter we discussed the three basic approaches of text chunking: rule-based, machine-learning, and hybrid approach. Since we are going to adopt the two machine-learning algorithms (CRFs and MBL), so we have explained the basic notion about them and the main reason we select the two machine-learning from other are explained in detail. finally, related work which have been done before are discussed.

Chapter three: Basic Amharic grammar structure

3.1 Introduction

Amharic is the official working language of the government of Ethiopia. It ranked as the second by the number of the speaker under Semitic language in the world next to Arabic [16].Any language has its own grammatical structure for effective communication between communicating entity. In the following sub-section, we are going to discuss the Amharic word classes, phrase types and sentence type and their structure.

3.2 Basic Amharic Word Classes

The basic word category of Amharic was studied and classified by a different scholar in different time. According to the early scholar Mersehazen [28], the basic word category of Amharic was divided into eight group i.e. nouns, pronouns, verbs, adjectives, prepositions, interjections, adverbs and conjunctions. However, the recent scholar Baye [29] reduce from eight to five i.e. nouns, verbs, adverbs, adjectives and prepositions. As indicated by Baye Yimam, the main reason reduced from eight to five were: pronoun, which have been one of the eight categories are considered as noun, because pronoun can replace noun. Conjunction have the same function and property as preposition so that conjunction grouped under preposition word class. Interjection word class in the earlier scholar are not considered as word class categories in the recent because they are word without any syntactic category.

In this work, we adopt the classification scheme of early scholars but pronouns and nouns are included in the same class as Baye[29]. The main reason we follow such kind of classification scheme is due to our dataset Walta information Center (WIC) corpora, which used for our experimentation treat pronouns and nouns as one class and the rest follow early scholar classification scheme. The discussions made in this chapter are based on the information took out from Baye [29],Mersehazen [28],Abeba [1], Abdurehman [30].

Noun word class

Amharic Noun are word or group of words used as name of class of people, place, thing, etc. The plural form of nouns in Amharic language take the suffix -ኦች "och". Amharic noun can be primitive or derived. a noun which is not derived from other word categories is primitive noun whereas a noun which is derived from other word categories is a derived noun.

Verb word class

verbs in Amharic shows action, occurrence, state of existence in a sentence. Most of the time take the end position of the sentence. It uses suffixes like *-ሁ “-hu”, -ህ “-h”, ሽ “-c”* as subject marker for I, you, she respectively. For instance, the word *መጣ* **“**come**”**, *እኔ መጣ-ሁ “*I have come*”, አንተ መጣ-ህ* **“**you come**”, *አንቺ መጣ-ሽ*** **“**she has come**”**. Similar to nouns, verbs can also be derived from other word categories by affixing different morphemes and form compound words of stems with verbs.

Adjective word

Amharic adjective is a word that describes or qualifies a noun or pronoun. Most of the time adjective comes before pronoun or noun but all the words that come before nouns cannot always be adjective.

For example: - ይህ ልጅ “this child”

In the above example ይህ “this” which precede noun ልጅ “children”, in this context even if ይህ “this” precede the noun ልጅ “children” its word category of ይህ “this” is pronoun rather than adjective. Like noun adjective can be primitive or derived. For instance, adjective can be derived from verbal root by infixing Vowels ኧ፣ኢ፣ኡ፣ አ between consonants.

Adverb

In Amharic, adverb is a word that modifies a verb, an adjective, another adverb, or a sentence. It refers to place, time, cause, degree or circumstances, etc., of the action mentioned by the verb.it answers questions like እንዴት "how", መቼ "when", የት "where". Adverb can be either in their primitive form or as their derivative form. The primitive adverbs are very few in number that are: “ገና” (yet), “ክፉኛ” (severely), “ቶሎ” (quickly), “ጅልኛ” (foolish), etc.

Preposition word class

A type of word that usually precedes a noun or pronoun and expressing a relation to another word or element is said to be preposition. The unique property of preposition is: they never use affixes and they don’t help to form new word from it. Usually, preposition precede noun and it acts like adverb to show ጊዜን “time”, ቦታን “place”, ምክናያትን “reason” and so on. Like adverbs, prepositions also few in number and these are: ስለ "for", እንደ "like", ከ" from", ወደ "to", አጠገብ "near to", ማዶ "there", ወዲህ "here". Preposition by itself has no meaning but when it combines by other word like a noun or pronoun it gets meaning. Some of preposition which use as affix and suffix are ስለ "for", እንደ "like", ወደ "to", እስከ "up to", ከ "form" and አጠገብ "near to", ማዶ "here", respectively.

Conjunction word class

A word that is used to link words, phrases, clauses or sentences is said to be a conjunction**.** "ነገር ግን" (but) "ወይም" (or) and "እና" (and) are some examples of Amharic conjunctions .For example, in the sentence "አበበ እና ከበደ ወደ አዲስ አበባ ሊመጡ ነው" "Abebe and Kebede are going to come to Addis Abeba" the conjunction "እና" connects the two nouns Abebe and Kebede. There are two types of conjunctions: coordinating conjunction and subordinating conjunction. Coordinating conjunctions are used to connect two words, phrases, clauses or sentence which are equal, whereas subordinating conjunctions connect two parts of words that are not equal (i.e., dependent and independent clauses). Subordinating conjunctions introduce dependent clause(s) and indicate the nature of the relationship among the independent clause(s) and the dependent clause(s).

3.3 Amharic Phrasal categories

The Phrase is a group of the word, which make sense but not complete sense. It is a syntactic structure that is wider than a word and smaller than a sentence. The Phrase can be constructed only from a head word or a head word combined with other words or phrases. When a phrase is constructed from a head word and other words or phrases, the other words or phrases can be specifiers, modifiers or complements. Specifiers are words used to specify the identity, location, number, possession, etc. of the head word. Specifiers can be either primitive or derived. For example, in the phrase "የካሳ መጽሀፍ" (Kassa's book) the specifier is "የካሳ" (Kassa's) that shows the owner of the book.

Modifiers on the other hand are used to indicate the amount, time, place, type, etc. of the head word or phrase. Adjectival phrase, noun phrase, prepositional phrase or sentences can be considered as modifiers.

For example, in the phrase “ቀይ ድመት” (red cat), the word “ቀይ” (red) indicates what type of color the car has. Unlike specifiers and modifiers complements are words or phrases that are used to make the ideas complete. For example in the sentences “ዳቦ በላሁ” (I ate bread) and “የስንዴ ዳቦ በላሁ” (I ate wheat bread), the first sentence does not give complete information about the bread but in the second sentence “የስንዴ” (wheat) is a complement that indicates from what the bread is made.

The types of the phrases for a language are determined by the categories of words in a language[29].Therefore, since there are five word categories in Amharic language, there are five phrase classes which are noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), adverbial phrase (AdvP), and prepositional phrase (PP) [29].

