ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
DEPARTMENT OF INFORMATION SCIENCE

PART OF SPEECH TAGGING FOR WOLAITA LANGUAGE

By
BERHANU HERANO GANTA

A thesis Submitted to School of Information Science of Addis Ababa University in Partial Fulfilment of the Requirements for the Degree of Masters of Science in Information Science

June, 2015
ADDIS ABABA UNIVERSITY
SCHOOL OF INFORMATION SCIENCE
DEPARTMENT OF INFORMATION SCIENCE

PART OF SPEECH TAGGING FOR WOLAITA TEXT

BY
BERHANU HERANO GANTA

A thesis Submitted to School of Information Science of Addis Ababa
University in Partial Fulfilment of the Requirements for the Degree of
Masters of Science in Information Science

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martha Yifiru (PhD)</td>
<td>Advisor</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>Solomon Teferra (PhD)</td>
<td>Examiner</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>Wondwossen Mulugeta (PhD)</td>
<td>Examiner</td>
<td>----------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
DECLARATION

This thesis is my original work and has not been submitted for a degree in any other University.

________________________________________

Berhanu Herano

June, 2015

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

________________________________________

Martha Yifiru (PhD)

June, 2015
ACKNOWLEDGEMENT

Almighty GOD, thank you for everything you have done for me throughout my lifetime. Next, I would like to thank my advisor, Dr. Martha Yifiru for her support, patience, guidance, and encouragement from the beginning of this thesis to the end. This thesis would never have been possible without her insightful observations and advice.

Next, I would also like to thank Mikael Melese for connecting me with Tewodrose Abebe Who provided me with all the materials and connected me with linguist advisor. I also like to thank Akililu Abera, PhD student at linguistics department of AAU for helping me in the language aspects and for manually tagging the corpus used for this study.

I also thank Thorsten Brants for providing me with TnT tool which was used for the development of HMM based tagger for Wolaita language in this thesis.

My thanks also go to Addis Ababa University staff for providing me with all the necessary services throughout my stay in the university.

I am grateful to express my gratitude to my classmates Biruk Retta, Dejene Hundessa, Aschalew Belay and Ashenafi Girma for all the support they provided me and their encouragement throughout my study. I also thank all my other friends for their good wishes and blessings.

Finally, I wish to express my appreciation and love to my family who has always been supporting, encouraging and loving me.
DEDICATION

DEDICATED TO MY MOTHER:

RAHEL JAGGISSO
# Table of Contents

ACKNOWLEDGEMENT ........................................................................................................... i
Dedication ................................................................................................................................. ii
Table of Contents .................................................................................................................... iii
List of Tables ............................................................................................................................. vii
List of Figures ............................................................................................................................ viii
Acronym .................................................................................................................................... ix
Abstract ................................................................................................................................... xi

## CHAPTER ONE ...................................................................................................................... 1

1.1 Background ......................................................................................................................... 1
1.2 Historical Overview of part of speech tagging ................................................................. 2
1.3 Application Areas of Part of speech Tagging ................................................................. 4
1.4 Statement of the problem ................................................................................................. 5
1.5 Objective of the Study ....................................................................................................... 6
   1.5.1 General Objective ....................................................................................................... 6
   1.5.2 Specific Objectives .................................................................................................... 7
1.6 Methodology ....................................................................................................................... 7
1.7 Significance of the Research ............................................................................................ 8
1.8 Limitation of the Study ..................................................................................................... 8
1.9 Scope of the Study ............................................................................................................ 9
1.10 Organization of the Thesis ............................................................................................. 9

## CHAPTER TWO ..................................................................................................................... 11

PART OF SPEECH TAGGING ................................................................................................. 11
2.1 Introduction ....................................................................................................................... 11
2.2 Methods of POS Tagging .............................................................................................. 14
3.4.11 Tags for Verbs ................................................................. 43
  3.4.11.1 Tag for Main Verbs .................................................. 43
  3.4.11.2 Tags for Subordinate verbs ..................................... 44
  3.4.11.3 Tags for Relative verbs .......................................... 44
  3.4.11.4 Tags for Infinitive verbs ......................................... 45
  3.4.11.5 Tags for verbs with Infix ........................................ 45

3.4.12 Tags for punctuation marks in the language .................. 45

CHAPTER FOUR ............................................................................. 50

METHODOLOGY OF THE STUDY ..................................................... 50
  4.1 Introduction ....................................................................... 50
  4.2 Data Preparation ............................................................... 50
  4.3 Evaluation Procedures ....................................................... 52
  4.4 Tools and techniques used ................................................ 53
    4.4.1 TnT (Traigrams’n’ Tagger) ....................................... 53
    4.4.2 CRF++ ....................................................................... 58

CHAPTER FIVE ............................................................................... 66

EXPERIMENTATION AND DISCUSSION ....................................... 66
  5.1 Introduction ....................................................................... 66
  5.2 The Experiment ................................................................. 66
    5.2.1 Experiment one .......................................................... 66
      5.2.1.1 Experiment 1 using TnT ...................................... 67
      5.2.1.2 Experiment 1 using CRF++ ................................. 67
    5.2.2 Experiment two ........................................................... 68
      5.2.2.1 Experiment 2 with TnT ...................................... 69
      5.2.2.2 Experiment 2 with CRF++ ................................. 69
    5.2.3 Experiment three ........................................................ 70
      5.2.3.1 Experiment 3 using TnT using 17 tags ................. 71
      5.2.3.2 Experiment 3 using CRF++ using 17 tags ............. 71
    5.2.4 Experiment four .......................................................... 72
LIST OF TABLES

Table 3.1 Description of tags .................................................................................................................................................. 49
Table 4.1 format of tagged and untagged file for TnT tool ........................................................................................................... 54
Table 4.2 File format used by CRF++ ....................................................................................................................................... 60
Table 4.3 Output of crf_learn .................................................................................................................................................... 61
Table 5.1 result of experiment using TnT .................................................................................................................................... 67
Table 5.2 experimental result of CRF++ ..................................................................................................................................... 68
Table 5.3 Experimental result of TnT .......................................................................................................................................... 69
Table 5.4 Experimental result of CRF++ ..................................................................................................................................... 69
Table 5.5 Experimental result of TnT on reduced tag set ............................................................................................................. 71
Table 5.6 Experimental result of CRF on the reduced tag set ...................................................................................................... 71
Table 5.7 experimental result of TnT and CRF on further reduced tag set. .................................................................................... 72
Table 5.8 Experimental result of TnT .......................................................................................................................................... 73
Table 5.9 Experimental result of CRF++ ..................................................................................................................................... 73
Table 5.10 Experimental result of TnT .......................................................................................................................................... 74
Table 5.11 Experimental result of CRF++ ..................................................................................................................................... 75
Table 5.12 Error analysis of TnT tagger on test data set. ................................................................................................................ 76
Table 5.13 Error analysis of CRF tagger on test data ................................................................................................................... .77
LIST OF FIGURES

Figure. 2.1 Classification of POS tagging Models (L. van Guilder, 1995) ............................................. 23
Figure 3.1 word classes of Wolaita (Wokasa, 2008) ............................................................................. 30
Figure 4.1 Example of unprocessed Wolaita corpus.............................................................................. 51
Figure 4.2 Example of preprocessed corpus......................................................................................... 52
Figure 4.3 Example of tagged corpus................................................................................................. 52
Figure 4.4 Training using TnT ............................................................................................................ 56
Figure 4.5 Tagging using TnT tool..................................................................................................... 57
Figure 4.6 Architecture of the tagging process................................................................................. 63
<table>
<thead>
<tr>
<th>ACRONYM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>POS</td>
<td>Part of Speech</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional Random Fields</td>
</tr>
<tr>
<td>MEM</td>
<td>Maximum Entropy Model</td>
</tr>
<tr>
<td>TnT</td>
<td>Traigams’n’Tags</td>
</tr>
<tr>
<td>No.</td>
<td>Number</td>
</tr>
<tr>
<td>%</td>
<td>Percentage</td>
</tr>
<tr>
<td>Pl.</td>
<td>Plural</td>
</tr>
<tr>
<td>Sg.</td>
<td>Singular</td>
</tr>
<tr>
<td>Ms</td>
<td>masculine</td>
</tr>
<tr>
<td>NNS</td>
<td>Common and Proper Nouns Plural</td>
</tr>
<tr>
<td>NN</td>
<td>Common and Proper Nouns Singular</td>
</tr>
<tr>
<td>NP</td>
<td>Nouns Suffixsed with Postpositions</td>
</tr>
<tr>
<td>NC</td>
<td>Nouns Suffixsed with Conjunctions</td>
</tr>
<tr>
<td>NV</td>
<td>Verbal Nouns</td>
</tr>
<tr>
<td>NA</td>
<td>Nouns with Adverbial Ending</td>
</tr>
<tr>
<td>PP</td>
<td>Personal Pronouns</td>
</tr>
<tr>
<td>DP</td>
<td>Demonstrative Pronouns</td>
</tr>
<tr>
<td>AD</td>
<td>Adverbs</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjectives</td>
</tr>
<tr>
<td>CC</td>
<td>Conjunctions</td>
</tr>
<tr>
<td>UJ</td>
<td>Interjections</td>
</tr>
</tbody>
</table>
DD  Demonstrative Determiners
WP  Wh-Pronouns
CD  Cardinal Numbers
OD  Ordinal Numbers
VT  Infinitive Verbs
VI  Verb Infixes
VS  Subordinate Verbs
VR  Relative Verbs
VB  Main Verbs
PU  Punctuation Marks

>>  The rest of 180 sentences
ABSTRACT

Natural Language Processing (NLP) has emerged as a solution to problems of helping computers understand natural languages and as a means of increasing computers capability to understand natural languages, by which most of human knowledge is recorded. NLP is used to design and implement tools, techniques and frameworks that enable computers to understand natural languages.

NLP applications take advantage of machine learning strategies in order to analyze large amounts of textual data. One of such NLP tasks to be performed on textual data is Part-of-Speech (POS) tagging, which is used for labeling or classifying every word of a text with its correct part of speech category like noun, verb, adjective, adverb, preposition, interjection etc.

Currently, there are many taggers developed for different languages of the world like English, French, and German etc. from abroad and also for Amharic, Afan Oromo, and Kaffi-noo-no from local languages. But for Wolaita language no attempt has been made in the area of NLP and particularly on POS tagging. Therefore, the aim of this study is to explore possibility of developing part of speech tagger for Wolaita language using supervised and semi-supervised machine learning approaches like Conditional Random Fields (CRF) and Hidden Markov Model (HMM) using small manually tagged corpus.

The models or taggers were developed based on the review and the study made on the Wolatia language word classes. The tags developed for this study were the first attempt for the language and are based on the study and review of word class of Wolaita language. In this research, 200 sentences were manually tagged and of these 200 sentences, 90% of the sentences (180 sentences) were used for training and the rest of the sentences were used for testing the performance of the tagger. The result of this experiment showed that HMM based taggers perform better than CRF based taggers. The performance that have been achieved has accuracy of 83.58% and 74.63% using reduced tag set for supervised Hidden Markov Model (HMM) and Conditional Random fields(CRF) based taggers respectively.

**Keywords:** POS Tagging, NLP, Supervised, unsupervised, semi-supervised, HMM, CRF, TnT.
CHAPTER ONE

INTRODUCTION

1.1 Background

These days, due to growth of scientific and technical advances, there is large amount of information that is retained and processed by business and organizations. Some of such information is stored in the form of text. Processing and retrieving useful information from such hugely available data and information is very difficult for human beings. To tackle such problem professionals and scientists from different area of studies like Artificial intelligence, information retrieval, natural language processing, data mining etc. started to conduct research that mainly focuses on helping computers understand natural languages (Allen, 1996).

Therefore, as a solution to problems of helping computers understand natural languages, Natural Language Processing (NLP) has emerged as a means of increasing computers capability to understand natural languages, by which most of human knowledge is recorded. NLP is used to design and implement tools, techniques and frameworks that enable computers communicate effectively with each other and with humans. NLP as scientific study encompasses a set of related disciples like psycholinguistic, linguistic and computational linguistic and other related fields to study and design effective components like morphological analyzer, syntax parser, semantic analyzer, speech recognizer, part of speech (POS) tagger and many more application that can help computers understand text, sounds, images and other forms of information just like that of human being (Jurafsky and Martin, 2007).

NLP applications take advantage of machine learning strategies in order to analyze large amounts of textual data. One of such NLP tasks to be performed on textual data is Part-of-Speech (POS) tagging, which is used for labeling or classification of every word of a text with its correct part of speech category like noun, verb, adjective, adverb, preposition, interjection etc. based on its definition and context of adjacent and related word (Marcos and Pablo, 2014). But the effort to make computers understand natural language is not an easy task rather it is a difficult task. One of the challenges that makes development and application of NLP systems difficult is ambiguity which is the tendency of words and sentences to have more than one
linguistic structure or meaning i.e. a word in a sentence can have different meaning or more
than one (POS) based on the context. Ambiguity, which is due to the existence of words that
have more than one POS tag, is challenge to NLP systems in making the right inference that is
needed in natural language understanding. For example, as indicated by Arslan and Patan
(2009), the sentence “we can can the can” has different possible tag for one word i.e. can. Here
the word ‘can’ corresponds to auxiliary verb, verb and noun respectively. There is ambiguity at
different level of NLP. It can be at structural, lexical and semantic level of NLP. So it demands to
carry researches that focus to solve ambiguity at different levels of NLP.

1.2 Historical Overview of part of speech tagging

The study of POS tagging has been known and begun since 1960’s and 1970’s for languages like
English and its evolution is highly tied to the evolution of linguistics as a science. Stolz et al in
1965 presented a stochastic approach to the coding of English language which shows one of the
efforts made to provide solution on these word class problems or part of speech tagging. A
complete probabilistic tagger with Viterbi algorithm was designed by some years later by Bahl
and Mercer in 1976 (Stolz et al in, 1965)

Research on part of speech tagging is also closely tied to corpus linguistics. One of such corpus
is the Brown Corpus which is the first major corpus of English for computer analysis developed
at Brown University by Henry Kucera and Nelson Francis in the mid-1960s. This corpus was used
for the study of POS tagging in English language and inspired the development of similar tagged
corpora in other languages over the world. Green and Robinson (1971) developed TAGGIT
which is a rule based system to annotate textual data by using lexicon and suffix list. In 1980s,
machine readable text and as well as large annotated bodies of text (corpora) became available
and as the result statistical NLP and machine learning methods that use probability became
prominent (Karen, no year).

In 1996s, low-level NLP tasks like POS tagging, sentence boundary detection, tokenization,
morphological segmentation and shallow parsing become areas of interest and high-level NLP
tasks like spelling error or grammatical error identification and recovery and named entity recognition became area of interest as well (Karen, no year).

In addition, different POS tagging approaches became available lately having different classification. One of the classifications of POS tagging approaches is supervised or unsupervised based on the source of the data used for the process of tagging. In the case of supervised tagging, human being is involved to guide the tagger to some extent by providing the tagger with manually tagged data. A supervised tagger depends mainly on manually tagged data, dictionary and so on. On the other hand, unsupervised tagger does not depend on pre-tagged corpora and dictionary opposed to supervise ones (Eric, 2005).

The other classification of POS tagging approach, in addition to Supervised and unsupervised tagging is rule based and statistical. Both rule based and corpus based tagging methods are either supervised or unsupervised. Corpus based and rules based POS tagging approaches differ from each other by the manner in which they solve the tagging problem. While rule based taggers solve the tagging problem by using hand-written rules, Statistical taggers solve the tagging problem by estimating probability of a given word from the training corpus.

Steven (1996) reported that the advantages of rule based taggers over corpus based taggers are that they need small storage and have better speed. But writing disambiguation rules is expensive and time consuming. During early time, the problem related to stochastic approach was unavailability of large corpora. But currently due to advances in computers capabilities and increase in the amount of digitized text, stochastic approaches are becoming popular and widely used.