Noun phrase

A noun phrase (NP) is a phrase which contains noun or pronoun as its head and other constituents. It is used to refer objects, places, concepts, events or qualities. NP can be simple or complex. In Simple NP their head is a single noun or pronoun while as in complex NP their head is a noun with other constituents (i.e. specifiers, modifiers and complements). ድመት “cat” እኔ “me”, እሷ “she”, እሱ “he” and የምግብ ቤት “restaurant house” , የወርቅ ቀለበት “Gold ring” are example of simple noun phrases and complex noun phrases respectively.

Verb phrase

A verb phrase (VP) is a phrase which contain verb as its head and other constituents like complements, modifiers or specifiers. Based on the complement the verb take in the verb phrase can be categorized into transitive and intransitive. Transitive verbs take noun phrases as their complement whereas intransitive verbs do not; instead they take prepositional phrase.

For example, in the VPs “አለሙ ወንበሩን ሰበረዉ” (alemu broke the chair) and “ከበደ በሬውን ጎተተው” (kebede pull the ox), the complements of the head verb are transitive NPs. Therefore, the verbs “ሰበረው” (broke) and “ጎተተው”” (pull) are transitive verbs. The verbs in the VPs “ወደ መርካቶ ሄደ” (he went to merkato) and “ወደ ቤቱ ገባ” (he entered to his home) on the other hand are examples of intransitive verbs because the complement is prepositional phrase. Like noun phrase (NP), verb phrase (VP) can be classified into simple and complex. Simple VP contain only single verb as its head while as complex VP contains more than one verb as its head.

Adverbial phrase

Adverbial phrases (AdvPs) are build using one or more adverb as head and other constitutes. The unique feature unlike from other phrases AdvPs do not take complements as their constituents. Most of the time, the modifiers of adverbial phrases are prepositional phrases that come before adverbs.

Example: - "አበበ [እንደ ወንድሙ ገና] አልሄደም" (Yet Abebe doesn't go like her brother),

Here the adverbial phrase is "እንደ ወንድሙ ገና" and the head word is ገና "yet". The modifier that is found in the AdvP is *እንደ ወንድሙ* (like his brother) which is comparative Preposition.

Adjectival phrases

Adjectival phrases (AdjP), like noun phrases and verb phrases, are composed of an adjective as a head and other constituent such as complements, modifiers and specifiers. For example, in the sentence "ከበደ እንደ እናቱ በጣም ታመመ" (Kebede is very sick like his mother) the adjectival phrase is "እንደ እናቱ በጣም ታመመ" (very sick like his mother). In this phrase the modifier is "እንደ እናቱ" (like his sister) and the specifier is "በጣም" (very). AdjP can be simple or complex like that of VPs and NPs. Examples like the above are said to be simple AdjPs whereas AdjPs which contain embedded sentences are complex AdjPs.

Prepositional Phrase

Prepositional phrases (PPs) are made up of a preposition head and other constituents like nouns, noun phrases, verbs and verb phrases. Unlike other types of phrases, preposition cannot be a phrase by itself unless it is combined with other constituents. Noun or noun phrase constituents come after the head preposition whereas when the complement is prepositional phrase the head preposition appears on the right-hand side of the phrase.

Example. "ወደ ትልቁ ትምህርት ቤት" (to the large school)

"ወንበሩ ከጠረጴዛው አጠገብ" (the chair near to the table).

3.4 Amharic Sentence types

In Amharic grammar,a sentence can be formed by combining a group of phrases together which conveying a statement, question, exclamation, or command and typically containing a subject and predicate. The phrase can be simple or complex NP and simple or complex VP but NP always comes first as a subject. For example, when we see the following phrases: “ሌባው ልጅ” (the thief child) - Noun phrase, “የተማሪውን ኮምፒውተር” (student’s computer) – Noun phrase and” ወሰደው” (he take)-Verb phrase, they do not covey full meaning when they are spoken in separate, instead they raise questions like what did? who did? and what has done? However, when these phrases are combined together they form a sentence “ሌባው ልጅ የተማሪውን ኮምፒውተር ወሰደው”) and answers all the above questions. The structure of a sentence can be either simple or complex based on the number of verbs it contains.

Simple Sentences

In Amharic language, simple sentences are a sentence which constructs using NPs followed by VPs, the verb phrase contain only one verb. Consider the following sentence each of them has single verb indicated by the bracket.

- “አበበ ከተማ [ሄደ]” (Abebe went to city)
- “እሱ [መታኝ]” (he is clever)
- “አበበ ቁርሱን [በላ]” (Abebe eat his breakfast)

According to Baye Yimam [42] simple sentences are classified into four based on the purpose for which they are spoken. These are declarative, interrogative, negative, and imperative sentences. The structure of a sentence is also determined by its type.

Declarative sentences

Declarative sentences have the nature of making a declaration or they are just simple statements. It used to convey ideas and feelings of the speaker about things. In Amharic, declarative sentences always end with the Amharic punctuation mark አራት ነጥብ (“::”) which is equivalent to full-stop (.) in the English language. For example, sentences like “በሬው ዛሬ ሞተ” (today the ox died), “አስቴር አስተማሪ ሆነች” (Aster became a teacher) are some types of declarative sentences.

Negative sentences

These are sentences that make declarative sentences negative in meaning. They simply negate a declarative sentence. For example, the sentence “ካሳ ወደ ትምህርት ቤት አልሄደም” (Kassa didn’t go to school) is negative because the prefix “አል” in the verb “አል-ሄደም” makes the sentence negative declarative sentence.

Interrogative Sentences

A sentence that asks about the subject, complement, or the action specified by the verb is said to be an interrogative sentence. The question can be the one that asks about known things to be sure or the one that asks the unknown thing. An interrogative sentence always ends with a question mark (?). For example, in the sentence “ደመወዝክን ወሰድክ?” (did you take your salary?), the inquirer knows that the person will take his salary but he doesn’t know whether he took or not. However, if the inquirer asks questions like “የማን መኪና ናት?” (who is the owner of the car?), he is asking for unknown thing (i.e., the owner) so that he uses the pronoun “ማን” (who). Therefore, in order to construct such interrogative sentences, the inquirer usually uses interrogative pronouns like “ማን” (who), “ምን” (what), “የት” (where), “ስንት” (how many), and “መቼ” (when).

Imperative sentences

Imperative sentences are used to convey commands or instructions. Most of the time, the subject (that is second person pronoun) of the sentence is omitted but since Amharic words are highly inflected, subject marker prefixes indicate the specific subject. However, sometimes when the command is passed for a third person, the subject of the sentence can be third person pronoun or noun. For example, in the sentence “ደብተር ስጠኝ!” (bring exercise book!), the subject is “አንተ” (you) which is second person Masculine singular).

Complex Sentences

Complex sentences are formed from either complex noun phrase or complex verb phrase or both. In other words, a complex sentence can have a complex NP and a simple VP, a simple NP and a complex VP or both complex NP and complex VP. Complex NPs contain at least one embedded sentence which can be complement or other type phrase. On the other hand, complex VPs contain at least one sentence or more than one verb.

Chapter four: Amharic text chunker model

4.1 Introduction

In this chapter, we are going to discuss about the overall architecture of the system, dataset preparation which we used as an input for our model training and testing. In the first section we discuss the dataset preparation techniques we follow to prepare the data and in the second section about the overall architectural design of the model.

In the previous two chapters, we describe basic structure of Amharic language (i.e. word class, phrase category and sentence structure) to be taken into consideration before designing the model of Amharic text chunker and related work on text chunking techniques for different languages. This chapter gives a detail description of the model designed for this thesis work. As discussed in the previous chapter, supervised machine learning is selected for this study and the procedures to develop and design the model is explained in the following sub-section.

4.2 Dataset preparation and description

As indicated on the previous chapter we use machine-leaning to design Amharic text chunker. Mostly, this approach need syntactically annotated corpus. Using this approach, the success or failure rate of different NLP applications depends on the quality and availability of appropriate dataset and also selecting optimal features of the target language. The term dataset in computational linguistic mostly called as corpus (plural corpora). Corpora can be categorized into annotated and unannotated corpora. Annotated corpora are a collection of text with some syntactic notation like POS tag, chunk tag, NER tag etc. While as unannotated corpora are a collection of large amounts of text without any syntactic notations. supervised machine learning often use annotated corpora while as unsupervised use unannotated corpora. Since our investigation follow supervised machine learning approach so we use annotated dataset.

Every text chunker which is implemented using machine-learning approach needs two types of dataset[31]: training and testing dataset. Training dataset used to train and create a model for machine learning components and after completing training the performance of the model is evaluated using the test dataset.

Initially, before going to prepare our dataset we have been tried to get previously annotated dataset by[1] ; the researchers were not able to provide us the needed data, after several attempt finally we gone to prepare our own dataset. In this thesis work, we have collected 450 Amharic sentences that has total of 4,020 tokens from WIC corpus, Amharic grammar book and magazines . The sentences which was collected from WIC corpus was tagged by POS tag but the other data which were collected from Amharic grammar book and magazines didn't tagged by POS tag due to that we manually tagged them and receive comment and suggestion on it from linguistic expert. Only POS tagging is not enough for training the model in addition to it, it needs chunk tag. To the best knowledge of the researchers, chunk tagged documents do not exist for Amharic. For this reason, sentences are chunk tagged manually by the researchers.

To sum up,31 tag set were used for POS tagging the corpus and 11 tag set were used for Chunk tagged the corpus (See Appendix 2 for POS tag set and Appendix 3 for Chunk tag set). These sample sentences were transcribed according to the Amharic ፊደል (Fidel or alphabet) Unicode standard (See Appendix 1). After completing the corpora preparation by the researchers' comment and suggestions received from a linguistic expert.

4.2.1 Chunking specification

To identify the chunks, it is necessary to find the positions where a chunk can end and a new chunk can begin. The POS tag assigned to every tokens is used to discover these positions. As discussed in chapter two, there are four kinds of complete chunk boundary representations namely IOB1, IOB2, IOE1 and IOE2 [32]. In this study, to identify the boundaries of each chunk in sentences the IOB2 tag set is used for chunk tagged annotated text. Here "I" is a token inside a chunk, "O" is a token outside a chunk and "B" is a token that exists at the beginning of the chunk. See the following example በቤንች-ማጂ ዞን 30ሺ ህዝብ ለመምረጥ ተመዘገበ "in benchmaj zone thirty thousand people register for voting" with the chunk representation in Table 4.1.

Table 4. 1 Chunk specification

Word	IOB2 chunk
በቤንች-ማጂ	B-NP
ዞን	I-NP
30ሺ	B-NP

ህዝብ	I-NP
ለመምረጥ	B-VP
ተመዘገበ	I-VP

4.3 Architecture of the system

Under this sub-section, the overall architecture of both CRFs and MBL based Amharic text chunker models are discussed. Both models have two phases the training phase and the testing phase. During the training phase the system accepts word, POS tag of the word, and chunk tag of the word. Likewise, in the test phase, the system accepts words with their POS tags. Finally, the system outputs appropriate chunk tag sequences against each POS tag using CRF and MBL model. Figure 4.1 and Figure 4.2 shows the architecture of the chunker using CRF and MBL respectively.

4.3.1 Architecture of CRFs based ATC

ATC is intended in a means that, first it learns properties and parameters associated with chunk tag from the training data. It then receives input word with their POS tag and predicts the possible chunk tags. The architecture has two processes: learning process and prediction process.

Training is handled by components in the learning process. Preprocessing is initially performed on the training corpus. The corpus is chunk tagged based on IOB2 chunk specification format. Then it is passed to the ATC encoder which identifies a token and its corresponding POS and chunk tag through encoding. The tokens and tag sequence generated by the ATC encoder is handled to the Feature Extractor. Feature Extractor extracts necessary features to identify chunk tag based on the generated token and tag sequence, the extracted features will then be used as an input to the Model Builder. After the above all fulfilled, the builder begins the model building procedure to generate a trained model. Generally, here in the learning process, based on the training corpus we have generated a model that used to predict chunk tag classes of testing.

Training phase

Training phase contains necessary components that are used in the process of training the machine learning component. The components within this phase operate on the data from the training corpus and they are the ATC Encoder, Feature Extractor and the Model Builder.

The ATC Encoder provides information to feature extractor by identifying a token and its tag. From the training data, the encoder analyzes which one is a token and which one is a tag then supplies to the next process. Hence, the encoder provides an important clue to feature extractor, setting it wrongly also leads to wrong model creation. To avoid this, we have kept the consistency between corpus and ATC encoder.

As we have discussed in chapter two, Features are properties of a text that are used to provide necessary information associated to a given chunk tags and increase the confidence level of predicting correct chunk tag of a word. ATC feature extractor is responsible for identifying and extracting all the necessary features from the training data. It is one of the essential components that supply necessary information (features) during the building of the model. The feature extractor is designed to extract features from the training data and store them for future use.

The main concern with the machine learning component is to build a trained model that will be used in prediction. The model builder is designed to generated a trained model that will be used in prediction phase[33].

Testing phase

The testing phase is the final phase in ATC system. It is a process of recognizing chunk tag of the given preprocessed tokens. The knowledge in model builder contain features that are extracted and stored during the training phase, are supplied to the recognizer to identify chunk tag from the given Amharic text. Identification of chunk tag is performed based on the calculated probability which is discussed on chapter two.