In addition to rule based and corpus based, there are also combined approaches to POS tagging problem i.e. Transformation based POS tagging. These types of taggers use rules but the rules are learned from the corpus by using stochastic scoring function (Eric, 2005). Machine learning approaches like Decision tree, support vector machine and neural network also became available for solving POS tagging problem.
1.3 Application Areas of Part of speech Tagging

Some of the applications of Part of speech tagging are: **Parsing**: Parsing is used to assign syntactic description of the sentence. Parsers are useful for text analysis, corpora analysis, machine translation etc. The output of a POS tagger can be used as an input for syntactic analyzer or parser and the output of a parser in again can be used as an input for a semantic parser and automatic machine translation. According to Allen (1995), if the output of a tagger is used as input for a sentence parser, the performance of the parser will be improved. Since POS tagging assigns unique tag to each word this can help to reduce the number of parses.

**Information Extraction**: Information extraction, on the other hand, is a means of retrieving certain types of information from natural language text automatically. The main aim of information extraction is to process natural language text and to retrieve occurrences of a particular class of objects or and occurrences of relationships among objects. Information extraction is also a form of natural language processing in which certain types of information must be recognized and extracted from text. It extracts their semantic contents tagging and helps to identify useful terms and relationships between them. In addition, POS references are used by patterns which are used for information extraction from text. POS tagging helps Information extraction to identify useful terms and relationships between the terms.

**Information Retrieval**: POS tagging provides information retrieval system with POS information which helps information retrieval system with more refined information so that information retrieval system can eliminate retrieval of irrelevant documents to the query.

**Question Answering**: POS tagging helps to analyze a query to what type of entity the user is looking for and how this entity is related to other noun phrases mentioned in the question. The study of question answering systems, which enable people to locate the information they need directly from large free-text databases by using their queries, has become one of the important aspects of natural language processing and information retrieval and it is highly related to part POS tagging.
**Speech Synthesis and Recognition**: The most important information about the word and its neighbors (words that occur before and after the word) are useful in a language model for speech recognition. Such information can be derived from POS tagging. POS of a word can also indicate us something about how the word is pronounced depending on the grammatical category of the word (Heeman et al., 1997).

**Machine translation**: POS tagging has also great influences on the probability of translation of word in source language to the word in target language in machine translation. POS tagging is the first and the very important task and step in machine translation and it can be part and component of machine translation system. In addition to above mentioned areas of application, POS tagging has also been used in other application like lexicography, word sense disambiguation etc. (Sandipan, 2009).

### 1.4 Statement of the problem

According to Bender (1976), Ethiopian language families such as Semitic, Cushitic and Omotic are categorized under Afro-Asiatic phylum. Wolaita belongs to Omotic language family that is spoken among Wolaita people in Southern part of Ethiopia. Wolaita language is also widely spoken around Gemu, Gofa, Kucha, Dawro and Kullo with small difference in dialect (Bender, 1976).

There are a lot of translations that have been made so far from Amharic to Wolaita and from English to wolaita and most of the translations made are religious books and materials (Wakasa, 2008). Since the language is used as a means of communication in government offices, and also serving as working language at Wolaita Zone, there are huge number of documents and news articles produced and stored in this language both in hard and soft copy.

In spite of availability of huge amount of information in different formats like text and so on and a good number of speakers, to my knowledge, there are only two researches that have been carried out on Wolaita language from NLP point of view. These are: development of stemming algorithm for Wolaita language by Lemma (2003) and Text-to-Speech Synthesizer for
Like stemming and speech synthesis, POS tagging is also very important NLP task that has to be done for Wolaita language.

Lemma (2003) recommended that someone in the future can come up with a corpus that is useful in natural language processing for Wolaita language and Mesfin (2001) reported that it is difficult for researchers who are interested to work on linguistics and computational linguistics without the availability of part of speech tagging systems which are a source of useful annotated corpora for a particular language.

Because of the above mentioned facts and the wider application of POS taggers, it is important to study or carry out research from natural language perspective on Wolaita language specially on POS tagging which is the first and lower level NLP task that can be used as an input for researchers who are interested in other higher level NLP tasks like syntactic parsing, machine translation, word sense disambiguation etc. in the language.

To the best of my knowledge, no effort has been made to investigate POS tagging for Wolaita language. Therefore, it makes worth to do a research on part of speech tagging for Wolaita language which has many NLP applications and can open the door for other researches who are interested to carry out NLP related research in the language.

**Research Questions**

1. What is the effect of reducing the number of tags on the performance of taggers?
2. Which tagger (HMM or CRF) has better performance for small data?
3. What are the challenges in developing POS tagger for Wolaita?

**1.5 Objective of the Study**

**1.5.1 General Objective**

The general objective of this study is to investigate possibility of developing part of speech tagger for Wolaita text using small manually tagged text.
1.5.2 Specific Objectives

The specific objectives which are important to achieve the general objectives of the research are:

- To review literature on POS tagging approaches and word categories of Wolaita language.
- To collect, preprocess, manually annotate Wolaita text.
- To identify tag sets for Wolaita text.
- To select and use tools and algorithm to train the POS tagger.
- To carry out POS tagging experiment on Wolaita text.
- To evaluate the performance of the tagger.
- To report the result of the experiment.
- To draw conclusion and forward recommend for future research.

1.6 Methodology

**Literature review:** Different relevant and related literature resources such as books, research reports, journal articles, and other published and unpublished documents including those from the Internet were reviewed for the purpose of this study. All these have helped the researcher to understand both the issues related to NLP, particularly POS tagging, approaches used in POS tagging, techniques, strategies used in POS tagging, the language structure and word class.

**Discussion:** Continuous discussion with linguists and experts in the area of Wolaita language has been made to understand the language structure, to determine the tag sets and to tag the corpus manually.

**Data collection and preparation:** Text data (corpus) was collected by the researcher from a text book which is used as reference for students of language department at Wolatia Soddo University and the data collected was preprocessed by removing foreign words, brackets and tokenized by the researcher in order to make it suitable for the tools which were used to develop the model in this thesis. Python programming language was used for tokenization.
Since there is no POS tagged text data for Wolaita language, the corpus which contains 200 sentences was collected, prepared by the researcher and manually tagged by linguist. The models were trained using the training data and tested using test data set to obtain best tag sequence for each word sequence in the corpus. The taggers were trained on training data of which was about 90% (180 sentences) of the total corpus (200 sentences) and tested with test data of about (20 sentences) or 10% of the total corpus.

**Tools and algorithms:** After the analysis of Wolaita language, tools, algorithms and approaches that are available were selected and used for developing the taggers for the language. In this study supervised and semi-supervised Hidden Markov model (HMM) based TnT tool and conditional Random Fields (CRF) based CRF++ tool were used because of their simplicity, availability, language independence, good performance on both foreign and local languages.

**Performance evaluation:** The performance of the systems or the developed taggers was evaluated by comparing the output of the taggers with test data that is manually tagged and accuracy was measured as the percentage of the number of correctly tagged words in the test data divided by total number of tokens in the test set. The accuracy was also measured in terms of known, unknown, and overall and the results were reported.

### 1.7 Significance of the Research

So far it has been discussed in this chapter that POS tagging is basic and the first NLP task for any language and it can be used as an input for other high level NLP tasks. So, investigating part of speech tagging for Wolaita language can be used as input for other NLP tasks in the language like sentence parsing, machine translation, word sense disambiguation etc. This research can also be used as a reference for other researchers who are interested to work on Wolaita language at different level of natural language processing.

### 1.8 Limitation of the Study

The taggers developed in this research are trained and experimented using small amount of data due to the absence of large size POS annotated corpora for the language and also POS
annotation of large size corpora is time consuming, expensive and laborious. Semi-supervised approach based experiments carried out in this research have not been accompanied by hand labeling and only two semi-supervised experiments were conducted to due to time constraint.

1.9 Scope of the Study

The scope of this research is limited to investigate part of speech tagging for Wolaita text using small manually tagged corpus and supervised and semi-supervised POS tagging approaches like HMM and CRF. Other approaches like rule based and unsupervised approaches were not included in this thesis due to shortage of time and resource available for this work. The taggers developed in this research tag only words or it cannot tag sentences because the tools used to develop the taggers in this research do not support sentence level tagging.

1.10 Organization of the Thesis

This research was organized in six chapters. The first chapter introduces what NLP is and highlight about historical overview of POS tagging, and some application areas of POS tagging. This chapter also presented and discussed the statement of the problem, the objective of the study, methodology used in the study, significance of the study, scope and limitation, and organization of the research.

Literature review to understand the areas of NLP in general and POS tagging in particular, different approaches used and strategies for developing POS tagging were presented in chapter two. This chapter also presented different supervised (machine learning) approaches like HMM, CRF, Decision Tree, n-gram etc. based approaches. In addition, review of related works like POS tagging for some other and local languages was also briefly presented in chapter two.

The third chapter was mainly dedicated to describe the Wolaita language in relation to word class of the language and POS categories like verbs, nouns, adjectives, adverbs etc. in the language. The fourth chapter mainly discussed and described methodology used in this study, data preparation and preprocessing and sample selection.
Experiments conducted, and the discussion on results observed was reported in chapter five. Finally, the conclusions and recommendations made based on the findings of the study were presented in sixth chapter.
CHAPTER TWO
PART OF SPEECH TAGGING

This chapter presents review of literature on part-of-speech (POS) tagging. The chapter begins with a brief introduction to language, POS tagging and presents further discussion on various approaches and techniques used for POS tagging. The chapter also gives explanation and discussion on unsupervised, supervised, statistical or machine learning and rule based approaches to POS tagging. Finally, the chapter ends with reviewing and discussing locally and globally carried out researches in the area of POS tagging.

2.1 Introduction
Natural language or language in general is a means of communication that helps humans to share knowledge, opinion and ideas among themselves. Natural language which is used by human beings can be expressed by speech, sign or writing. It is a system of rules and conventions i.e. it is a combination of words, a pause and alphabetic letters. A single word, a pause and alphabet letters provide us with only immediate meaning of the word itself or the alphabets but when they are combined together they provide us with more information than they are separated. (Allen, 1996).

Words are the important and the main building blocks of a language. Every human language, spoken, signed or written is composed of words. Research areas of NLP like speech recognition, machine translation, text to speech, spelling and grammar checking, language-based information retrieval on the Web and POS tagging require extensive knowledge about words that are based on the lexical knowledge. Knowledge of the language is very important and useful in every language processing applications (Jayaweera and Dias, 2014).

Part of speech, which is also known as word class, lexical class or lexical category, is defined as a linguistic category of lexical items or words. It is defined by morphological and syntactic behavior of the words. The known part of speech include: nouns, verbs, adjectives, adverbs, prepositions, interjections etc. Word category of one language may be different from that of
other language. Part of speech provides important information about a word and its neighbors in language processing (Jurafsky and Martin, 2005).

POS tagging, which is one of the fundamental processes in the field of Natural Language processing (NLP), is an area of research for any language and fundamental processing step for any language in NLP and language automation, i.e., the capability of a computer to automatically POS tag a given sentence.

As it has been briefly defined in chapter one, POS tagging is the process of classifying each word in a text or in a corpus to some particular POS such as noun, verb, adjective, adverb etc. based on the context in which the word is used in the text. It is a technique that is used to annotate words and assign appropriate POS to each word in a text corpus. The string used as a label in POS tagging is called a tag, the set of labeling strings is called a tag set, and a tagger is a program that performs tagging. POS tagging is an important natural language processing activity that is widely used in language analysis. As the first and early analysis of linguistic text, POS tagging has many applications in NLP especially in higher level NLP tasks like machine translation, speech recognition, parsing, question answering, information retrieval, information extraction and so on (Jurafsky and Martin, 2007).

POS tagging can be done either manually or automatically. In the case of manual, POS tagging is done by language experts who are familiar with some particular language. Tagging text corpus manually is expensive and time consuming. Automatically, POS tagging is done using NLP tools and software that are developed and designed for purpose of POS tagging (Jayaweera and Dias, 2014).

For the purpose of conducting POS tagging experiment, we need a large collection of text data called corpus which is a kind of written material that is used for linguistic analysis. Corpus helps grammarians, lexicographers and other parties by providing better description of the language. By analyzing a corpus, we can get lexical, morph syntactic, semantic and pragmatic information of the language. A sentence or a word in the corpus can be POS annotated as indicated in Wolaita sentence ‘битánee/NN bullúkkuwa/NN dáddiis/VB’ ‘The man wove the blanket.’ that means each word or sentence has part of speech such as noun verb or adjective etc. attached
to it or un-annotated where word in the text do not have part of speech attached to as in the sentence ‘mishiriya bambariya daaTasu.’ ‘The woman ground the red paper.’ As we can see from the second example, words in the sentence do not contain structural or linguistic information. Corpus that contains annotated words is used in the development of NLP tools in the areas of computational linguistics, speech recognition, machine translation and POS tagging. Its applications are spelling-checking, grammar-checking, speech recognition, text-to-speech and speech-to-text synthesis, automatic abstraction and indexing, information retrieval and machine translation. Examples of corpus includes: British National Corpus, the Brown corpus, etc. (Marquez, 1999).

The task of POS tagging is very difficult because of ambiguity that is related to the nature of words to have more than one meaning and more than one POS. The globalization of the world, having access to information, media-industry, Internet, the ongoing advances in communication technologies etc. play a crucial role in the forming of today’s and tomorrow’s languages and due to globalization and change in technology every natural language is under constant change and the words in the language become ambiguous (Sandipan, 2009).

In NLP ambiguity can occur at different level of the natural language transforming assignment. There are many words that take different POS. The right tag of a word in a text depends on the context in which the word is used in particular text. Sentences contain words which have a lot of POS ambiguity, and it is necessary to disambiguate words in the sentence before the sentence is understood. For example, based on their context, the words like "keep" and "book" can be a verb or a noun, and "on" can be a preposition, an adjective, an adverb and "top" can be an adjective or a noun in the sentence "Keep the book on the top shelf" (Sandipan, 2009). This is true for Wolaita word ‘qera’ (which means young), (name of market) or (Saturday) in sentences:

- Qera na7i yiis. ‘The young boy has come.’
- Ta qera bassi. ‘I went on Saturday.’
- Ha giya suntai qera. ‘the name of this market place is qera’

The word qera is adjective, adverb and noun respectively in the above sentences.
In addition to ambiguity, the following factors make part of speech tagging very difficult:

- **Unknown word**: this is due to the appearance of new words, other words are becoming forgotten and some other words get new spelling.
- **Indeterminacy**: is related to the idea that natural language is open and evolving system and there is no agreement among human experts about which tag is the correct one for different words in the text.
- **Noise**: there are errors in the data resources which are used to build language technology applications (Johan, 2003).

Currently, there are many POS tagging systems developed for many languages like English, French, Spanish, and German etc. from abroad and for Amharic and Afan Oromo from local which are used to annotate corpora written in these languages. Due to the availability of POS taggers for the above mentioned languages, the annotated corpora became widely available for the languages. But this is not true for some of Ethiopian languages like Wolaita to annotate the text written in the language.

### 2.2 Methods of POS Tagging

There are different approaches or methods of POS tagging. But the most popular and well known approaches are classified into two. They are rule based and corpus based or corpus based. Corpus based tagging methods are either supervised or unsupervised based on whether they use manually tagged corpus or not. Rule based and corpus based differs from each other by the manner in which they solve the tagging problem. While rule based taggers solve the tagging problem by using hand-written rules, corpus based taggers solve the tagging problem by estimating probability of a given word from the training corpus.

Steven (1996) reported that rule based taggers are better than corpus based taggers based on storage and speed i.e. rule based taggers need small storage and have better speed. But is difficult and time consuming to develop rules. During early time, corpus based approaches suffered from unavailability of large corpora. But currently due to advances in computers capabilities and increase in the amount of digitized text, stochastic approaches are becoming
popular and widely used. In addition to rule based and stochastic, there are also combined approaches to part of speech tagging problem. These types of taggers use rules but the rules are learned from the corpus by using stochastic scoring function (Brill, 2005).

2.2.1 Rule Based POS Tagging
Rule based POS tagging approaches use contextual information to assign tags to unknown or ambiguous words. It is the oldest approach to POS tagging that uses hand-written rules for tagging. Rule based POS taggers are highly dependent on dictionary or lexicon to assign possible tags to each word. Morphological and lexical information contained in the words of the language is used in the process of tagging unknown words. These taggers are developed for the purpose of tagging specific type of language opposed to corpus based taggers (Loftsson, 2008).

Rule based taggers do POS tagging by analyzing the linguistic features of the word like the preceding and the following word. Rule based taggers use two stage architecture to assign part of speech. Lexicon is used in the first stage to assign the word with possible parts of speech but in the case of the second stage, list of hand-written disambiguation rule is used to identify the correct part of speech when a word has more than one possible tag. For example, in English if the word is preceded by an article, then that word is necessary a noun (Jurafsky and Martin, 2005).

TAGGIt is a known and the first large rule based tagger that uses context-pattern rules. It used a set of 71 tags and 3300 disambiguation rules. These rules disambiguated 77% of words in the one million-word of Brown University corpus. This method usually assigns lists of potential POS tags to each word based on dictionary and is has manual rules for Out of Vocabulary (OOV) words. Some of constraints that can be used by this tagger are: determiner cannot immediately precede a verb and no verb can immediately precede a tensed verb (Brill, 2005). Rule based taggers require different components to perform tagging. The components include: POS tag set: is a list of all word categories used in the tagging process.
A lexicon or dictionary: this component is required for rule based tagger because rule based taggers process sentence by looking up words in the dictionary to get the possible part of speech assigned to words.

Lexical or morphological rule: this rule is used to provide information that is important for words that are not in the lexicon of the tagger and it used to guess POS category of unknown words.

According to Brill (1192), rule based tagger has the following advantages over corpus based ones:

- High reduction in stored information that is required for tagging.
- It requires small set of rules unlike that of stochastic taggers that need large table of statistics.
- Better portability from one tag set or corpus type to the other.

Some of Disadvantages of rule based taggers are:

- Writing disambiguation is expensive and time consuming.

2.2.2 Corpus Based POS Tagging

Due to advances in computers capabilities, increasing amount of digitized text and availability of large tagged corpora, the development of corpus based POS taggers was encouraged since 1960. This approach works based on probability to solve POS tagging problem. Taggers developed based on this approach are classified into supervised and unsupervised based on whether they are trained on manually annotated corpora or not. Taggers that need manually annotated corpora are called supervised taggers and the method or approach is called supervised approach (Sandipan, 1996).

On the other hand, taggers that do not need manually tagged corpora are called unsupervised and the method is called unsupervised approach. Supervised corpus based taggers use lexicon and contextual probability. Supervised stochastic tagging techniques use only manually tagged data. They also require large amount of tagged data in order to achieve high level of accuracy.
On the other hand, unsupervised corpus based techniques are data-driven techniques which do not require a pre-tagged corpus but instead they use some sort of computational methods to automatically induce word groupings or tag sets, and based on these automatic groupings, they calculate the probabilistic values needed by stochastic taggers (Sandipan, 1996).

Currently, Machine learning techniques have been used to solve POS tagging problems. Machine learning algorithms take advantages of annotated corpus to derive language knowledge for different NLP tasks. With the help of Machine learning techniques, it has become possible to develop POS taggers within short period of time. Until now, lots of researches have been carried out on POS tagging tasks using machine learning approach. Taggers based on machine learning need to be trained with a large amount of tagged data. These days, there are many languages in the world for which we can get a lot of tagged corpus which is very useful for training the taggers (Marquez, 2000).

According to Marquez (1999), the application of machine learning based approaches to NLP problems is getting attention in NLP community these days. He also argues that machine learning techniques are very useful to address natural language disambiguation problems which appear at all levels of natural language understanding process (Marquez, 1999).

### 2.2.2.1 Supervised Versus Unsupervised POS Tagging

As it is explained in this chapter, depending on what data-sources is used in the tagging process each tagging procedure may be classified as being more or less supervised or unsupervised. A supervised method is supervised by a human being at some level to guide the tagger. A supervised tagger may rely on pre-tagged corpora, word-tag frequencies lists, dictionaries etc. (Jayaweera and Dias, 2014).

The two approaches, supervised and unsupervised, are differ from each other based on the degree of availability of tagged corpus for training. For unsupervised POS tagging models, we do not need annotated corpus. As alternative to annotated corpora, unsupervised POS tagging models use advanced computational techniques to generate transformation rules, tag set etc.
For supervised POS tagging models, we need annotated corpus which is used for training to get information about the tag set, tag sequence and probabilities (Jayaweera and Dias, 2014).

In the case of supervised POS tagging, the goal is to learn and produce the correct output given a new input and the tagger is also given a sequence of desired outputs but unsupervised tagging is learning without teacher because correct answer is not provided and it does not split data into training and test sets (Jayaweera and Dias, 2014).

**Supervised tagging:** Characteristics of this approach are:

- It is supervised by human being or human being is involved in the process by providing manually tagged data or train the tagger by providing training data.
- It requires manually labeled or tagged data
- It uses word tag frequency lists, dictionaries and rule sets etc. (Fahim, 2006).
- It usually uses Viterbi algorithm.

**Unsupervised tagging:** This approach has the following characteristics:

- No involvement of human being
- No requirement of manually tagged corpus.
- No need of word tag frequency list and dictionaries.
- Use advanced computational methods like Baum-Welch algorithm in order to induce tag sets (Fahim, 2006).

In addition to unsupervised and supervised POS tagging, there are also semi-supervised approaches to POS tagging. Semi-supervised POS tagging approaches use a combination of small manually tagged and more untagged data to develop POS taggers. In this approach, first tagger is developed using small manually tagged corpus. Then the tagger which is developed using small manually tagged corpus is used to tag more and more untagged words to increase the size of training data and most of the time it is accompanied by hand correcting by expert for some incorrectly untagged words to improve the performance. It is the most important strategy for under resourced languages which have no standard POS tagged corpus.
Some of machine learning approaches used in part of speech tagging like N-gram model, Hidden Markov model (HMM), Maximum Entropy model (MEM), Decision Tree approach, Neural network and Conditional Random Field (CRF) models are briefly described below:

**N-gram model**

It is a statistical approach that is used to predict probabilities of a word on the basis of few previous words. It uses Markov assumption which states that the probability of a word is calculated on basis of the last few words. The probability of a word \( w \) is calculated on the basis of \( N-1 \) previous word/s where \( N \) indicates number of words in a sequence.

There are three types of N-gram models:

- **Unigram model**: Here the probability of the \( n^{th} \) word does not depend on any of the previous word or the unigram model only considers the probability of a word for a given tag \( t \). The surrounding context of that word is not considered. For each token it assigns the most likely tag for that token in the text. The unigram tagger must be trained with training corpus before it is used to tag the data. It assigns the default none to any token not encountered in the training corpus.
- **Bigram model**: Here Probability of the \( n^{th} \) word depends only on one previous word.
- **Trigram model**: Here Probability of the \( n^{th} \) word depends only on previous two words (Jurafsky and Martin, 2005).

**Hidden Markov Model (HMM)**

It the most important and widely used statistical and machine learning approach in NLP application areas likes POS tagging and speech recognition. It is a tool that is used to represent probability distribution over sequence of observation and used to solve classification problems that have inherent sequence representation. The model is visualized by the set of states which are connected by a set of transition probabilities that indicate moving between two states. Given a sentence or a word sequence, HMM taggers choose the tag sequence that maximizes the formula: \( P \text{(word | tag)} \times P \text{(tag | previous n tags)} \).
HMM based taggers are characterized by the following components:

- **N**, the number of distinct states in the model and for part of speech tagging, N is the number of tags that can be used by the system. Each possible tag for the system corresponds to one state of HMM. A transition probability matrix which helps to represent probability of moving from one state to the other.

- **M**, the number of distinct output symbols in the alphabet of the HMM. For part of speech tagging, M is the number of words in the lexicon of the system.

- **A= \{a_{ij}\}** is the state transition probability distribution. The probability \(a_{ij}\) is the probability that the process moves from state \(i\) to state \(j\). In part of speech tagging \(a_{ij}\) shows the probability that the model will move from tag \(t_i\) to \(t_j\).

- **B= \{b_j (K)\}**, the observation symbol probability distribution that represents the k-th output symbol which is emitted when the model is in state \(j\). For part of speech tagging it is the probability that the word \(k\) will be emitted when the system is at tag \(t_j\) or \(P(w_k | t_j)\).

- **Π= \{π_i\}**, is the initial state distributions and is the probability that the model will start in state \(i\). In part of speech tagging this indicates the probability that the sentence will begin with tag \(t_i\) (Jurafsky and Martin, 2005).

Some of Successful Application Areas of HMM are:

- POs tagging
- On-line handwriting recognition
- Speech recognition
- Gesture recognition
- Language modeling
- Motion video analysis and tracking
- Protein sequence or gene sequence alignment
- Stock price prediction etc. (Sung-Jung, 2005).

**Maximum Entropy model (MEM)**
The maximum entropy approach to POS tagging is one family of machine learning or corpus based approaches to classification in which many features are computed for the word to be tagged, and all the features are combined in a model based on multinomial logistic regression. Based on this approach, whenever we choose among different probabilistic models for a set of data, the most valid model is the one that makes fewest arbitrary assumptions about the nature of the data. The unknown word model in the tagger uses features of a word which represent property of a word and some of the features are like:

- Word contains a number.
- Word contains an upper-case letter.
- Word contains a hyphen.
- Word is all upper-case.
- Word contains a particular prefix (from the set of all prefixes of length ≤ 4)
- Word contains a particular suffix (from the set of all prefixes of length ≤ 4)
- Word is upper-case and has a digit and a dash (like CFC-12)
- Word is upper-case and followed within 3 words by Co., Inc., etc. (Ratnaparkhi, 1996).

The main advantage of this framework is that it helps or enables us to represent problem specific knowledge in the form of feature (Ratnaparkhi, 1998).

**Decision Trees**

It is one of statistical or machine learning approach to POS tagging where internal nodes represent tests and leaves represent conditional probability distributions. In decision tree the path from the root to the leaf is obtained by applying test to the current node and then choosing a branch to a child node that corresponds to outcome of the test and this process is recursively repeated from the new child node. This method estimates transition probability by using decision tree. Unlike that of Hidden Markov model, decision tree approach avoids problem of estimating transition probabilities from spars data (Ratnaparkhi, 1998). Decision tree is a Markov model using DT for estimating transition probabilities (Ratnaparkhi, 1998, Ahmed, 2013).
Neural Network

Neural network is one of important machine learning technique of part of speech tagging. It has ability of distributing activation patterns learned from training set across the links via the learning algorithms in the way similar to the human brain. It has different layers that are interconnected and work as a processing unit together. Neural network system uses contextual information and tagged words as input. Schmid (1994) reported an accuracy rate of 96.22% for neural network based taggers and found that neural network is better than Markov model based taggers.

Conditional Random fields (CRF)

Conditional Random Fields (CRF) is one of popular probabilistic methods for structured prediction. CRFs have wide application in natural language processing (NLP), computer vision, and bioinformatics.

Graphical modeling is a framework for representation and inference in multivariate probability distributions. Graphical modeling is useful in different areas of stochastic modeling, including coding theory, computer vision, knowledge representation Bayesian statistics and natural-language processing (Aadil, 2014).

It has a single exponential model for the joint probability of the entire sequence of states given the observation sequence. A CRF based method can also deal with diverse and overlapping features. A CRF is a very flexible method which deals with the sparse data problem well. Under this model, a natural combination of diverse set of features can be easily incorporated, which cannot be done naturally in HMM (Aadil, 2014).

CRF is characterize by the following features

- Can redefine feature sets
- Written in C++ with
- Fast training
- Less memory usage both in training and testing
- Encoding or decoding in practical time
- Can perform n-best outputs
- Can perform single-best MIRA training
- Can output marginal probabilities for all candidates (Lafferty, McCallum and Pereira, 2001).

The following figure shows the general classification of POS tagging models. Supervised stochastic models like Hidden Markov, n-gram and maximum entropy use Viterbi algorithm. But unsupervised stochastic models use Baum Welch algorithm.

![Classification of POS tagging Models](image)

Fig. 2.1 Classification of POS tagging Models (L. van Guilder, 1995).
2.3 Related Work

There are so many researches that have been carried out so far in the area of NLP in general and POS tagging in particular for several languages of the world. Since there are many researches that are carried out abroad in the area of POS tagging for different languages of the world using different approaches, it is impossible to mention all of them. In this section, only few of them which are related to approaches and tools used in this thesis were systematically selected and discussed. And all local attempts in the area of POS tagging are briefly discussed.

Chirag and Karthik (2008) used Conditional Random Fields (CRF) to develop POS tagger for Gujarati language. They used 600 sentences tagged and 500 sentences untagged for training the Pos tagger. They achieved an accuracy of 92% using 10,000 words for training and 5,000 words for testing with 26 tag sets.

Kupeic (1992), developed standard POS taggers using Hidden Markov Models which was adopted from speech recognition and applied to tagging and achieved 96.3% accuracy (Abny, 1996).

Thorsten Brants (no year) carried out experiment on Penn Treebank corpus which contains 50,000 sentences (1.20million tokens) using HMM based TnT tool and achieved accuracy of 97%, 85.5% and 96.7% known, unknown and overall accuracy using 90% of the corpus as training and the rest for testing by averaging 10 test runs.

Thorsten Brants (no year) carried out another experiment on Negra (German) corpus which contains 20,000 sentences (355,000 tokens) using HMM based TnT tool and achieved accuracy of 97.7%, 89.0% and 96.7% known, unknown and overall accuracy using 90% of the corpus as training and the rest for testing by averaging 10 test runs.

Samira et al. (no year) experimented to develop POS tagger for Persian language using TnT (HMM based) tool using 2,192,982 token for training and 400,234 tokens for testing with 40 tags and achieved accuracy of 97.01%, 77.77%, and 96.64% for known, unknown and overall respectively.
In addition to foreign languages, there are also some researches in the area of POS tagging for local languages. The first research was carried out by Mesfin (2001) to develop part of speech tagger for Amharic language using a Hidden Markov Model (HMM). He used 25 part of speech tags that are extracted from 300 words. He used 300 words for both training and testing of the tagger which could not guess part of speech tag of unknown words. The result he achieved was 97% on training and 90% on test set (Mesfin, 2001).

The second attempt on developing part of speech tagger for Amharic language was made by Sisay (2005) using Conditional Random Fields. He developed a tag-set of 10 tags from text corpus of five Amharic news articles (1000 words) and achieved accuracy of 74% (Sisay, 2005).

Yenewomdim (2006) worked on part of speech tagging for Amharic language using multilayer perception neural network on 159 sentences (136 sentences used for training and 23 sentences used for testing) using 23 tag-sets which were identified by Mesfin (2001) and achieved the performance result of 93.11 on testing set.

Gambback et al. (2009) compared three tagging approaches such as Hidden Markov Models (HMM), Support Vector Machines (SVM) and Maximum Entropy (ME) for Amharic language. They used the manually annotated corpus that is developed at the Ethiopian Language Research Center (ELRC) of Addis Ababa University with tag-set of 30 and achieved average accuracies (after 10-fold cross validation) of 85.56, 88.30, and 87.87 for the TnT, SVM and ME-based taggers, respectively for the ELRC tag-set.

Martha and Menzel (2009) have conducted POS tagging experiments for Amharic language for the purpose of language modeling. They used total of 21000 token of which 95% (19950 tokens) used for training and the rest for testing. The achieved accuracy of 82.99% and 85.50% on TnT and SVM-based taggers using 10 fold cross validation. They developed TnT and SVM based taggers and compared the taggers in terms of performance, tagging speed and memory requirement. Their experiments indicated that SVM based taggers have better performance in terms of accuracy than TnT based taggers and TnT based taggers are more efficient in terms of speed and memory requirement.
Martha, Solomon and Laurent (2011) conducted experiment on part of speech tagging for Amharic language to identify best method for under resourced and morphologically rich languages like Amharic. They reported that memory based tagger (MBT) performs better than Support Vector Machine (SVMTOOL) and Conditional Random field (CRF) based taggers.

Million and Getachew (2011) developed POS tagger for Oromo Language using Hidden Markov model approach and implemented bigram and unigram model of Viterbi Algorithm on 159 sentences (with a total of 1621 words) that were manually annotated sample corpus and 17 tag-set. The performance result of their tagger with tenfold cross validation was 87.58 for unigram and 91.97 for bigram.