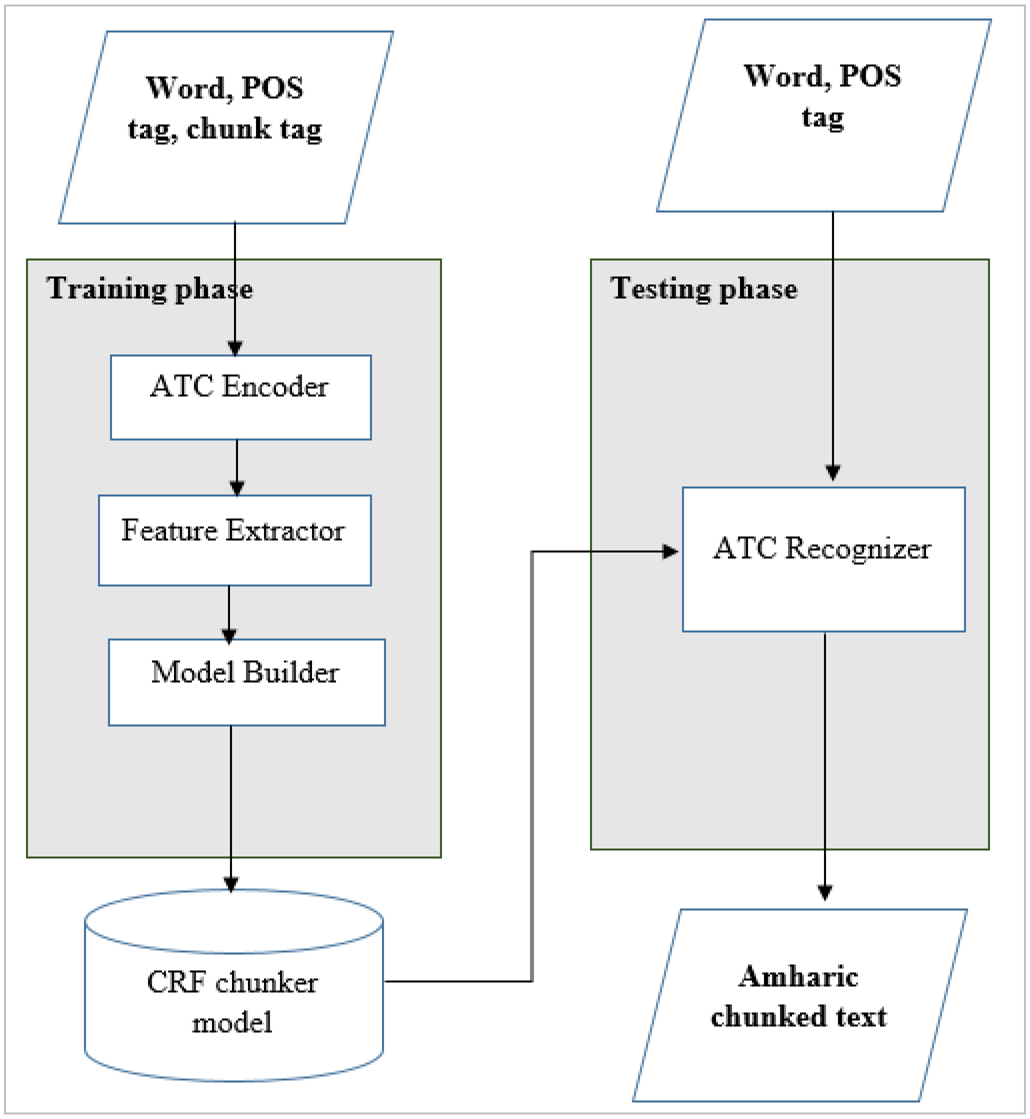

Figure 4. 1 Architecture of CRFs based ATC

4.3.2 Architecture of Memory-Based Learning ATC

Like CRFs, memory-based learning also has two phases. A training phase which consists of feature extraction to create instances in a fixed length of windows and the memory-based learning to train the dataset. On the other hand, the testing phase contains the feature extraction (instance making) to deconstruct a given text, classification and extrapolate to select appropriate chunk tag to a given token.

Train phase

The training phase is the backbone of the ATC module to successfully implement the system. The ATC is implemented by using the memory-based learning model. Therefore, in this phase, the feature extraction is used to make the input words to be suitable for memory-based learning. Once the annotated words are given to the system, instances are extracted automatically from the given dataset, based on the concept of windowing method in a fixed length of left and right context. Finally, MBL(TiMBL) built a model which used during predication phase

Test phase

All implemented algorithms have in common that they store some representation of the training set explicitly in memory. During the test, new cases are classified by extrapolation from the most similar stored cases. The main differences among the algorithms incorporated in TiMBL lie in the definition of similarity, the way the instances are stored in memory, and the way the search through memory is conducted. When new or unknown inflected words are deconstructed as instances and given to the system to be analyzed, an extrapolation is performed to assign the most likely neighborhood class with its morphemes based on their boundaries. The extrapolation is based on the similarity metric applied on the training data. If there is an exact match on the memory, the classifier returns (extrapolates) the class of that instance to the new instance. Otherwise,a new instance is classified by analogy in memory with a similar feature vector, and extrapolating a decision from their class. This instance is compared to each and every instance in the training set, recorded by the memory-based learner. In doing so, the classifier tries to find that training instance in memory that most closely resembles it.

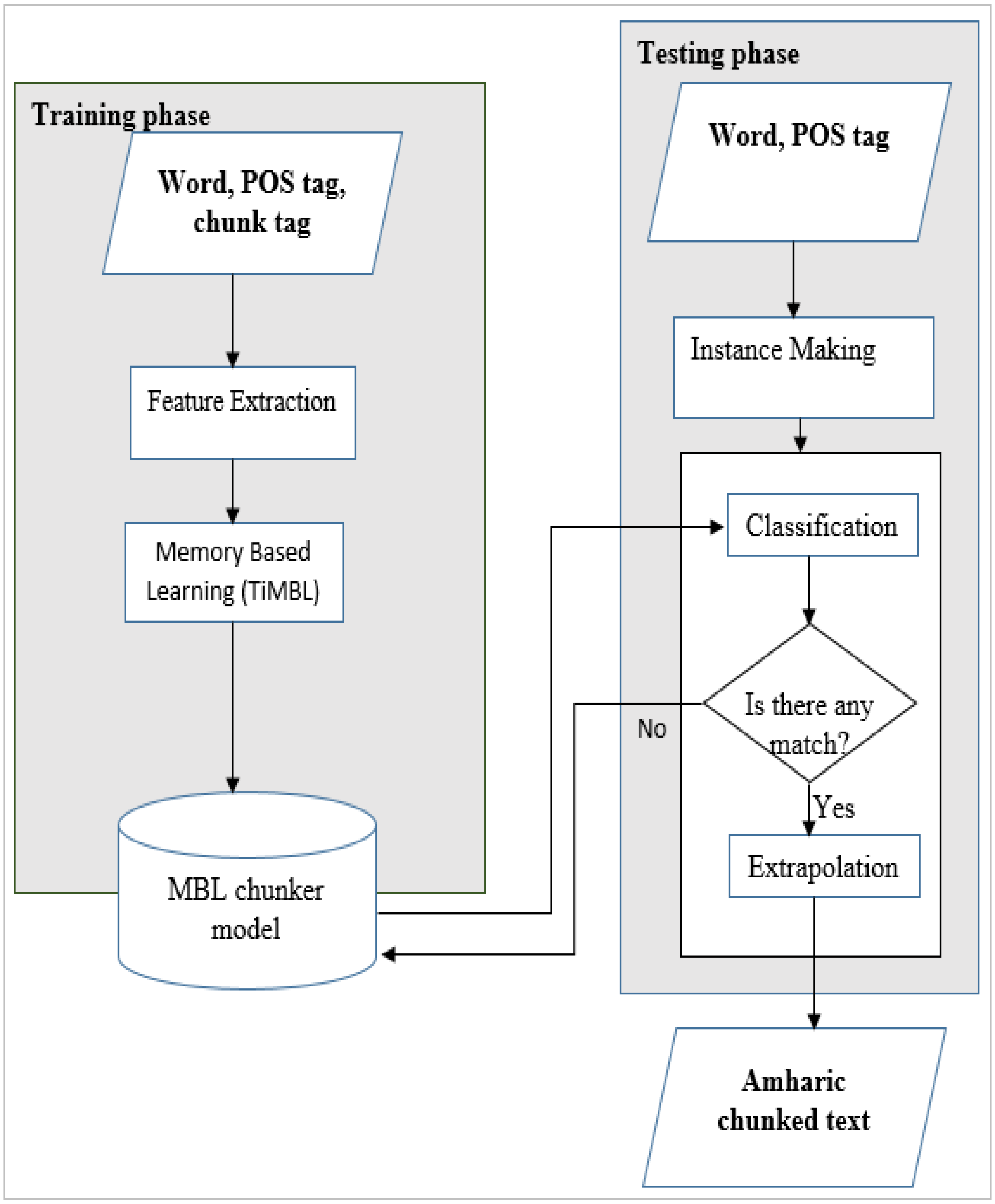

Figure 4. 2 Architecture of Memory-Based Learning ATC

Chapter five: Experiment and result discussion

5.1 Introduction

In chapter four, the design of Amharic text chunker with is implementation are discussed. In this chapter, we are going to present the process of the experiment conducted. Experiments are done in different scenarios so that the results are displayed accordingly with their explanations.

5.2 Feature representation and description

Feature selection plays a crucial role in CRFs framework[34]. Experiments were carried out to find out most suitable features for Amharic text chunking task. The main features for text chunking task have been identified based on the different possible combination of available word and tag context. In this study, it has been considered a different combination from the following set for inspecting the best feature set for the Amharic text chunker task. The various features used for developing our system are described below:

- Surrounding word: Preceding and following words of a current word can be used as a feature because surrounding words influence the current word. We have considered different window size of the word until we get a promising result.
- Combination of words: combinations such as the preceding/current word and current word/following word are used as feature.
- Window for POS of the current word: Preceding and following POSs of a present word are used as feature because surrounding POSs influence the present word.
- Combination of POSs: combination such as the preceding/current POS and current/following POS are used as feature.

Our empirical study found that the following combination of features gives the optimal feature set.

F(best)= [$W_{i-2}W_{i-1}W_iW_{i+1}W_{i+2}$, $POS_{i-2}POS_{i-l}POS_iPOS_{i+1}POS_{i+2}$]

5.3 Experiments

In this study, a total of 450 sentence were used and manually annotated with chunk tags. From the total 90 % used for model creation and the rest 10% used for testing the model. The detail of corpus statistics is shown in Table 5.1.

Table 5. 1 Corpus statistics

Number of tokens in the training set	Number of tokens in the testing set	Total number of tokens
3,618 tokens	402 tokens	4,020 tokens

To do the experiment, four different scenarios were considered. In the first scenario, one word left and one word to the right from the current word (w_{i-1} w_i w_{i+1}). The detail is shown in the bellow table 5.2.

Table 5. 2 Experiment one

Features used	Extracted Features	Time taken	Accuracy
w_{i-1} w_i w_{i+1}	41,556	3.33 s	84.57%

In experiment two, we used the previous experiment one features and add their corresponding POS tag of each tokens used as a feature.the accuracy increased by 9.95. the detail is shown in the below table 5.3.

Table 5. 3 Experiment two

Features used	Extracted Features	Time taken	Accuracy
$w_{i-1}w_i$ w_{i+1}, $POS_{i-1}POS_iPOS_{i+1}$	42,384	3.64 s	94.52%

In experiment three, two word to the left and two word from the right of the current word used as a feature.The detail is shown in the below table 5.4.

Table 5. 4 Experiment three

Features used	Extracted Features	Time taken	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	66,252	3.04 s	85.32%

In experiment four, we used the above experiment three and add the corresponding POS tag of each tokens as a feature. The detail is shown in the below table 5.4.

Table 5. 5 Experiment four

Features used	Extracted Features	Time taken	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	67,656	3.75 s	97.26%

Experimental result discussion

The above experiments show that different combination of features give different accuracy result. From our investigation using CRFs, we understand that when we increase the window size of the tokens and POS tag, the accuracy of the model predicting the correct chunk tag increased. The following table 5.6 presents the performance of the whole experiment.

Table 5. 6 Summary of CRFs based experiments

No.	Features used	Extracted Features	Time taken	Accuracy
1	w_{i-1} w_i w_{i+1}	41,556	3.33 s	84.57%
2	w_{i-1} w_i w_{i+1} &$POS_{i-1}POS_iPOS_{i+1}$	42,384	3.64 s	94.52%
3	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	66,252	3.04 s	85.32%
4	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	67,656	3.75 s	97.26%

Memory based text chunker implementation

Memory-based approaches borrow some of the advantages of both probabilistic and knowledge-based methods to successfully implement it in NLP tasks[25]. It performs classification by analogy. In order to learn any NLP classification problem, different algorithms and concepts are implemented by reusing data structures. We used TiMBL as a learning tool for our task[18].

There are a number of parameters to be tuned in memory-based learning using TiMBL. Therefore, to get an optimal accuracy of the model we used the default settings and also tuned some of the parameters. The optimized parameters are the MVDM (modified value difference metric) from distance metrics, no weighting from weighting metrics, and k from the nearest neighbor. These optimized parameters are used together with the different classifiers. The classifier engines we used are IGtree and IB1 which construct databases of instances in memory during the learning process. Instances are classified by IGTree or by IBI by matching them to all instances in the instance base. As a result of this process, we get a memory-based learning model which will be used later during the text chunking process.

Experiment with IB1

In the following sub-section, we conduct the experiment on memory-based Amahric text chunker. In experiment five, one word to the left and one word from the right of the current word used as a feature. The detail of the experiments is shown below table 5.7.

Table 5. 7 Experiment five

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
w_{i-1} w_i w_{i+1}	IB1	MVDM	No weighting	73.13%

In experiment six, one word to the left and one word from the right of the current word and add POS tag of each word as a feature. The detail of the experiment shown below table 5.8.

Table 5. 8 Experiment six

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
w_{i-1} w_i w_{i+1} &$POS_{i-1}POS_iPOS_{i+1}$	IB1	MVDM	No weighting	78.10%

In experiment seven, two words to the left and two words from the right of the current word used as a feature. The detail of the experiment shown below table 5.9.

Table 5. 9 Experiment seven

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	IB1	MVDM	No weighting	78.35%

In experiment eight, two word to the left and two word from the right of the current word and add the corresponding tokens of each tokens used as a feature. The detail of the experiment shown below table 5.10.