Abraham, Degen and Xiaoxia (2014) have carried out another experiment on POS tagging for Oromo language using Maximum entropy Markov Model and used a manually annotated corpus of 452 sentences (total of 6094 words) with 33 tag-set and achieved accuracy of 93.01% which is evaluated by using tenfold cross validation.

Zelalem and Yaregal (2014) developed POS tagger for Kafi-noonoo text using a hybrid approach which is the combination of rule based and stochastic approach. They carried out their experiment on 90% of training corpus and achieved accuracy of 80.47%.

But there is no such research made for one of Ethiopian languages like Wolaita. The researcher aims to explore possibility of developing part of speech tagger for Wolaita language using supervised and semi-supervised corpus based POS tagging approaches. This approach is chosen due to availability of tools that are developed based on the chosen approach and the approach is widely studied for other languages and language independence of the tools.
CHAPTER THREE

THE GRAMMAR OF WOLAITA LANGUAGE

3.1 Introduction

This chapter is devoted to describe Wolaita language, explain and discuss word classes such as noun, verb, adjectives, and adverbs etc. of Wolaita language and to identify tag sets which are used for the development of the taggers. The study of every natural language requires understanding different levels of language processing like phonology, morphology, lexical, syntax, and semantics which are used to analyze languages at sound, word, phrase and sentence level. Each of the levels of natural language processing is described shortly in the following paragraph (Jurafsky and Martine, 2005).

One of the levels used in the analysis of natural language processing is **lexical analysis**. Lexical analysis is used to interpret the meaning of individual words and it is used for word level understanding. Part of speech tagging plays great role at this level by assigning single part of speech tag to each word and the most probable part of speech to words that have more than one part of speech based on the context in which the words occur (Allen, 1995). In the following paragraph description of Wolaita language like writing system it uses will be presented shortly.

Wolaita is a North Omotic language spoken in the Wolaita Zone and some other parts of the Southern Nations, Nationalities, and People's Region of Ethiopia by between 1.6 and 2 million people. Wolaita was first written in the 1940s by a team of missionaries led by Dr. Bruce Adams. They translated the Bible into Wolaita in 2002 (Wokasa, 2008).

The language is used in social, political and economic activity. At primary school level, the language is used as medium of instruction and taught as a subject in secondary and high school. Currently, the language is offered as a subject and department at Wolaita Soddo University.

The language is used to be written by using Amharic script (fidel) before 1993. But after 1993, the language is being written using Latin script. There are a lot of translations that have been made so far from Amharic to Wolaita and from English to Wolaita using Amharic script as well.
as Latin script and most of the translations made are religious books and materials. Since the language is used as a means of communication in government offices, and also serving as working language at wolaita Zone, there are huge number of documents and news articles produced and stored in this language both in hard and soft copy.

Wolaita alphabet and pronunciation


Vowels

a  á e  é i  í o  ó u  ú
[a]  [a:]  [e]  [e:]  [a]  [i]  [o]  [o:]  [u]  [u:]

Consonants

b  c  C  d  D  g  h  nh  j  k  K  l  L  m  M
[b]  [tʃ]  [tʃ’]  [d]  [d]  [ɡ]  [h]  [h]  [dʒ]  [k]  [k’]  [l]  [l]  [m]  [m’]

n  N  p  P  r  s  sh  t  T  w  y  z  zh  7
[n]  [n’]  [p]  [p’]  [r]  [s]  [ʃ]  [t]  [t’]  [w]  [j]  [z]  [ʒ]  [ʔ]

The sound written nh, a nasalized glottal fricative, is very rare and only appears in a few words. The sound written D is said by some scholars to be an implosive, but Wakasa calls it glottalized (Wakasa, 2008).

3.2 Word formation in Wolaita

According to Lamberti and Sottile (1997), words are formed in wolaita language by the two word formation processes called affixation (by which affixes are added to root words) and compounding which is the way of joining of two linguistic forms that have different function. Lamberti and Sottile (1997) also reported that, of the three types of affixes (prefix, infix and suffix), Wolaita uses suffixation to form words and prefixes and infixes are not used as word formation in Wolaita language.
### 3.3 Category of words in Wolaita

List of Wolaita word classes by Wakasa (2008) are shown in the following Figure 3.1

<table>
<thead>
<tr>
<th>Common nouns</th>
<th>Independent indeclinable</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Adverbal common nouns</td>
<td>- nominal indeclinable</td>
</tr>
<tr>
<td>- Semi-independent common nouns</td>
<td>- Interjections</td>
</tr>
<tr>
<td>- special types of common nouns</td>
<td>- conjunctions</td>
</tr>
<tr>
<td>Proper nouns</td>
<td>Dependent indeclinable</td>
</tr>
<tr>
<td>- place-name nouns</td>
<td>Verbs</td>
</tr>
<tr>
<td>- person-name nouns</td>
<td>- main verbs</td>
</tr>
<tr>
<td>Numerical Expressions</td>
<td>- imperfective</td>
</tr>
<tr>
<td>- numerals</td>
<td>- affirmative declarative imperfective</td>
</tr>
<tr>
<td>- compound numerical expressions</td>
<td>- negative declarative imperfective</td>
</tr>
<tr>
<td>- simple numerical expressions</td>
<td>- interrogative imperfective</td>
</tr>
<tr>
<td>- ordinal numbers</td>
<td>- past form for a happy situation</td>
</tr>
<tr>
<td>- cardinal numbers</td>
<td>- perfective</td>
</tr>
<tr>
<td>Personal pronouns</td>
<td>- affirmative declarative perfective</td>
</tr>
<tr>
<td>- reflexive pronouns</td>
<td>- negative declarative perfective</td>
</tr>
<tr>
<td>- the interrogative who</td>
<td>- interrogative perfective</td>
</tr>
<tr>
<td>Nominalizers</td>
<td>- future</td>
</tr>
<tr>
<td>- gaa and geeta</td>
<td>- optatives</td>
</tr>
<tr>
<td>- ro</td>
<td>- affirmative optatives</td>
</tr>
<tr>
<td>- nno</td>
<td>Negative optatives</td>
</tr>
<tr>
<td>- nta</td>
<td>- peripheral interrogative forms</td>
</tr>
<tr>
<td>Demonstrative Expressions</td>
<td>- subordinate verbs</td>
</tr>
<tr>
<td>- demonstrative determiners</td>
<td>- converses</td>
</tr>
<tr>
<td>- demonstrative pronouns</td>
<td>- simultaneous</td>
</tr>
<tr>
<td>Interrogative expressions</td>
<td>- relative</td>
</tr>
<tr>
<td>- who, what, where, how many</td>
<td>- true relatives</td>
</tr>
<tr>
<td>- interrogative nominal</td>
<td>- derived relatives</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interrogative verbs</th>
<th>-interrogative verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postpositions</td>
<td>-postpositions</td>
</tr>
<tr>
<td>Adverbs</td>
<td>-adverbs</td>
</tr>
<tr>
<td>Preverbs</td>
<td>-preverbs</td>
</tr>
<tr>
<td>Deverbal nominal</td>
<td>-deverbal nominal</td>
</tr>
<tr>
<td>Bawa ‘not present’</td>
<td>-future infinitives</td>
</tr>
</tbody>
</table>

Figure 3.1 word class of Wolaita (Wakasa, 2008)

The following list shows some of Wolaita word categories and their short description (Wakasa 2008):

Common nouns: They are defined morphologically and sub classified in to four classes based on their inflection and brief explanation about common nouns is given below under section 3.3.1.

Proper nouns: Proper noun is not defined morphologically or syntactically but it label for words referring to specific individuals or unique entities. In Wolaita proper nouns may be common nouns, place-name nouns or person name nouns.

Numerical expressions: are used to express numbers in the language. Cardinal and ordinal numbers are included here.

Personal pronouns: are used to express personal nouns in the language.

Postpositions: are words that are suffixed which have the same meaning as English prepositions.

Demonstrative expressions: contains both demonstrative determiners and demonstrative pronouns.

Adjectives: are words that qualify nouns

Adverbs: are words that qualify verbs
Conjunctions: are words that are used to connect two sentences or words.

Interjections: are used to express emotion in the language.

Verbs: are words that are used expressed action or state of being.

The main or basic classes of words in Wolaita language (nouns, Adjectives, Adverbs, and Verbs) are described below. But the detailed sub classification of word class like noun and verb and other word classes is presented under Wolaita tag-set in section 3.4.

3.3.1 Noun

Words under noun category in Wolaita are characterized by:

- They can take bound morphemes like (–a-ta,–e-ta,–o-ta) to form plural.
- They are used as subject, object and predicate.
- They can be modified for number, gender and case (Wakasa, 2008).

Based on the endings they take for their inflection, Wolaita nouns are divided into four classes. The first class contains words that end in –a in the absolute case and have stress on the last syllable.

Examples of such words are:

- 7asa 'person'
- 7aa wa 'father'
- ' tuma 'darkness'

The second class of nouns includes nouns that have their absolute case ending –iya and the stress on the syllable before the last one. Examples of such words include:

- bitan ni ya 'married man'
- 7ayfiya 'eye'
- kusshiya 'hand'

The third class of nouns includes nouns that end in –uwa in absolute case.

Examples of such words include:
The fourth class of noun consists those words that end in –(i)yu in their absolute case.

Examples of such words include:

- na7yu 'girl'
- bollotiyu 'mother-in-law'
- bakuliyu 'she-mule'

### 3.3.1.1 Gender

As indicated by Lamberti and Sottile (1997) Wolaita, like other languages, exhibits two types of genders i.e. masculine and feminine. According to him, nouns that belong to the fourth class are feminine and the rest (first, second and third class) belong to masculine. Feminine and masculine differ from each other by their endings. While feminine ends in –u, masculine ends in –a in absolute case.

Example:

dorsa 'sheep', masculine vs. dorsu 'sheep', feminine

deshu 'goat', masculine vs. deshu 'goat', feminine

ha-ge 'this, masculine' vs. ha-nna 'this, feminine

taa-gaa 'he/it is mine' vs. taa-ro 'she/it is mine'

### 3.3.1.2 Number

According to Lamberti and Sottile (1997), Wolaita noun contains singular and plural. The plural are formed by using suffixes and singular contains the basic form of the word. The first class of nouns forms their plural by means of the ending –a-ta(in the absolute case).for example:

Noun (Singular) ------Noun (plural)
Those of the second class of nouns exhibit the ending –e-ta during plural formation. Example:

Noun (Singular) === Noun (plural)
har-iya 'donkey' === har-e-ta 'donkeys'
7org-iya 'he-goat' === 7org-e-ta 'he-goats'
laagg-iya 'friend' === laagg-e-ta 'Friends'

The third class of nouns are characterized by the ending –o-ta. Example:

Noun (Singular) === Noun (plural)
7od-uwa 'tale' === 7od-o-ta 'tales'
word-uwa 'lie' === word-o-ta 'lies'

Finally, nouns in the fourth class which consist of terms for female beings assume the plural form of their masculine counterpart. Example:

Noun (Singular) ==== Noun (plural)
7imatt-iyu 'female guest' === 7imatt-a-ta 'female guests'
boogaanc-iyu 'female robber' === boogaanc-a-ta 'female robbers'
laagg-iyu 'female friend' === laagg-e-ta 'female friends'

Feminine noun that does not have any masculine counterpart are similar with nouns of second class and ends in –e-ta. Example:

Noun (Singular)====== Noun (plural)
macc-iyu 'wife' == macc-e-ta 'wives'

misshir-iyu 'married woman' == misshir-e-ta 'married women'

sam-iyu 'agreement' == sam-e-ta 'agreements'

3.3.2 Adjectives

According to Lamberti and Sottile (1997), adjectives in Wolaita language are used to qualify nouns and they come before the noun they qualify. They are also defined as words that modify nouns by expressing their qualities, color, size etc. Adjectives in Wolaita language end in -a, -iya or -uw. Examples of adjectives in Wolaita are shown below.

**Adjectives ending in -a:**

geessh-a 'clean'

cinc-a 'clever'

kaant-a 'short'

**Adjectives ending in -iya:**

mal-iya 'sweet'

haankett-iya 'violent'

yesshiss-iya 'fierce'

**Adjectives ending in -uwa:**

Lo7-uwa 'good/nice'

yuush-uwa 'round'

Adjectives in Wolaita language precede the noun they modify and remain unchanged when used in attributive position because they do not have to agree in Wolaita with their governing
noun either in gender or in number or in case. But most adjectives ending in –uwa and a few in –iya are replaced by the endings -o and -e respectively. Example:

Lo7-uwa 'good/nice' ==_ lo7-o asa 'a good person'

haah-uwa 'wide/far' ==_ haah-o sohuwa 'a far place'

luul-iya 'straight' ==_ luul-e 7ogiya 'a straight road'

Lamberti and Sottile( 1997) also reported that when adjectives are used in predicative position, -uwa will be changed to -o. for example:

Lo7-uwa 'nice/good' ==_ lo7-o:

he bitanney lo7-o 'this man is/was good'

he gelaawiya lo7-o 'that girl is/was nice'

But every word that comes before a noun is not necessarily an adjective. For example, in the sentence Ha dorsai taga “this sheep is mine “here the word Ha “this” has the role of adjective but it is demonstrative determiner (Wakasa, 2008).

3.3.3 Verbs

As pointed by Wakasa (2008), verbs of Wolatita, like most Ethiopian languages, are very complex. Wolaita verbs usually have consonant-vowel-consonant sequence. For example: gel- 'enter' moog- 'bury' ker- 'split'. Some of Wolaita verbs are borrowed from Amharic language. Examples: 7azzaz 'command', nabbab- 'read', kassas- 'accuse'

Verbs in Wolaita language exhibit a very complex inflection system depending on mood, tense, kind of action and aspect. For example if we take verb 7imm-a (give), it has the following inflection for past tense:

7imm-a:si --- I gave

7imm-adasa – you gave
7imm-a:su – she gave
7imm-i:si ---- he gave
7imm-ida --- we gave
7imm-ideta – you gave (plural)
7imm-idosona – they gave

Just like that of Amharic language, Wolaita verbs are found at the end of the sentence and suffix bound morphemes which help to indicate the subject of the sentence in the sentences shown below.

- Na7-yamayuwa shama-su 'the girl bought the cloth'
- neenib kawuwa ma-dasa 'you eat your dinner'
- tani 7osuwa wursa-si 'I finished my job'

In the above three sentences, shama-su, ma-dasa, wursa-si are verbs. The bound morphemes {-su},{-dasa},{-si} show 3rd person, 2nd person and 1st person pronouns that are used as subjects of the sentence. Verbs in Wolaita language change their shape for person, gender, number and time by attaching suffixes (Wakasa, 2008).

3.3.4. Adverbs

Adverbs are words that modify verbs.

Examples of adverbs in Wolaita are:

Eesuwan ‘quickly’----- Na7i timirte ketappe esuwan yiis 'he came from school quickly'

Lodan ‘slowly’-------- Bitane kamiya lodan lages 'The man drives the car slowly'

Kase ‘before’-------na77ia ta yanappe kase gakis 'the boy reached before me'

Coo ‘freely’---- Na7itawu kata coo imiis 'he gave me the food freely'
As we can see from the above example, adverbs (words in bold) come before verbs (yiis ‘come’, lages ‘drives’, gakis ‘reached’, and imis ‘gave’) and qualify them (Wakasa, 2008).

In addition to word classes described here, there are also other types of word classes in Wolaita language. They are proper nouns, numerical expressions, personal pronouns, demonstrative expressions, interrogative expressions and postpositions, interjections and sub classes of verb are not discussed here and are presented in the following section (Wokasa, 2008).

### 3.4. Wolaita TagSets

Determining POS tags and tag set of the language is an important step in the process of development of POS tagger which is used to annotate corpus in that language. After the determination of the tags and tag sets, the corpus was manually annotated by expert in language based on the identified tags and that manually annotated corpus will be used as a source of information or knowledge for the tagger. In the next part, tags for Wolaita word classes are presented. The POS tags used for explanation in this section is in the form of word/tag followed by a single space. For example:

**bitánee/NN bullúkkwu/NNDáddiis/VB** ‘The man wove the blanket.’

### 3.4.1 Tags for Nouns

As it has been discussed, nouns in Wolaita language can be common nouns or proper nouns and nouns are also singular or plural. Nouns also indicate gender, number and case. For the purpose of this study, common and proper nouns are included as noun singular and noun plural i.e. common nouns singular or mass and proper nouns are treated as singular nouns (NN), and common and proper nouns that are plural are represented as NNS. There are also other classes of nouns in addition to this class like nouns not separated from post postpositions, Verbal nouns, adverbial nouns and nouns suffixed with conjunctions. Each of them was discussed separately. Examples of such noun types are given below:

**na7ái/NN meeCCído maayójí/NN booTTíis.** ‘The cloth that the boy washed became white.’
mishiriya/NN bambariya/NN daaTasu. ‘The woman ground the red paper.’