Table 5. 10 Experiment eight

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	IB1	MVDM	No weighting	82.08%

Table 5. 11 Summary of IB1 experements

No.	Features used	Algorithm	distance metrics	feature-weighting	Accuracy
Exp1	w_{i-1} w_i w_{i+1}	IB1	MVDM	No weighting	73.13%
Exp2	w_{i-1} w_i w_{i+1} &$POS_{i-1}POS_iPOS_{i+1}$	IB1	MVDM	No weighting	78.10%
Exp3	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	IB1	MVDM	No weighting	78.35%
Exp4	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	IB1	MVDM	No weighting	82.08%

In experiment nine, one word to the left and one word from the right of the current word used as a feature. The detail of the experiments shown below table 5.12.

Table 5. 12 Experiment nine

Features	Algorithm	distance metrics	feature-weighting	Accuracy
w_{i-1} w_i w_{i+1}	IGTREE	Overlap	Gain Ratio	71.38%

In experiment ten, one word to the left and one word from the right of the current word and add POS tag of each word as a feature. The detail of the experiment shown below table 5.13.

Table 5. 13 Experiment ten

Features used	Algorithm	Distance metrics	feature-weighting	Accuracy
w_{i-1} w_i w_{i+1} &$POS_{i-1}POS_iPOS_{i+1}$	IGTREE	Overlap	Gain Ratio	71.64%

In experiment eleven, two words to the left and two words from the right of the current word used as a feature. The detail of the experiment shown below table 5.14.

Table 5. 14 Experiment eleven

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	IGTREE	Overlap	Gain Ratio	72.88%

In experiment twelve, two words to the left and two words from the right of the current word and add the corresponding tokens of each token used as a feature. The detail of the experiment shown below table 5.15.

Table 5. 15 Experiment twelve

Features used	Algorithm	distance metrics	feature-weighting	Accuracy
W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	IGTREE	Overlap	Gain Ratio	72.88%

Experiments with IGTREE

Table 5. 16 Summary of IGTREE experiments

No.	Features used	Algorithm	distance metrics	feature-weighting	Accuracy
Exp1	w_{i-1} w_i w_{i+1}	IGTREE	Overlap	Gain Ratio	71.38%
Exp2	w_{i-1} w_i w_{i+1} &$POS_{i-1}POS_iPOS_{i+1}$	IGTREE	Overlap	Gain Ratio	71.64%
Exp3	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}	IGTREE	Overlap	Gain Ratio	72.88%
Exp4	W_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} &$POS_{i-2}POS_{i-1}POS_iPOS_{i+1}POS_{i+2}$	IGTREE	Overlap	Gain Ratio	72.88%

Chapter six: Conclusion and Recommendation

6.1 Conclusion

Text chunking is a process to identify non-recursive cores of various phrase types without conducting deep parsing of text. Many text chunking systems are developed for English and other European languages, next to these, text chunking for Arabic and South Asian languages have been developed. Amharic is one of the languages that have many speakers in Ethiopia, but it is under resourced language.

This work tried to contribute one important tool which plays a role for overcoming some challenges in NLP regarding to Amharic. Having a good Amharic text chunker could be used in information extraction of any domain specific tasks, question answering, information retrieval etc. to come up with a system that could be used in the above application areas, identifying the optimal feature set is the most important task of any text chunking task. This work is also delimited to identifying these optimal features set to recognize Amharic phrases which enhance the accuracy of existing Amharic text chunker.

This study has proposed and designed a system called Amharic text chunker. The system is designed based on a supervised machine learning component. We adopt the machine learning component using CRFs and MBL algorithm. To do this research, 450 sample sentence were taken from WIC corpus, Amharic grammar book,news article,magazines and chunk tagged manually. These datasets are classified into 90% training and the rest 10 % used for testing. By conducting different experiments, we have been identified optimal features for ATC, we have conducted four experiments using CRFs and eight experiments using MBL and the roles of each feature in different experiments were tested.

The highest accuracy achieved in this work using CRFs is 97.26%, with a window size of two on both sides, with their corresponding POS tag of each tokens. The worst performance achieved is 84.57%, with only the window size of one word on both left and right side. On MBL algorithm we implement IB1 and IGTREE classifiers. the best accuracy result scored using IB1 was 82.08% with a window size of two words on both sides, with their corresponding POS tag of each tokens and its worst result achieved 73.13% with a window size of one words on both sides. the best accuracy result scored using IGTREE was 72.88% with a window size of two words on both right and left sides, with their corresponding POS tag of each tokens and its worst accuracy result achieved 71.38% with a window size of one words on both left and right side.

The previous work on ATC has performed with an accuracy rate of 93.5 on a different tool, corpus size and feature set. Due to these reasons, we could not compare this work with the previous one. But from the finding we got, a tool achieved a promising result for Amharic text chunker especially CRFs perform best from MBL.

From the findings, two word left and two word right from the current words and the corresponding POS tags of each tokens are the observed optimal feature sets for recognizing Amharic text chunker. from the two machine-learning algorithm CRFs scored better accuracy than MBL.

6.2 Recommendation

This study showed some of the features that have never got the attention of previous ATC. The involvement of those new features has brought a promising performance and the following points are some of our observation that should be taken into consideration for future works of ATC**.**

- ❖ Replicate this work using large data and incorporating all types of sentences such as interrogative sentences and exclamatory sentences, and also it can be an extend work in other local languages.
- ❖ Developing a hybrid approach which incorporates a rule based on the nature and pattern of Amharic text chunker with CRF based approach is recommended since it might give a better performance.

References

[1] A. Ibrahim, "A Hybrid Approach to Amharic Base Phrase Chunking and Parsing," Addis Abeba University, 2013.

[2] N. Khoufi, C. Aloulou, and L. H. Belguith, "Chunking Arabic Texts Using Conditional Random," *IEEE*, pp. 428–432, 2014.

[3] A. Molina and F. Pla, "Shallow parsing using specialized HMMs," *J Mach. Learn. Res.*, vol. 2, pp. 595–613, 2002.

[4] K. Sarkar and V. Gayen, "Bengali Noun Phrase Chunking Based on Conditional Random Fields," *IEEE*, pp. 148–153, 2014.

[5] K. H. AMARE and A, "Tigrigna question answering system for factoid questions," Addis Abeba University, 2016.

[6] D. Abebaw, "LETEYEQ (ሌጠየቅ)-A Web Based Amharic Question Answering System for Factoid Questions Using Machine Learning Approach," Addis Abeba University, 2013.

[7] Muhe Seid, "TETEYEQ: Amharic Question Answering System for Factoid Questions," Addis Abeba University, 2009.

[8] Y. Zhao and T. Zhao, "Exploiting clause boundary information as features for Chinese functional chunk parsing," *IEEE*, pp. 874–878, 2016.

[9] A. Ibrahim and Y. Assabie, "Hierarchical Amharic Base Phrase Chunking Using HMM with Error Pruning," *Springer Int. Publ. Switz.*, vol. 8387, pp. 126–135, 2014.