7irlai/NN bukiiniNogééNhalliTiis. ‘When it rained, the road slipped.’

na7iya/NN maalládo TalKiya/NN ho77ásu. ‘The girl sunbathed in the morning.’

Naati/NNS keettáa/NN giddóóní Cábbottoosona. ‘The children are chattering in the house.’

Heezú 7ishatí/NNS dóosona. ‘There are three brothers.’

7umáa/NN haattáa/NN waDDádápinnaas. ‘I swam across the Omo River.’

ToPPée/NN ló77o biittá/NN. ‘Ethiopia is a good country.’

For the purpose of this study, days of the week and months of the year are also tagged as common nouns.

3.4.2 Other types of nouns

Here nouns other than common and proper nouns will be discussed. This category includes nouns not separated from postpositions, adverbial nouns, verbal nouns and nouns suffixed with conjunctions.

3.4.2.1 Tags for Nouns with adverbial ending

Here Stems of some common nouns take the endings like –i, -u and -iyo and they are used adverbially

He biittí/NA báas. ‘I went to that land.’

7usháccá bággau/NA simmá. ‘Turn to the right!’

Killifióy/NA bá. ‘Go around’

3.4.2.2 Tags for Postpositions:

Postpositions in Wolaita language are bound or non-autonomous words. There are eight postpositions in Wolaita: -u‘for, to’, -yyó‘for, to’, -kkó‘toward’, -ppé‘from’, -rā‘with’, -ssi‘for,
to’, -ni’in, at, by’, and -daani’like’. All of them are similar to “prepositions” in English languages. Example of them is given in the following sentences.

láíttaappe/NP 7issito7asái ‘Once a year people get together and . . .’

woláíttí 7alamúyyo/NP yelétabiittá. ‘Wolaytta is a birthplace for Alemu.’

7usúppunaassí/NP támmai paCa ‘It is 10 minutes to 12 o’clock

hegéé dadáani/NP giKogiyogéé 7álbee? ‘What is that, what one calls (say).’

protection by thunder?’

ta 7íshái maTáápaa 7ímmídoi balággiyassa/NP. ‘It is to his own friend that my brother gave the book.’

tá sho7éttido 7anjúllóona/NP. ‘It was by Anjulo that I was beaten.’

7i bitániyakko/NP bées. ‘He goes to the man’

The other type of noun under this category is nouns suffixed with conjunctions. The conjunctions are not separated from these nouns. Some of examples of these kinds of nouns are:

tanne/NC ta 7íshái timirte keta bida ‘I and my brother went to school’

tayaasi, ta 7íshaikka/NC yiis ‘I have come and my brother has come too’

The other class of noun under this category is postpositional phrase which is combination of postposition and independent words like ’kka’ and’nne’ are also included in nouns suffixed with conjunction(NC). Example of this tape is given here:

Bitániyayyo doonáarakka/NC sutaigogis ‘the man bleeds with his mouth too’

7i micciyo tohuwáaníkka/NC bocciis ‘he attached her by his feet too’
3.4.2.3 Tags for Verbal Nouns
These are nouns formed by adding non autonomous words like -gaa, -gee to verbs. Example of such nouns is given below:

**naagiyagéé** /NV pengiya dooyiis. ‘The person who watches (i.e. gatekeeper) opened the door.’

dapagiyogéé/NV 7ái-ée? ‘What is what is called (what one says) *dapa*?’

3.4.3 Tags for Personal pronouns
tá/PP heezzú sunkuruútuwa kóyyais. ‘I want three onions.’

tááni /PP heezzú sunkuruútuwa kóyyais. ‘I want three onions.’

né/PP múle káwwó báKKákká ‘You would never hit a king.’

núúní /PP hagáá mááCanau dandayókko. ‘We cannot judge this.’

7inté/PP yááni minnité. ‘You (Pl) be strong in that way’

7i/PP múle káwwó báKKena ‘He would never hit a king.’

7a/PP múle káwwó báKKuku ‘She would never hit a king.’

7etí/PP múle káwwó báKKokona ‘They would never hit a king.’

3.4.4 Tags for Demonstrative Determiners
According to Wokasa (2008), there are five demonstrative determiners in Wolaita language that serve as bases in various demonstrative expressions. They are ha ‘this’, he ‘that’, hi ‘that’, há’in the nearer place’, and yá’in the remoter place’. Examples of them are given below.

**ha/DD** godalíyau 7ái 7oottáí? ‘Hey this hyena, what are you doing?’

**ha/DD** Taliya Cammáusu. ‘This medicine is bitter.’

**he/DD** bitánee 7óónée? ‘Who is that man?’

**yá/DD** sóobí 7erénná. ‘He has not gone to that house (of the two).’
3.4.5 Tags for Demonstrative Pronouns

They are the result of the combination of demonstrative determiners and nominalizers (Wakasa, 2008). They are hagáá ‘this thing’, hegáá ‘that thing’, hannó ‘this thing fm.’, hinnó ‘that thing fm.’, hageetá,’those things’hegeetá ‘these things’. Examples of sentences with personal pronouns are given below:

Laa hagéé/DP 7áibée? ‘Hey, what is this?’

ha tóhoi hagáá/DP tóhuwa. ‘This leg is the leg of this (table, etc.).’

nééni hegáá/DP 7aggá. ‘You, stop that!’

hanná/DP mííziya_neerií? ‘Is this cow yours?’

3.4.6 Tags for WH-pronouns and interrogative expressions

In this section, all wh-pronouns and various interrogative expressions were discussed. Words categorized under this are: 7ai’what’, 7au’where’ and 7ooni’who’. Examples of such class are shown in the following sentences.

7óóni/WP yíidee? ‘Who did come?’

7áí /WP be7-ádii? ‘What did you see?’

7áu/WP bái? ‘Where are you going?’

3.4.7 Tags for Numerical Expressions

In this section various numerical expressions are described. It contains ordinal and cardinal numbers. Examples of sentences that include numerical expressions are shown below:

naa77ú/CD bitáneta be7-áas. ‘I saw two men.’

Hagéé tâu heezzánto/OD na7á. ‘This is the third child for me’

Bóórai hagéé tammántuwa/OD. ‘This ox is the tenth one’

*Compound Numeral Expressions (Combinations of Numerals)*
For Example, the numbers “60”, “70”, “80” and “90” are expressed as 7usúppuntámm-á “six tens”, lááppuntámm-á“Seven tens”, hóspuntámm-á“eight tens”, and 7uddúpun támm-á “nine tens”, respectively. For the purpose of this study, compound numerical expressions will be treated separately. Examples of compound numerical expressions are described below.

7usúppun/CD támmú/CD 7asatíyídosona. ‘Sixty people came.’

naa77ú/CD Téétánné/NC hástámánné/NC 7oiddú/CD shá7á/NN. ‘234000’

3.4.8 Tags for Adverbs

Adverbs in Wolaita come before verbs and are used to modify verbs.

Examples of adverbs are given below:

Táání haasayíshii ninééní sírPi/AD gá. ‘Be quiet and pay attention while I am speaking.’

tóKKu/AD gá. ‘Be elevated!’

nééní Cóo/AD haasayáasa. ‘You speak in vain.’

3.4.9 Tags for Interjections

Interjections, in Wolaita language, are autonomous words that are used to express emotion, to address or respond to the hearer(s), or to fill blanks in a sentence or between sentences caused by various reasons (Wokasa, 2008). Examples of them are:

7áne/UJ 7í woTTíini be7aná. ‘Let’s see him running’

‘nééní zíno báádiis?’ cíi/UJ bábe7íkke.’ ‘Did you go yesterday?’ ‘No, I did not go.’

laa/UJ hágéé 7áibée? ‘Hey, what is this?’

bii/UJ tamiccéewáanaí? ‘Hey my sister, how are you?’
3.4.10 Tags for Conjunctions

Conjunctions in Wolaita language are autonomous words whose function is to introduce or insert a linguistic unit (word, phrase, etc.) into an utterance that is complete by itself. Examples of sentences with conjunctions in Wolaita language is given below:

nééni  wóíkkó/CC  7ibó. ‘You (go), or let him go.’

Heezzánto 7ogée Kássí/CC  làba géétettees. ‘The third way is called “labä”.’

3.4.11 Tags for Verbs

For the purpose of this study, Wolaita verbs are classified into five. They are main verbs, subordinate verbs, relative verbs, infixes and infinitives.

3.4.11.1 Tag for Main Verbs

This section includes different verb forms used as predicates of main clauses, which are usually at the end of sentences, are discussed. It also includes imperfective (which again includes affirmative declarative imperfective, negative declarative imperfective and interrogative imperfective), perfective (which also includes affirmative declarative imperfective, negative declarative imperfective and interrogative imperfective), future and optative. Examples of main verbs are given below:

táání káttaa máis/VB. ‘I was having a meal’

ha77í nabbábuwa meezetáis/VB. ‘Now I am practicing reading.’

dawwálee dawwaléttees/VB. ‘The bell is ringing.’

táání Kúmaa míkke/VB. ‘I was not having a meal.’

7ogíyani 7áí haasayéetii/VB? ‘What were you talking on the road?’

na7átettaani kaassáa siiKíkkíi/VB? ‘Didn’t you love playing when you were a child?’

táná Talée Cammíis/VB. ‘This medicine was bitter’

káhoi giigídee/VB? ‘Is the dinner ready?’
néení mínna 7oottikkó ló77o miishsháa go7éttna/VB. ‘If you work hard, you will find
good payment.’

3.4.11.2 Tags for Subordinate verbs.
In this section, different verb forms that are not used as predicates of main clauses are
discussed. Some examples of such verbs are given below:

ló77o boináas hammádá/VS kattáas. ‘I bought good taros, and I cooked (them).’

7asái shiiKídi/VS kúúyiis. ‘People gathered, and decided.’

gaCinnádá/VS Tiihéttada/VS 7órdaasu. ‘After giving birth, she was fed, and became fat.’

Verbs that are not separated from postpositions and conjunctions are also included here:

7i tamaaridínne/VS eridi diCCis. ‘He grow educated and wise’

7i orde gidikkókká/VS assai 7a yayena. ‘Even if he is fat, no body fears him’

3.4.11.3 Tags for Relative verbs.
A relative form is a verb form that is used to modify its following nominal. Most of the time
relative verbs in Wolaita come before noun. Examples of relative verbs are given below:

Kasétá 7immídó/VR maayúwa wáátadii? ‘What did you do about the clothes that I gave
before?’

yíída/VR bitániya be7adi? ‘Did you see the man who came?’

táání 7ashúwa muTido/VR mashsháa 7epá. ‘Take the knife with which I cut the meat
into small pieces.’

táání 7issíppé bíído/VS bitániya. ‘The man with whom I went together’
3.4.11.4 Tags for Infinitive verbs

In Wolaita language, Infinitives are formed regularly from every verb stem, and they are, therefore, can be regarded as verbs. Examples of Infinitives are given below:

ha buddeenáa maanáí/VT 7óónee?  ‘Who will eat this Injera’

ha záápiya KanTanáu/VT waissées. ‘It is difficult to cut this tree.’

Káwuwa maanáu/VT bíis. ‘He went in order to eat dinner.’

ta yaanái neni Kúma maanáashiina/VT. ‘I will come until you eat’

gaCíniyo 7anjaná/VT báasu. ‘She went to bless the woman in child bed.’

3.4.11.5 Tags for verbs with Infix

Infixes are independent linguistic forms that are inserted before different verb endings. Such forms are called “infixes” (Wakasa, 2008). Examples of infix verbs are given below:

7í gákkiyodé táání báiccaas/VI. ‘I had already gone when he reached.’

nééni yiyowodé táání Kúma márgaas/VI. ‘When you came, I had already taken a meal.’

tawarái ta mááttaa gussírgiis/VI. ‘The cat spilled my milk completely.’

hagéé 7írai táná sóóní peeshshírgiis/VI. ‘This rain made me spend the day at home completely.’

3.4.12 Tags for punctuation marks in the language

A tag for all Punctuation marks in the language. ‘Full stop’, ‘comma’ ‘Exclamation mark’ and ‘question mark ’{., , !, ?}. Example is given below:

ha buddeenáa maanáí/VT 7óónee?/PU  ‘Who will eat this Injera?’

ha záápiya KanTanáu/VT waissées./PU  ‘It is difficult to cut this tree.’
Different literatures have been analyzed and reviewed on Wolaita language to understand the language structure and to categorize words in the language into, noun, verb, adjective, adverb, interjection etc.

After reviewing grammars of Wolaita language, rules and word classes, the following tag sets are proposed for the purpose of this work. Since there is no tag set prepared for the purpose of natural language processing for Wolaita language except studying word classes for Wolaita language by different language experts, the following 22 tags have been identified and used for the purpose of this study.

Tags are given their name according to the name and meaning they have in Wolaita language. Most of the tags are based on the work of Wakasa (2008) on Wolaita word class. The list of the tag sets together with their descriptions and example of Wolaita words under each category and tags identified and used for the purpose of this research are shown in the following table.

<table>
<thead>
<tr>
<th>No</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>NNS</td>
<td>A tag for all common and proper nouns which are Plural ,naatí ‘children’, ‘wives’ 7asata ‘person’, 7aawata ‘father’ ‘tumata ‘darknesses’</td>
</tr>
<tr>
<td>Page</td>
<td>Tag</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>NV</td>
<td>A tag for all nouns derived from verbs, mazggapiyoge ‘writing’ lágg-iyogéé ‘deriving’</td>
</tr>
<tr>
<td>5</td>
<td>DD</td>
<td>A tag for all Demonstrative determiners, ha ‘this’, he ‘that’, yá ‘there’, há ‘here’</td>
</tr>
<tr>
<td>6</td>
<td>DP</td>
<td>A tag for all Demonstrative pronouns, hagáá ‘this thing m. sg’, hégáá ‘that thing m. sg’, hannó/hinnó ‘this thing f.sg’, hageetá ‘these things pl.’, hegeetá ‘those things pl’</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
<td>A tag for all nouns not separated from conjunctions. tanne’l and’, bitanekka’the man too’, mikiniyáteekka ‘the reason is that’</td>
</tr>
<tr>
<td>10</td>
<td>WP</td>
<td>A tag for all WH-pronouns and interrogative expressions, 7óóná/7óóni’whom’, 7ái ‘what’, 7áu ‘where’</td>
</tr>
<tr>
<td>11</td>
<td>CC</td>
<td>A tag for all Conjunctions, woi/woikko ‘or’ simi’after that or then’, sima</td>
</tr>
<tr>
<td>12</td>
<td>CD</td>
<td>&quot;therefore&quot;, Ká/Kassí ‘furthermore’, shiíni ‘but’, gidíkkókká ‘even though’</td>
</tr>
<tr>
<td>13</td>
<td>OD</td>
<td>A tag for all Cardinal Numbers, 7istá ‘one’, naa77á ‘two’</td>
</tr>
<tr>
<td>14</td>
<td>AD</td>
<td>A tag for all Ordinal Numbers. kóíruwa/koiro’1st’, naa77ant-á/a/naa77ânto ‘2nd’</td>
</tr>
<tr>
<td>17</td>
<td>VB</td>
<td>A tag for all Main verbs which includes positive and negative imperfective, negative and positive perfective, interrogative perfectives, future, optative forms, máis ‘I eat’, KanTáis ‘I cut’, miikké ‘I do not eat’, haasayétii ‘talking 2p. pl’, gákkusú ‘she reached’, yáasi ‘I came’, demm-ábe7ikke ‘I did not find’, yaaná ‘will come’, Teellá ‘see’, büutee ‘will you go?’, wáánuutée ‘what shall you do?’, 7immóo ‘shall I give?’, biinóo ‘shall we go?’, 7oottáshsha ‘work continuously’ ‘daapuráshsha’ ‘be tired completely’</td>
</tr>
</tbody>
</table>
| 18 | VS | A tag for all Subordinate verbs. These contain converb, simultaneous

| 19 | VT | A tag for all Infinitive verbs which contain future negative and other infinitives. ziNNaná ‘sleep’, maanáappe ‘eat from’, gákkanaassi ‘reach till’, 7ekkanáu ‘to take’, 7anjaná/u ‘to bless’, béénnai ‘see not’, 7oott-énn-aa-ni ‘work not’, súntiyakkonne ‘name if and’, miyadaani ‘as if eating’ |
| 20 | VI | A tag for all verb Infixes which include completive and durative infixes, mírgiis ‘eat completly’, 7eráiccaas ‘completly’, háíKKaiccarkii ‘die completely’, Tayíicciini ‘lost completely’, báiccaas ‘go completly’ |
| 22 | PU | A tag for all Punctuation marks in the language. ‘Full stop’, ‘comma’ ‘Exclamation mark’ and ‘question mark {., , !, ?}’ |

Table 3.1 tags and description of tags for Wolaita language

This chapter described Wolaita language, word classes of Wolaita language and tag sets of the language. Based on review of literature on word class of the language, about 22 tag sets were identified and described using example for each of them
CHAPTER FOUR

METHODOLOGY OF THE STUDY

4.1 Introduction
This chapter is devoted to explain methodology used in this study which includes source of data, sampling technique used for data collection, data preprocessing, preparation and evaluation techniques used for the accuracy of the tagger.