[10] X. Vwhp, "shallow parsing natural language processing implementation for intelligent automatic customer service," *IEEE*, pp. 274–279, 2014.

[11] W. Ali, M. K. Malik, S. Hussain, S. Shahid, and A. Ali, "urdu noun phrase chunking," *IEEE*, pp. 494–497, 2010.

[12] D. Guleria and M. K. Chavan, "a study and comparative analysis of conditional random fields for intrusion," vol. 2, no. 4, pp. 31–38, 2012.

[13] A. M. and F. C. N. P. J. Lafferty, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. Eighteenth Int. Conf. Mach. Learn. (ICML 2001)*, pp. 282–289, 2001.

[14] D. Walter and den B. Antal van, *Memory-Based langauge processing*. Cambridge University Press, 2005.

[15] C. T. Rekha Raj and P. C. Reghu Raj, “Text chunker for Malayalam using Memory-Based Learning,” *IEEE*, pp. 595–599, 2015.

[16] “CSA (Central Statistics Agency), Addis Ababa, Ethiopia: Central Statistics Agency,” *http://www.csa.gov.et*, 2007. .

[17] T. Dawit, “Design of Amharic Anaphora Resolution Model,” Addis Abeba University, 2014.

[18] A. van den Bosch, W. Daelemans, *Memory-Based Language Processing.* Cambridge University Press, New York, 2005.

[19] U. Jain and J. Kaur, “A Review on Text Chunker for Punjabi Language,” vol. 4, no. 7, pp. 116–119, 2015.

[20] L. John, M. Andrew, and P. Fernando, “Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data.”

[21] K. Roman and T. Katrin, “Classical Probabilistic Models and Conditional Random Fields,” Dortmund, 2007.

[22] D. Walter, Z. Jakub, der S. Ko van, and den B. Antal van, “TiMBL : Tilburg Memory Based Learner Reference Guide,” 2010.

[23] S.-B. Park and B.-T. Zhang, “Text chunking by combining hand-crafted rules and memory-based learning,” in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, 2003, pp. 497–504.

[24] J. Lin and J. S. Vitter, “A Theory for Memory-Based Learning *,” *Brown Univ.*, pp. 143–167, 1994.

[25] W. Daelemans and A. Van Den Bosch, “Memory-Based Learning,” in *Blackwell Computational Linguistics and Natural Language Processing Handbook,* A. Clark, C. Fox, and S. Lappin, Eds. Blackwell, 2009, pp. 1–58.

[26] L. A. Ramshaw and M. P. Marcus, “Text chunking using transformationbased learning,” in *Proceedings of the Third ACL Workshop on Very Large Corpora*, 1995, pp. 82–94.

[27] S. Nm, R. K, and K. Shiva Kumar, “Part-Of-Speech Tagging And Parsing Of Kannada Text Using Conditional Random Fields (CRFs),” in *International Conference on Intelligent Computing and Control,* 2017.

[28] W. Mersehazen, የአማርኛ ሰዋሰው. Addis Ababa: Birhanena Selam Printing Press, 1934.

[29] Yimam Baye, “የአማርኛ ሰዋሰው,” EMPDA, Addis Ababa, 2002.

[30] A. D. Mohammed, “A TOP-DOWN CHART PARSER FOR AMHARIC SENTENCES,” Addis Abeba University, 2015.

[31] G. B. Kumar, “UCSG Shallow Parser : A Hybrid Architecture for a Wide Coverage Natural Language Parsing System,” 2007.

[32] Taku Kudo, “Machine Learning and Data Mining Approaches to to Practical Natural Language Processing,” Nara Institute of Science and Technology, 2003.

[33] M. J. Althobaiti, “Minimally-supervised Methods for Arabic Named Entity Recognition,” 2016.

[34] P. Abhijith, “Parts Of Speech (POS) Tagger for Kannada Using Conditional Random Fields (CRFs),” 2016.

List of Appendixes

Appendix 1. Amharic ፊደል (Fidel or alphabet) Unicode standard

	120	121	122	123	124	125	126	127	128	129	12A	12B
0	ሀ	ሐ	ሠ	ሰ	ቀ	ቐ	በ	ተ	ኀ	ነ	አ	ኰ
1	ሁ	ሑ	ሡ	ሱ	ቁ	ቑ	ቡ	ቱ	ኁ	ኑ	ኡ	▨
2	ሂ	ሒ	ሢ	ሲ	ቂ	ቒ	ቢ	ቲ	ኂ	ኒ	ኢ	ኲ
3	ሃ	ሓ	ሣ	ሳ	ቃ	ቓ	ባ	ታ	ኃ	ና	ኣ	ኳ
4	ሄ	ሔ	ሤ	ሴ	ቄ	ቔ	ቤ	ቴ	ኄ	ኔ	ኤ	ኴ
5	ህ	ሕ	ሥ	ስ	ቅ	ቕ	ብ	ት	ኅ	ን	እ	ኵ
6	ሆ	ሖ	ሦ	ሶ	ቆ	ቖ	ቦ	ቶ	ኆ	ኖ	ኦ	▨
7	ሇ	ሗ	ሧ	ሷ	ቇ	▨	ቧ	ቷ	ኇ	ኗ	ኧ	▨
8	ለ	መ	ረ	ሸ	ቈ	ቘ	ቨ	ቸ	ኈ	ኘ	ከ	ኸ
9	ሉ	ሙ	ሩ	ሹ	▨	▨	ቩ	ቹ	▨	ኙ	ኩ	ኹ
A	ሊ	ሚ	ሪ	ሺ	ቊ	ቚ	ቪ	ቺ	ኊ	ኚ	ኪ	ኺ
B	ላ	ማ	ራ	ሻ	ቋ	ቛ	ቫ	ቻ	ኋ	ኛ	ካ	ኻ
C	ሌ	ሜ	ሬ	ሼ	ቌ	ቜ	ቬ	ቼ	ኌ	ኜ	ኬ	ኼ
D	ል	ም	ር	ሽ	ቍ	ቝ	ቭ	ች	ኍ	ኝ	ክ	ኽ
E	ሎ	ሞ	ሮ	ሾ	▨	▨	ቮ	ቾ	▨	ኞ	ኮ	ኾ
F	ሏ	ሟ	ሯ	ሿ	▨	▨	ቯ	ቿ	▨	ኟ	ኯ	▨
	12c	12d	12e	12f	130	131	132	133	134	135	136	137
0	ዀ	ዐ	ዠ	ደ	ጀ	ጐ	ጠ	ጰ	ፀ	ፐ	፠	፰