4.2 Data Preparation
The tagged corpus is the immediate requirement for different analyses in the field of Natural Language Processing (NLP). Most of the language processing works like part of speech (POS) tagging is in need of such large collection of texts, which provide a real, natural, native language of varying types. Annotation of corpora can be done at various levels through, Part of Speech, phrase or clause level, dependency level, etc. POS tagging is the basic step towards building an annotated corpus.

Since there is no manually tagged corpus which is useful for NLP tasks like POS tagging for wolaita language, it demands to collect and prepare such a corpus to conduct POS tagging experiment. For the purpose of this research, data is collected from text book which was used as a reference book in Wolaita Soddo University in college of Social science and humanity in the department of Wolaita language study. The book was prepared by Waru Alemayehu in 2006 E.C. The title of the book was ‘Xaafo Hiillaa-II’ writing skill-two and the book has 108 pages. About 200 Sentences are randomly selected from this text book and are manually tagged by PhD student (who is currently studying Wolaita Language) in the Institute of Ethiopian language studies of Addis Ababa University.

The data selected for the research is prepared and preprocessed manually and no special tool is used for data preprocessing. Data preprocessing includes: spelling checking, removing brackets, quotation marks and removing foreign words (English words) of which the researcher is not interested to deal with in this research.
To conduct experiment in this research, only small sample of about 200 sentences was selected due to there is no corpus annotated with part of speech for Wolaita language and it is also time consuming, laborious and expensive to tag large amount of corpus manually. Sisay (2005) used corpus of about 1000 tokens to conduct segmentation and POS tagging experiment for Amharic language and also Mesfin (2001), Yenewondim (2006) and Getachew and Million (2009) developed POS tagger using small amount of text (about 159 sentences) for Amharic and Afan Oromo Languages. TnT developers also reported known accuracy of 96% could be achieved by using only 1000 tokens.

The corpus collected and prepared in the previous stage was not in a suitable format to process by Natural Language Processing (NLP) tools. In order to be used by various NLP tools, it needs to be converted to a format that is suitable for them. This process of conversion is known as preprocessing and involves identifying sentence boundaries (Sentence Detection), and word forms or tokens (Tokenization). Both Tokenization and sentence detection were done using notepad++ and python programming language.

The corpus prepared to conduct experiment was divided into two sets, training and test set. The training set was used to develop the model and the test set was used to evaluate the performance of the model. The training set consists of 90 percent of for the total data (200 sentences) and is about 180 sentences and the testing set was about 10 percent of the total data (200 sentences) and is about 20 sentences. The test set was prepared by removing 20 sentences from the total manually tagged corpus and it was 10 percent of manually tagged corpus. Examples of untagged and tagged words are shown below.

|---|

Figure 4.1 Example of unprocessed Wolaita corpus

As indicated in Figure 4.1 the text selected is not preprocessed i.e. it contains foreign words brackets and incorrectly spelled words.
As indicated in Figure 4.2, the corpus is preprocessed i.e. foreign words in the brackets and the brackets removed, two words are correctly spelled. Finally the preprocessed corpus was manually tagged and tokenized. Tagged corpus is shown in Figure 4.3 below.

As indicated in figure 4.3 the corpus is tagged using correct POS tags after being preprocessed. Then the corpus is tokenized using python programming language since both of the tools selected for tagging in this thesis require token per line for training. The data format used by both of the tools is shown in Table 4.1 and 4.2.

**4.3 Evaluation Procedures**

The taggers were trained on training set which was prepared in the previous stage. The result obtained on the training set was evaluated by comparing it with manually tagged corpus. The taggers or the models were then tested on the test set with untagged data. The test was used to see how well the models or the taggers perform on unseen data. Finally, the output of the tagger was compared with that of manually tagged data. The performance of the tagger was evaluated by dividing the number of correctly tagged words to the total number of words in the test set. The performance evaluation was done using TnT (Brants, 200) tool by tnt-diff command.
The corpus developed consists of about 200 sentences. The training corpus which is 90% of the total corpus, was about 180 sentences and consists of 1184 tokens, 22 tags and the test corpus which is 10% of the total corpus, was about 20 sentences and contains 134 tokens, 17 tags.

### 4.4 Tools and Techniques used

In this thesis, tools which are freely available, simple and easy to use have been used to develop the taggers. Two tools used are TnT (Brants, 2000) and CRF++ (Taku, 2005). Both of the tools used are windows version. TnT (Brants, 2000) is based on Hidden Markov Model (HMM) and CRF++ (Taku, 2005) is Based Conditional Random Fields (CRFs).

#### 4.4.1 TnT (Traigams’n’ tagger)

Brants’s TnT tagger a statistical Part of speech (POS) tagger which can be trained on different languages and any tag set. The component for parameter generation was trained on tagged corpus. IT uses different methods of smoothing to handle unknown words. It is optimized for training on a large variety of corpora but not a particular language. The tagger can easily be adapted to new language, domain and tag set and it was also optimized for speed. It is an implementation of the Viterbi algorithm for second orders Markov models. It uses linear interpolation for smoothing in which the respective weights are determined by deleted interpolation. Unknown words are handled by suffix trie and successive abstraction. POS tagging accuracy reported for different languages is between 96% and 97%. This is comparable to state of the art result that is reported in the literature.

**File format used by TnT**

<table>
<thead>
<tr>
<th>%% tagged Wolaita corpus for TnT</th>
<th>%%Unntagged Wolaita corpus for TnT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazggapiyoge         NV</td>
<td>Mazggapiyoge</td>
</tr>
<tr>
<td>Guuggiyappe         NP</td>
<td>Guuggiyappe</td>
</tr>
<tr>
<td>soridi              VS</td>
<td>soridi</td>
</tr>
<tr>
<td>qoliyooga           VB</td>
<td>qoliyooga</td>
</tr>
<tr>
<td>.                   PU</td>
<td>.</td>
</tr>
</tbody>
</table>
TnT application consists of two steps: The first step is parameter generation and the second step is tagging.

1. **Parameter generation**

   This step is used to create model parameters from the tagged corpus. It is performed when we want to modify the model parameters by using large corpus. Parameter generation requires a tagged corpus to be tokens per line where tokens and tags are in separated columns. Training corpus has to be large and the accuracy of assigned tags has to high. Using large size of the corpus and highly accurate training corpus results in better performance of the tagger.

   The parameter generation is performed by using the command `tnt-para [options] <corpus file>` or `(tnt-para.exe [options] <corpus file> for windows version) where the <corpus file> is the file containing tagged training corpus. The program generates lexical and contextual frequency from the training corpus and stores them in two files in the current directory with the same
name as the corpus file but with the extensions .lex and .123 respectively for lexical and contextual frequencies.

[Options] can be one of the following:

- -i: which means ignore case; all upper case characters mapped in to lower case characters and lexicon will contain lower case entries only.
- -l: generate lexicon only; but by default lexicon and n-gram are generated.
- -n: generate n-gram only; without this option both lexicon and n-gram are generated.

2. Tagging

This tagging process requires the two model parameter files that contain lexical and contextual frequencies which are generated in the first step and the input file or test file in specific format suitable for the tool. In the tagging process the tagger, the tagger is started using the command tnt [options] model corpus or (tnt.exe [options] model corpus for windows version) where model is the language model to be used. The tagger will for the two models that are generated by tnt-para.exe i.e. model.lex and model.123 that contain lexical and contextual frequencies. Corpus is the file with text to be tagged and it should be in the format of one token per line. Option is one of the following:

- alength: use suffix trie of maximum suffix length to handle unknown words (default value is – a10).
- dmode: use sparse data mode with default value of (–d4) and it can be one of the following:
  
  1/c: replace zero frequencies with constant c. If d1/0.4, replace 0 with 0.4.
  
  2/c: add constant c to all frequencies.
- umode: use mode to handle unknown words with default value= u3. And has the following options:
  
  0: No unknown word is allowed. The tagger exits when detecting unknown words.
1: Take lexicon entry @UNKNOWN to determine lexical probabilities for unknown words.

2: combine statistics of all words to handle unknown words.

3: combine statistics of all words that occurred once to determine unknown words.

The following Figure shows the parameter generation (training) using TnT tool.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\user> cd C:\Int\tnt
C:\Int\tnt>tnt-para.exe --help
TnT-Para: Generate trigram parameters from corpus - Version 2.2
(C) 1993 - 2000 Thorsten Brants, thorsten@coli.uni-sb.de
usage: C:\Int\tnt\tnt-para.exe [options] <corpusfile> ...
  -c : encode capitalization in tag
  -h : show this message
  -H : ignore HTML tags for parameter generation
  -i : ignore case
  -l : generate lexicon only
  -n : generate ngrams only
  --o<name>: base name for output files, default-basename of corpus
  -v : generate verbose ngrams
default: generate lexicon and ngrams
<corpusfile> contains tagged corpus
gzip-ed or compress-ed files are recognized by the suffix .gz or .Z
to read from stdin, set filename to `-`
```

Figure 4.4 Training using TnT

**Counting differences (tnt-diff.exe)**

If we have a file that contains correct tags, we can use this command to compare the correct version with the one tagged by the tagger. The command used to compare correct version with statistically tagged one is `tnt-diff.exe [options] <original file> <new file>` where `<original file>` is the file that contains the correct assignment and the `<new file>` is the one tagged by the tagger. The output of this command gives something like:

Overall result:

Equal : 69 / 102 (67.65%)

Different: 33 / 102 (32.35%)
Which shows out of 102 tokens, 69 (67.65%) of statistically tagged are similar to the file with correct tag and 33 (32.35%) out of 102 are different from that of correct tag.

The following Figure shows tagging using TnT tool.

Counting tokens and types

TnT can also be used to count tokens, types and different tags in the corpus. This task is performed using the command `tnt-wc.exe [options] <corpus file>` where <corpus file> is the file containing corpus in the format of token per line and it can be either tagged or untagged. And [options] can be one of the following:

- `i`: Ignore upper upper/lower case of the tokens.
- l: count word types.

-t: count different tags.

-w: count word tokens.

For example, using the command `tnt-wc.exe -i -l -t -w fulwolaitacorpus.txt` give the output 1321 tokens, 730 types, 22 tags. Where -i for ignore upper and lower cases, -l for count word types, -t count different tags and -w count word tokens.

### 4.4.2 CRF++

CRF++ is a simple, customizable, and open source toolkit of Conditional Random Fields (CRF) for segmenting and labeling sequential data. CRF++ is applied to a variety of NLP tasks such as Part of Speech Tagging, Named Entity Recognition, Information Extraction, Text Chunking etc. CRF++ uses a combination of forward Viterbi and backward A* search algorithms (Lafferty et al., 2001).

According to Taku (2005), CRF++ has the following features:

- Can redefine feature sets
- Written in C++
- Fast training based on LBFGS, a quasi-Newton algorithm for large scale numerical optimization problem
- Less memory usage both in training and testing
- encoding/decoding in practical time
- Can perform n-best outputs
- Can perform single-best MIRA training
- Can output marginal probabilities for all candidates
Available as an open source software

Training and test file format

Like that of TnT, both the training file and the test file of CRF++ need to be in a particular format to carry out the tagging process properly. In General, training and test file must consist of multiple tokens. In addition, a token consists of multiple (but fixed-numbers) columns. The definition of tokens depends on task to be performed. However, in most of typical cases, they simply correspond to words. Each token must be represented in one line, with the columns separated by white space (spaces or tabular characters). A sequence of token becomes a sentence. To identify the boundary between sentences, an empty line is put.

We can give as many columns as we like, but the number of columns must be fixed through all tokens or it has to be the same in number for all of the tokens used. Furthermore, there are some kinds of arrangement among the columns. For example, 1st column is word; second column is POS tag third column is sub-category of POS and so on. There are two columns for each word. The first column contains the word and the second column contains part of speech associated with the word. Example of file format for CRF++ is given below:

<table>
<thead>
<tr>
<th>Mazggapiyoge</th>
<th>NV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guugggiyappe</td>
<td>NP</td>
</tr>
<tr>
<td>soridi</td>
<td>VS</td>
</tr>
<tr>
<td>qoliyooga</td>
<td>VB</td>
</tr>
<tr>
<td>.</td>
<td>PU</td>
</tr>
<tr>
<td>Qofatanne</td>
<td>NC</td>
</tr>
<tr>
<td>Naqaashata</td>
<td>NNS</td>
</tr>
<tr>
<td>qorada</td>
<td>VS</td>
</tr>
<tr>
<td>shiishshaasa</td>
<td>VB</td>
</tr>
</tbody>
</table>
Table 4.2 file format used by CRF++

Training (Encoding)

Training is performed by using crf_learn.exe command:

```
crf_learn.exe template train.txt model
```

Template is and the output of training prints the following screen:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\user>cd C:\CRF++\CRF++-0.58
C:\CRF++\CRF++-0.58>crf_learn.exe template train4.txt model
CRF++: Yet Another CRF Tool Kit
Copyright (C) 2005-2013 Taku Kudo, All rights reserved.
reading training data: 100..
Done!0.03 s
Number of sentences: 180
Number of features: 191107
Number of thread(s): 4
Freq: 1
eta: 0.00010
C: 1.00000
shrinking size: 20
iter=0 terr=0.96728 serr=1.00000 act=191107 obj=3737.50911 diff=1.00000
iter=1 terr=0.44128 serr=0.97802 act=191107 obj=3245.98099 diff=0.13151
```
iter=2 terr=0.44128 serr=0.97802 act=191107 obj=2768.69531 diff=0.14704
iter=3 terr=0.54195 serr=0.93407 act=191107 obj=2142.30342 diff=0.22624
iter=4 terr=0.44044 serr=0.93956 act=191107 obj=1909.55369 diff=0.10864
iter=5 terr=0.36997 serr=0.90110 act=191107 obj=1569.11628 diff=0.17828
iter=6 terr=0.53272 serr=1.00000 act=191107 obj=2745.39691 diff=0.74965
iter=7 terr=0.29362 serr=0.88462 act=191107 obj=1332.47507 diff=0.51465
iter=8 terr=0.31124 serr=0.87912 act=191107 obj=1118.29220 diff=0.16074
iter=9 terr=0.12500 serr=0.52198 act=191107 obj=814.82254 diff=0.27137
iter=10 terr=0.05201 serr=0.29121 act=191107 obj=757.97879 diff=0.06976
iter=11 terr=0.02852 serr=0.16484 act=191107 obj=722.63627 diff=0.04663
iter=12 terr=0.00252 serr=0.01648 act=191107 obj=679.30937 diff=0.05996
iter=13 terr=0.01846 serr=0.12088 act=191107 obj=674.59820 diff=0.00694
iter=14 terr=0.00168 serr=0.01099 act=191107 obj=654.40071 diff=0.02994
iter=15 terr=0.00084 serr=0.00549 act=191107 obj=650.28944 diff=0.00628
iter=16 terr=0.00000 serr=0.00000 act=191107 obj=645.26080 diff=0.00773
iter=17 terr=0.00000 serr=0.00000 act=191107 obj=642.57358 diff=0.00416
iter=18 terr=0.00000 serr=0.00000 act=191107 obj=640.09126 diff=0.00386
iter=19 terr=0.00000 serr=0.00000 act=191107 obj=639.26768 diff=0.00129
iter=20 terr=0.00000 serr=0.00000 act=191107 obj=638.62528 diff=0.00100
iter=21 terr=0.00000 serr=0.00000 act=191107 obj=638.11561 diff=0.00080
iter=22 terr=0.00000 serr=0.00000 act=191107 obj=637.69835 diff=0.00065
iter=23 terr=0.00000 serr=0.00000 act=191107 obj=637.59088 diff=0.00017
iter=24 terr=0.00000 serr=0.00000 act=191107 obj=637.51096 diff=0.00013
iter=25 terr=0.00000 serr=0.00000 act=191107 obj=637.52479 diff=0.00002
iter=26 terr=0.00000 serr=0.00000 act=191107 obj=637.46909 diff=0.00009
iter=27 terr=0.00000 serr=0.00000 act=191107 obj=637.42348 diff=0.00007

Done! 1.98 s

Table 4.3 Output of crf_learn

Where

- iter: number of iterations processed
- terr: error rate with respect to tags. (# of error tags/# of all tag)
- serr: error rate with respect to sentences. (# of error sentences/# of all sentences)
- obj: current object value. When this value converges to a fixed point, CRF++ stops the iteration.
- diff: relative difference from the previous object value.
Generally, there are 4 major parameters to control the training condition

-a CRF –L2 CRF-L1:

This is used to change the regularization algorithm. Default value is to L2. In General, L2 performs slightly better than L1, when the number of non-zero features in L1 is very smaller than that in L2

-c float:
This option used to change the hyper-parameter for the CRFs. With larger C value, CRF tends to overfed to the give training corpus. This parameter is used to have balance between over fitting and under fitting.

-f num:
This parameter sets the cut-off threshold for the features. CRF++ uses the features that occur no less than NUM times in the given training data. The default value is 1. When we apply CRF++ to large data, the number of unique features would become several millions.

-p num:
If the PC has multiple CPUs, you can make the training faster by using multi-threading. NUM is the number of threads.

**Testing (Decoding)**

Testing uses crf_test command: crf_test.exe -m model utest.txt > resultc.txt where –m model indicates the model developed in crf_learn and utest.txt indicates untagged test file and resultc.txt shows the result of tagging process.
Architecture of the tagging process that shows tasks performed in the process:

Corpus collected
(Unstagged file)

Annotation
(Files are manually tagged)

Tagged file

Untagged file

Tagger (Tool)

Corpus collected

Figure 4.6 Architecture of the tagging process

In general the tagging process includes: text data selection and preparation, manually tagging the text corpus, dividing the data into training and test set, training the tagger to develop the model, and testing the model with untagged test set. Description of the process of tagging or steps in developing the tagger model is briefly described below:

**Corpus or untagged file selection and preprocessing:**

As it was explained in this chapter, corpus is the most important component of Natural Language Processing task like part of speech tagging. For the purpose of this research, corpus of about 200 sentences is prepared to train the system.
Corpus Annotation
It is the process of adding linguistic information to an electronic corpus of written or spoken language data.

Annotated corpus
It is the output of the above corpus annotation process and it used to train the system or the model.

Training Data
It is part of the annotated corpus that contains linguistic information. It is used to train taggers to develop the model.

The model
IT is the system that is developed in the process of training and used for tagging process after being developed.

The tagger
It is the algorithm or software tool that is used to develop the model from the training data

Untagged file
It is part of the data that is not annotated and used to test the performance of the model which was developed in this process.

Tagged file
Is the data which is tagged by the new model developed in the training process and it is the output of untagged file which is feed to the system.

In this chapter, methodologies used was described from perspective of data selection, preparation and preprocessing the tools used in this thesis for the development of POS taggers for Wolaita language are presented. While TnT tool uses HMM based approach, CRF++ tool uses CRF based approaches. The two tools are selected because of their availability, simplicity, interest of the researcher in the tools and the fact that they have been used to develop POS taggers for other languages local and abroad and the language independence nature of the tools. Although Martha, Solomon and Laurent (2011) reported that Memory based Tagger tool has better performance over CRF++ for small data, it was not used in this thesis due to its unavailability and difficulty to install.
In addition to supervised tagging, semi-supervised tagging which uses labeled and unlabeled data was also experimented in this thesis. This approach is used to take the advantage of hugely available unlabeled data with small manually labeled data to develop taggers for under resourced languages.
5.1 Introduction
This chapter is mainly concerned to present, the experiments carried out using each tool, results of the experiment, discussion on results of experiment, error analysis and finally conclusion of the chapter.

5.2 The Experiment
In this section different experiments which were carried out using each tool are discussed. The first experiment was conducted to see the performance of the taggers using 10 fold cross-validation i.e. by dividing the data which contains 200 sentences which was manually tagged into 90% (180 sentences) training and 10% (20 sentences) testing and by taking training and test set from different parts of the corpus. The second experiment was carried out by selecting test set from the corpus systematically i.e. the 1st, the 11th, 21st, etc until the last 191th sentence for test set and the rest of sentences for training. The third experiment was conducted to see the effect of reducing tag set from 22 to 17 on the performance of the taggers. The forth experiment was conducted by using semi-supervised approach which uses both manually tagged and untagged data by tagging new 180 sentences using one of the best performing model developed in experiment two. Finally, fifth experiment was conducted by tagging additional 180 sentences by using the tagger which is developed in experiment four in order to increase the size of training data to 540 sentences again using semi-supervised approach.

5.2.1 Experiment one
This experiment was carried out to see the performance of the taggers on the corpus taking 10% of test data from different parts of the corpus and then to see the average performance of the taggers. The test data was selected in the manner of the first 20 sentences (sentence 1-sentence 20) as test set and the remaining 180 sentences for training, second 20 sentences (sentence 21 – sentence 40) were used for the test set and the rest of 180 sentences for training and so on until the last test set (sentence 181- sentence 200) for test set and the
remaining 180 sentences (sentence 1- sentence 180) for training. This experiment was carried out without parameter options during training for both of the methods (HMM and CRF). The result of experiment was shown below using the two tools.

5.2.1.1. Experiment 1 using TnT

<table>
<thead>
<tr>
<th>Test set</th>
<th>Training set</th>
<th>Total no. of tokens</th>
<th>No. of correctly tagged tokens</th>
<th>Accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>The 1st 20 sentences</td>
<td>The rest of 180 sentences</td>
<td>144</td>
<td>106</td>
<td>73.61</td>
</tr>
<tr>
<td>The sentences 21-40</td>
<td>&gt;&gt;</td>
<td>109</td>
<td>88</td>
<td>80.73</td>
</tr>
<tr>
<td>The sentences 41-60</td>
<td>&gt;&gt;</td>
<td>126</td>
<td>101</td>
<td>80.16</td>
</tr>
<tr>
<td>The sentences 61-80</td>
<td>&gt;&gt;</td>
<td>135</td>
<td>112</td>
<td>82.96</td>
</tr>
<tr>
<td>The sentences 81-100</td>
<td>&gt;&gt;</td>
<td>126</td>
<td>85</td>
<td>67.46</td>
</tr>
<tr>
<td>The sentences 101-120</td>
<td>&gt;&gt;</td>
<td>146</td>
<td>100</td>
<td>68.49</td>
</tr>
<tr>
<td>The sentences 121-140</td>
<td>&gt;&gt;</td>
<td>138</td>
<td>110</td>
<td>79.71</td>
</tr>
<tr>
<td>The sentences 141-160</td>
<td>&gt;&gt;</td>
<td>102</td>
<td>79</td>
<td>77.45</td>
</tr>
<tr>
<td>The sentences 161-180</td>
<td>&gt;&gt;</td>
<td>147</td>
<td>98</td>
<td>66.67</td>
</tr>
<tr>
<td>Sentences 181-200</td>
<td>&gt;&gt;</td>
<td>128</td>
<td>95</td>
<td>74.22</td>
</tr>
<tr>
<td>Average accuracy</td>
<td></td>
<td></td>
<td></td>
<td>75.15</td>
</tr>
</tbody>
</table>

Table 5.1 result of experiment using TnT

As it was indicated on Table 5.1, the performance of TnT i.e. the average of the sum of accuracies that were calculated by taking test set from different part of the corpus was 75.15% and result achieved was good for such small data.

5.2.1.2. Experiment 1 using CRF++

<table>
<thead>
<tr>
<th>Test set</th>
<th>Training</th>
<th>Total no. of</th>
<th>No. of correctly</th>
<th>Accuracy in %</th>
</tr>
</thead>
</table>

67 | Page
Table 5.2 experimental result of CRF++

As indicated in Table 5.2, the performance of CRF++ across the corpus is 64.04% and the result is low compared to TnT which was 75.15% but not the worst performance for such small data.

5.2.2 Experiment two

The second experiment was conducted to see the effect of the performance of the tagger by systematically selecting test set from the total of 200 sentences unlike that of experiment two which used contiguous of 20 sentences. The corpus was divided into two parts in the ratio of 90% (180 sentences) were used for training and the rest of 20 sentences (10% of the corpus) were used as test set. The test sets were selected from the total of 200 sentences systematically. For the purpose of this experiment, the 1st, 11th, 21th, 31th- - - 191th sentences were selected from the corpus to be used for the testing and the rest were used for training and the result of this experiment using the two tools was presented below.
5.2.2.1 Experiment 2 with TnT

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Over all accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>-d1/0.4</td>
<td>88.57</td>
<td>64.06</td>
<td>76.12</td>
</tr>
<tr>
<td>Trigram</td>
<td>-d2/0.4</td>
<td>88.57</td>
<td>73.44</td>
<td>81.34</td>
</tr>
<tr>
<td>Trigram</td>
<td>default</td>
<td>82.86</td>
<td>68.75</td>
<td>76.12</td>
</tr>
</tbody>
</table>

Table 5.3 Experimental result of TnT

As we can see from Table 5.3, the result of second experiment shows improvement on the performance when the test set is systematically selected from the whole corpus especially Trigram model with parameter -d2/0.4 (-d2/0.4 means add 0.4 to all frequency), has accuracy of 81.34%, 88.57% and 73.44% for overall, known and unknown. Trigram model with default option has the same overall accuracy as the bigram model with the option –d1/0.4 (which means replace all zero frequencies with 0.4) but has better performance for known and unknown accuracy than bigram model. As we can see from Table 5.3 the performance difference of (81.34% - 75.15%) 6.19% achieved due to systematic selection of test set from different parts of the corpus instead of taking contiguous sentences for HMM based taggers.

5.2.2.2 Experiment 2 with CRF++

<table>
<thead>
<tr>
<th>Training parameter option used</th>
<th>Total number of tokens in test set</th>
<th>Number of correctly tagged tokens</th>
<th>Accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using –a CRF-L1 algorithm option</td>
<td>134</td>
<td>89</td>
<td>66.42</td>
</tr>
<tr>
<td>Default option</td>
<td>134</td>
<td>84</td>
<td>64.17</td>
</tr>
</tbody>
</table>

Table 5.4 Experimental result of CRF++

As we can see from this experiment, CRF++ has improved performance when the test set was systematically selected from the whole corpus. The performance result of CRF based taggers
with default parameter is almost similar to the result achieved by 10 fold cross-validations and with parameter option or using parameter option during training like changing regularization algorithm (the default algorithm used by the tool is CRF-L2 is changed to –a CRF-L1) improved the performance from 64.17% which is achieved by the default algorithm to 66.42% as it was suggested by the developers of the tool for small data. As we can see from Table 5.4 the performance difference of (66.42% - 64.04%) 2.38% achieved due to systematic selection of test set from different parts of the corpus instead of taking contiguous sentences for CRF based taggers.

5.2.3 Experiment three

This experiment was conducted to see the effect of reducing the number of tag sets on the performance of taggers.

We have been using 22 tag sets for all of the above carried experiments and now by reducing the number of tag sets i.e. by leaving tags which are not significant in number in the corpus and also by merging tags like NA (Adverbial nouns), NV (verbal nouns, NC (nouns with conjunction) to the common noun and proper noun (NN). The performance of the taggers was checked for the reduced number of tag set. For the purpose of this experiment or to see the effect of reducing the number of tag sets on the performance of the taggers, I have experimented again only experiment three i.e. I used the data set similar to experiment three (out of 200 sentences, 180 sentences (90%) for training, 20 sentences (10%) for the testing and the test set were selected randomly and the 1st, 11th, 21th, 31th, - - - - , 191th sentences are selected).

Based on analysis done on Wolaita word class and the corpus used for this research, tags like VI (verb infixes) which appeared only three times in the corpus is reduced to VB (main verb) because it has similar properties with main verb. The other tag that was very rare in the corpus was interjection (UJ) which was occurred not more than two times was also reduced to (personal pronouns) PP based on its context in the corpus for the purpose of this experiment. And the other three tags reduced to common and proper nouns (NN) singular were verbal noun (NV), adverbial noun (NA) (which occurred about 3 times) and noun suffixed with conjunction (NC occurred 4 times).
5.2.3.1 Experiment 3 using TnT using 17 tags

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Parameter Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Over all accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>-d1/0.4</td>
<td>89.86</td>
<td>70.77</td>
<td>80.60</td>
</tr>
<tr>
<td>Trigram</td>
<td>-d1/0.4</td>
<td>91.30</td>
<td>70.77</td>
<td>81.34</td>
</tr>
<tr>
<td>Trigram</td>
<td>default</td>
<td>88.41</td>
<td>70.77</td>
<td>79.85</td>
</tr>
</tbody>
</table>

Table 5.5 Experimental result of TnT on reduced tag set

As it was indicated in Table 5.5, reducing the size of tag set from 22 to 17 increased the performance of the taggers from 76.12% to 80.60% for bigram model with option d1/0.4 (replace zero frequencies by constant (0.40), no change for trigram model with option d1/0.4 and from 76.12% to 79.85% for trigram mode with default option. The model developed on reduced tags has also better performance in terms of known and unknown accuracy.

5.2.3.2 Experiment 3 using CRF++ on reduced tag set 17 tags

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Parameter Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Over all accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF model1</td>
<td>Default option</td>
<td>79.71</td>
<td>56.92</td>
<td>68.66</td>
</tr>
<tr>
<td>CRF model2</td>
<td>Using CRF-L1 algorithm</td>
<td>81.16</td>
<td>58.46</td>
<td>70.15</td>
</tr>
</tbody>
</table>

Table 5.6 Experimental result of CRF on the reduced tag set

As it could be seen from the experimental result presented in Table 5.6, there is an increase in the performance of CRF due to decrease in the number of tag sets. The result showed that there is an increase in performance of the tagger i.e. from 64.18% to 68.66% and from 66.42% to 70.15% because of the decrement of tag set from 22 to 17.
Further reduction in the tag set like leaving the distinction between plural common and proper noun (NNS) and singular proper and common noun (NN) i.e. reducing tag set again from 17 to 16 yielded the following result.

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Parameter Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Over all accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM Trigram(TnT)</td>
<td>d1/0.4</td>
<td>91.30</td>
<td>75.38</td>
<td>83.58</td>
</tr>
<tr>
<td>CRF (CRF++)</td>
<td>Using –a CRF-L1</td>
<td>85.51</td>
<td>63.08</td>
<td>74.63</td>
</tr>
</tbody>
</table>

Table 5.7 experimental result of TnT and CRF on further reduced tag set.

This result shows that further reduction has brought better performance in accuracy of two taggers. For HMM based (TnT) taggers the performance of the tagger due to reduction from 17 tags to 16 was no change for known accuracy. There was increased from 70.77% to 75.38% for unknown and the difference was about 4.61%. The result also increased from 81.34% to 83.58% for over all accuracy and performance increment was 2.24%.