1	▨	ዑ	ዡ	ዱ	ጁ	▨	ጡ	ጱ	ፁ	ፑ	፡	፱
2	ዂ	ዒ	ዢ	ዲ	ጂ	ጒ	ጢ	ጲ	ፂ	ፒ	።	፲
3	ዃ	ዓ	ዣ	ዳ	ጃ	ጓ	ጣ	ጳ	ፃ	ፓ	፣	፳
4	ዄ	ዔ	ዤ	ዴ	ጄ	ጔ	ጤ	ጴ	ፄ	ፔ	፤	፴
5	ዅ	ዕ	ዥ	ድ	ጅ	ጕ	ጥ	ጵ	ፅ	ፕ	፥	፵
6	▨	ዖ	ዦ	ዶ	ጆ	▨	ጦ	ጶ	ፆ	ፖ	፦	፶
7	▨	▨	ዧ	ዷ	ጇ	▨	ጧ	ጷ	ፇ	ፗ	፧	፷
8	ወ	ዘ	የ	ዸ	ገ	ጘ	ጨ	ጸ	ፈ	ፘ	፨	፸
9	ዉ	ዙ	ዩ	ዹ	ጉ	ጙ	ጩ	ጹ	ፉ	ፙ	፩	፹
A	ዊ	ዚ	ዪ	ዺ	ጊ	ጚ	ጪ	ጺ	ፊ	ፚ	፪	፺
B	ዋ	ዛ	ያ	ዻ	ጋ	ጛ	ጫ	ጻ	ፋ	▨	፫	፻
C	ዌ	ዜ	ዬ	ዼ	ጌ	ጜ	ጬ	ጼ	ፌ	▨	፬	፼
D	ው	ዝ	ይ	ዽ	ግ	ጝ	ጭ	ጽ	ፍ	▨	፭	▨
E	ዎ	ዞ	ዮ	ዾ	ጎ	ጞ	ጮ	ጾ	ፎ	▨	፮	▨
F	ዏ	ዟ	ዯ	ዿ	ጏ	ጟ	ጯ	ጿ	ፏ		፯	▨

Appendix 2: POS tags by WIC (used by this study)

NO	TAG	DESCRIPTION	EXAMPLE
1	ADJ	Adjective	ትንሹ
2	N	Noun	
3	VREL	Relative verb	
4	NUMCR	Cardinal Number	
5	V	Verb	
6	ENDPUNC	Sentence end punc	
7	NPrep	Noun with preposition	
8	VPrep	Verb with preposition	
9	NUMPrep	Number with preposition	
10	PREP	Preposition	
11	VN	Verbal noun	
12	ADJPrep	Adjective with preposition	
13	NC	Noun with conjunction	
14	ADV	Adverb	
15	PUNC	Punctuation	
16	NPC	Noun with preposition and conjunction	
17	AUX	Auxiliary verbs	
18	PRONP	Pronoun with preposition	
19	CONJ	Conjunction	

20	NUMOR	Ordinal Number	
21	VPC	Verb with preposition and conjunction	
22	PRON	Pronoun	
23	PRONPC	Pronoun with preposition and conjunction	
24	ADJC	Adjective with conjunction	
25	VC	Verb with conjunction	
26	PRONC	Pronoun with conjunction	
27	UNC	Unclear	
28	ADJPC	Adjective with preposition and conjunction	
29	INT	Interjection	
30	NUMC	Number with conjunction	
31	NUMPC	Number with preposition and conjunction	

Appendix 3: Chunk tags(used by this study)

No	Chunk tags	Description
1	B-NP	Beginning of noun phrase
2	I-NP	Inside noun phrase
3	B-VP	Beginning of verb phrase
4	I-VP	Inside of verb phrase
5	B-PP	Beginning of preposition phrase
6	I-NP	Inside of preposition phrase
7	B-ADJP	Beginning of adjective phrase
8	I-ADJP	Inside of adjective phrase
9	B-ADVP	Beginning of adverb phrase
10	I-ADVP	Inside of adverb phrase
11	O	Outside from any phrase construction

Appendix 4: Sample chunk tagged document for the training data set

በአዳማ NP B-NP

ልዩ ADJ I-NP

ዞን N I-NP

በተጠናቀቀው VP B-NP

የበጀት NP I-NP

አመት N I-NP

በ6.8ሚሊየን NUMP B-NP

ብር N I-NP

ከተጀመሩት VP B-NP

12 NUMCR I-NP

ፕሮጀክቶች N I-NP

መካከል PREP O

አብዛኞቹ ADJ B-VP

ተጠናቅቀው V I-VP

50ሺ NUMCR B-NP

ነዋሪዎችን N I-NP

ተጠቃሚ N B-VP

ማድረጋቸው VN I-VP

ተገለጸ V I-VP

። PUNC O

ኢትዮጵያ N B-NP

የተያያዘችው VREL B-VP

ዘርፈ N B-NP

ብዙ ADJ I-NP

የልማት NP B-NP

እንቅስቃሴ N I-NP

ሊሳካ V B-VP

የሚችለው VREL I-VP

ዜጎቿ N B-NP

በሚያበረክቱት VP I-NP

የፈጠራ NP B-NP

ስራዎች N I-NP

መሆኑን VN I-NP

የሳይንስና NPC B-NP

ቴክኖሎጂ N I-NP

ኮሚሽን N I-NP

ጊዜያዊ ADJ I-NP

ሃላፊ N I-NP

ገለጹ V B-VP

። PUNC O

በኦሮሚያ NP B-NP

ሶስት NUMCR I-NP

ዞኖች N I-NP

የኦህዴድ NP B-VP

ኮንፈረንስ N I-VP

ተጀመረ V I-VP

። PUNC O

የጋምቤላ NP B-NP

ክልል N I-NP

መምህራን N I-NP

ማህበር N I-NP

አስቸኳይ ADJ B-NP

ጉባኤ N I-NP

የማህበሩን NP B-NP

ፕሬዚዳንት N I-NP

ከሃላፊነት NP B-NP

በማስወገድ NP I-NP

ትናንት ADV B-VP

ተጠናቀቀ V I-VP

። PUNC O

Appendix 5: Sample chunked text output

```
የበጎ ADJP B-NP  B-NP
ፈቃደኛ N I-NP  I-NP
ደም N  I-NP I-NP
ለጋሾች N I-NP I-NP
ቁጥር N I-NP I-NP
እያደገ ADV B-VP B-VP
መሆኑ AUX I-VP I-VP
ተጠቆመ V I-VP I-VP
። PUNC O O

የአውሮፓ NP B-NP B-NP
ኮሚሽን N I-NP I-NP
ከፍተኛ ADJ I-NP I-NP
ሉክ N I-NP I-NP
ለፖሊሲ NP B-NP B-NP
ውይይት N I-NP I-NP
ወደ PREP B-NP B-NP
እትዮጵያ N I-NP I-NP
ሊመጣ VP B-VP  B-VP
ነው AUX I-VP I-VP
። PUNC O O

በሸራተን NP B-PP B-PP
በፈረንጆች NP B-NP B-NP
አዲስ ADJ I-NP I-NP
አመት N I-NP I-NP
ዋዜማ N I-NP I-NP
ልዩ ADJ B-NP B-NP
ኮንሰርት N I-NP  I-NP
ሊቀርብ VP B-VP B-VP
ነው AUX I-VP I-VP
። PUNC O O
```