The performance change of CRF due to decrease in tag set form 17 to 16 was 81.16% to 85.51 (4.35%) for known accuracy, 58.46% to 63.08% (4.62%) for unknown and 70.15% to 74.63% (4.48%) for overall accuracy.

As it was indicated in the explanation of reduced tag set, the performance of both taggers (HMM (TnT) and CRF (CRF++)) has shown significant change in the performance for unknown, known (only for CRF), and overall accuracy.

5.2.4 Experiment four

This experiment was conducted by tagging more 180 sentences using one of the best performing tagger which was developed in the second experiment with 180 training sentences (90%) of the total 200 sentences and also tested with 20 test sets which were selected systematically as explained in experiment two. The 180 sentences tagged by tagger were added to increase the training set from 180 to 360 sentences (using semi-supervised learning). Then
models were developed using the 360 training data and tested using the same test set in experiment two and the results of this experiment were presented below using the two tools.

5.2.4.1 Experiment 4 with TnT

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Overall accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>-d2/1</td>
<td>83.53</td>
<td>63.27</td>
<td>76.12</td>
</tr>
<tr>
<td>Trigram</td>
<td>-d2/0.4</td>
<td>84.62</td>
<td>64.29</td>
<td>76.12</td>
</tr>
<tr>
<td>Trigram</td>
<td>Default</td>
<td>83.33</td>
<td>58.93</td>
<td>73.13</td>
</tr>
</tbody>
</table>

Table 5.8 Experimental result of TnT

As indicated in Table 5.7, the performance result of semi-supervised HMM based TnT tagger is low compared to the result of experiment two (supervised) but it is better than experiment one. The decrease in the performance of the tagger which was developed using training data of 360 sentences when compared to the one in experiment two was due to addition of erroneous data or since the new added 180 sentences were tagged by the model which was developed using small manually tagged 180 sentences and its accuracy was also only 81.34% as indicated in experiment two.

5.2.4.2 Experiment 4 with CRF++

<table>
<thead>
<tr>
<th>Training parameter option used</th>
<th>Total number of tokens in test set</th>
<th>Number of correctly tagged tokens</th>
<th>Accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using –a CRF-L1 algorithm option</td>
<td>134</td>
<td>89</td>
<td>66.42</td>
</tr>
<tr>
<td>Default option</td>
<td>134</td>
<td>87</td>
<td>64.93</td>
</tr>
</tbody>
</table>

Table 5.9 Experimental result of CRF++

As It was indicated in Table 5.9 the performance of semi-supervised CRF based taggers developed using 180 sentences manually tagged and 180 sentences tagged by the model
developed in experiment two is low but good when we compare it with experiment one. It also has the same performance as experiment two. The accuracy achieved by this tagger is good despite the addition of more erroneous data which was tagged by a model which is developed with only 180 sentences (small corpus) with accuracy of only 81.34%.

5.2.5 Experiment five

This experiment is conducted using a semi-supervised tagging. It used manually tagged data in the experiment four and additional untagged data of 180 sentences. This experiment was conducted to see the effect of addition of more 180 sentences by tagging them using the tagger which was developed in experiment four or the tagger developed with 360 sentences of training data and to have training data of about 540 sentences and then to develop taggers again with training data of 540 sentences and to see performance on the test set which is used in experiment two. The result of this experiment was presented below using the two tools.

5.2.5.1 Experiment 5 with TnT

<table>
<thead>
<tr>
<th>Model/tagger</th>
<th>Option used</th>
<th>Known accuracy in %</th>
<th>Unknown accuracy in %</th>
<th>Overall accuracy in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>-d2/1</td>
<td>83.53</td>
<td>59.18</td>
<td>74.63</td>
</tr>
<tr>
<td>Trigram</td>
<td>-d2/1</td>
<td>84.62</td>
<td>66.07</td>
<td>76.87</td>
</tr>
<tr>
<td>Trigram</td>
<td>default</td>
<td>83.53</td>
<td>57.14</td>
<td>73.88</td>
</tr>
</tbody>
</table>

Table 5.10 experimental result of TnT

As indicated in Table 5.10, the performance of the semi-supervised TnT tagger developed using 540 sentences as training set is better compared to experiment four in spite of addition of more erroneous data on training set. The performance difference achieved in experiment five is not significant, due to the addition of new more 180 sentences with addition of more erroneous data.

5.2.5.2 Experiment 5 using of CRF++
Using –a CRF-L1 algorithm option

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>134</td>
<td>93</td>
<td>69.40</td>
</tr>
<tr>
<td>Default option</td>
<td>134</td>
<td>88</td>
<td>65..67</td>
</tr>
</tbody>
</table>

Table 5.11 Experimental result of CRF++

As indicated in Table 5.11, the performance result of semi-supervised CRF based taggers improved when it was trained on 540 sentences than it was trained on 360 sentences. The performance difference of 2.98% (performance changed from 66.42% to 69.40%) using –a CRF-L1 algorithm option and 0.74% (performance changed from 64.93% to 65.67%) with default training algorithm option.

It is difficult to compare the performance of taggers developed in this research with taggers developed for other languages since they are trained on small data due to unavailability of large manually tagged corpus for Wolaita language. One of the researches conducted on Amharic language by Sisay (2005) using CRF based approach can somewhat be compared with this research. Sisay (2005) achieved accuracy of 70% without using any information source for taggers except training data and 74% accuracy using dictionary. The numbers of tag sets used by him and in this research are different even though the number of tokens in training set is about the same (number of tokens in training data was about 1000). Sisay (2005) used only 10 tags but in this research 22, 17 and 16 tags are used. In addition, Amharic is morphologically complex than Wolaita language. So it is difficult to compare this work with Sisay (2005).

5.3 Discussion

5.3.1 Error Analysis

Error analysis is done to see the confusion of the taggers among different tags and to see which of the word classes like noun, adjectives etc are similar to each other.

5.3.1.1 Error Analysis for TnT
Table 5.12 Error analysis of TnT tagger on test data set.

As it was indicated in the above table, common and proper nouns singular (NN) was mostly miss classified as personal pronouns (PP), ordinal numbers (OD), and cardinal number (CD). Relative verbs (VR) were miss classified as common and proper nouns singular (NN), main verbs (VB) and subordinate Verbs (VS). Unknown words contributed more to the error.

### 5.3.1.2 Error analysis on CRF++

<table>
<thead>
<tr>
<th>Miss classified tag</th>
<th>Miss classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NN</td>
</tr>
<tr>
<td>NN</td>
<td>-</td>
</tr>
<tr>
<td>NNS</td>
<td>0.026(1)</td>
</tr>
</tbody>
</table>
Table 5.13 Error analysis of CRF tagger on test data

As it was indicated on Table 5.12 CRF tagger miss classified most of the tags as common and proper noun singular (NN) and it was about 60% of the total error or miss classification and most of miss classified words are unknown words or words which were not in training corpus.

HMM (TnT) based taggers which are developed using semi-supervised approach in experiment four and five miss classified adjectives (JJ) as nouns (NN) most of the time, main verbs (VB) as verbal nouns (NV) and nouns (NN) as ordinal number (OD). This error analysis may indicate that the mentioned word class may be similar morphologically.

On the other hand CRF based taggers developed in experiment four and five miss classified postpositions (NP) as nouns common and singular (NN) and adjectives (JJ) as nouns common proper and singular (NN) most of the time. The confusion for adjectives and nouns is the same as that of HMM (TnT) and error analysis result here also shows that there is morphological similarity between nouns and adjectives. Finally, in thesis, five experiments have been conducted. The first three experiments conducted using the corpus of 200 sentences of which 90% (180 sentences) used as training set and 10% (20 sentences) used as the test set. The last two experiments conducted using semi-supervised approach which uses both tagged and untagged data and the results of both supervised and semi-supervised approaches have shown better performance for both HMM and CRF based taggers.
CHAPTER SIX
CONCLUSION AND RECOMMENDATION

6.1 Conclusions

In this thesis, the development of POS tagger using supervised and semi-supervised Hidden Markov model (HMM) and Conditional Random Fields (CRF) for Wolaita text have been explained and the experimental results have been reported.

This thesis presented brief discussion on Natural language Processing (NLP) and it’s role towards enabling computers to understand natural languages by which most of the human language is recorded. NLP, as it encompasses linguistics, computational linguistics and other relate fields of study, is important in design and development of part of speech (POS) taggers, parsers, morphological analyzers etc.

Application areas of part of speech tagging like parsing, information extraction, information retrieval, question answering, speech synthesis and recognition and machine translation were also briefly illustrated in this thesis. Different approaches used in part of speech tagging like rule based, corpus or stochastic and machine learning (supervised and unsupervised) methods were discussed. The advantages and disadvantages related to each approach were also presented in thesis. Some well-known machine learning algorithms like Hidden Markov Model (HMM), Maximum Entropy Model (MEM), Conditional Random Fields (CRF), N-gram model, Decision Tree etc. were presented in brief.

Different literatures that were reviewed in the area of part of speech tagging to understand the approaches used, the language structure, word classes to determine the tag sets were also described in this thesis. Related works done in the area of part of speech tagging for other languages abroad and local languages like Amharic, Afan Oromo and Kaffi-noonoo were also presented in the second chapter briefly.

Based on literature reviewed on Wolaita language, about 22 tag sets identified were described and presented in the third chapter. The Data or the corpus collected for this research, the
preparation and preprocessing, sampling methods used for the data selection, sampling methods used for training and test data selection, tools and techniques used in the thesis were also described also presented in the forth chapter.

Experiments were carried out in five phases using the two tools; the first three experiments were carried using 200 manually tagged sentences by dividing them into training and test set. In all of the first three experiments, of 200 sentences, 180 (90%) were used for training and 20 (10%) sentences were used for testing.

The first experiment was done on the total corpus of 200 sentences using 180 (90%) sentences as training set and 20 (10%) sentences as testing set. This experiment was carried to see the average performance of taggers across the corpus by taking the test set from different part of the corpus for 10 round, the first test set was sentences from 1-20, the second test set was sentences from line 21-40 until the 10th round which used the corpus at line 181-200 as test set and the rest as training and the average performance of both taggers was presented in experiment Table 5.1 and 5.2.

In the second experiment, the total of 200 sentences was combined and the test set is systematically selected. 180 (90%) of sentences were used for training and 20 (10%) of the sentences were used for testing. The test set was selected from the total of 200 sentences by systematically. The 1st, 11th, 21st, ----191th sentence are used as test set and the rest are used as training sets.

The third experiment was carried out on the reduced number of tag set i.e. the tag set reduced to 17 from 22 ( which used in all the above experiments except this one ) to see the effect of reducing the number of tag set on the performance of the taggers. The result achieved in this experiment was 81.34% for Hidden Markov models (HMM) based taggers and 70.15% for Conditional Random fields (CRF) based taggers using 17 tags. Further reduction in the number of tag sets to 16 showed accuracy of 83.58% for HMM and 74.63% for CRF based taggers. The fourth and the fifth experiments were conducted by using semi-supervised approach which uses both manually tagged and untagged corpus by adding more training data to the training set by using the model developed in experiment two with semi-supervised approach. The
performance of the taggers developed in semi-supervised manner was 76.87% for HMM (TnT) and 69.40% for CRF based taggers and this result shows possibility of developing POS taggers for Wolaita language with such small data compared to the models developed for other languages using very large size of training data.

The result of experiments conducted in this research showed that it is possible to develop POS taggers with acceptable performance using small manually tagged data using both supervised and semi-supervised approach. The result of experiment also showed that supervised and semi-supervised HMM based TnT taggers have better performance than CRF based CRF++ taggers for small data. Reduction of the number of tags from 22 to 17 and 16 increased the performance of both of the supervised taggers (HMM and CRF). The performance of supervised HMM based TnT taggers was better than that of semi-supervised HMM based TnT taggers but the performance of semi-supervised CRF based taggers was better than that of supervised CRF based taggers. The two semi-supervised experiments also showed that CRF based taggers show better performance when increasing training data than HMM based taggers.

It is difficult to compare the taggers developed in this thesis with taggers developed for other language due to the taggers developed in this thesis are developed using small manually tagged text data. The performance of the taggers developed in this thesis was very low related to that of English language and even far from the performance of HMM and CRF based taggers developed for Amharic Language due to small training data, no knowledge source was provided for taggers other than manually tagged corpus and the manually tagged corpus cannot be error free. Since the taggers developed in thesis were trained on small data, they may not have immediate practical application even though the accuracy achieved by taggers was fairly acceptable. The taggers developed in thesis were not trained on large amount of data because of lack of large POS tag annotated data, manually tagging large amount of data is expensive, laborious and time consuming, and scope of the thesis. Therefore, this thesis indicated the possibility of developing part of speech tagger for Wolaita text in spite of using small data for training the taggers. This research also showed that it is possible to develop an efficient part of speech tagger for Wolaita language if it is possible to get large training data. This thesis also will...
encourage students and researchers who are interested to carry research in the language especially in higher level NLP application tasks like parsing, machine translation, and word sense disambiguation. Since it is the first POS tagging research in this language, understanding the language structure and determining the tag set, finding and installing tools suitable for POS tagging process were the challenges faced by the researcher. There has to be cooperation and coordination between department of Information Science and linguistics for students carrying their research in the areas of NLP tasks like POS tagging, Parsing, etc.

6.2 Recommendations

Pos taggers developed in this thesis are the first attempt for Wolaita text and further research has to be done to improve the performance of the taggers to operational level. In addition, this research has limitations and gaps which can open the door for future researchers to develop POS tagger for Wolaita Language that has better performance. Therefore, the following are some of future research directions.

- One can replicate this work using large corpus for training the taggers for Wolaita language.
- Due to time constraint, only two experiments have been conducted using the semi-supervised approach in this research. So in the future one can conduct more experiments to improve the performance of semi-supervised taggers.
- In this research reduction of tags was based on number of occurrences of tags in the corpus or tags that occur rarely are reduced to the nearest tag but in the future one can use error analysis to reduce tags and can see the effect on the performance of the taggers.
- Since most of Ethiopian languages are under-resourced and do not have large size POS annotated corpus, they can benefit from semi-supervised approach. So one can develop POS Taggers for other local languages following approaches used in this thesis especially semi-supervised approach with small manually tagged corpus.
References


9. Berna Arslan and Özlem Patan (2009), Developing Methods for Part of speech tagging in
Turkish, Bachelor’s thesis, Boğaziçi University.


25. Karen Sparck Jones (no year), Natural Language Processing: A Historical Review, Computer Laboratory, University of Cambridge e-mail: kn11@phx.cam.cl.


27. Lamberti, Marcello and Roberto Sottile (1997), The Wolaytta Language, In Studia


University.


39. Samira Tasharofia, Fahimeh Rajaa, Farhad Oroumchiana,b, and Masoud Rahgozar
   Evaluation of statistical Part of speech tagging of Persian Text, a Electrical and Computer Engineering Department, University of Tehran
   {s.tasharofi,f.raja}@ece.ut.ac.ir Oroumchian@acm.org, and Rahgozar@ut.ac.ir
   University of Wollongong in Dubai FarhadO@uow.edu.au

40. Sandipan Dandapat (2009), Part-of-Speech Tagging and Chunking with Maximum Entropy Model, Department of Computer Science and Engineering, Indian Institute of Technolog, Kharagpur India 721302, sandipan@cse.iitkgp.ernet.in.


43. Sisay Fissaha (2005), Part of speech tagging for Amharic using conditional random fields, In ACL, pp. 47–54, Workshop on Computational Approaches to Semitic Languages.


46. Taku A. (2005), CRF++: yet another Toolkit, available online:
http://taku910.github.io/crfpp/


48. Thorsten Brants (200), TnT- A Statistical Part of Speech Tagger, Saarland University, Computational Linguistics, thorsten@coli.uni-sb.de.

49. Thorsten Brants (no year), TnT- A Statistical Part of Speech Tagger, Saarland University, Computational Linguistics, thorsten@coli.uni-sb.de.


APPENDIX I: Sample Untagged Wolaita text

APPENDIX II: Sample manually tagged Wolaita text

APPENDIX III: Sample manually tagged Wolaita Text token per line for training the tools

Qofatanne NC
naqaashata NNS
qorada VS
shiishshaasa VB
Neeni PP
shiishshido VR
qofata NNS
ziireyaasa VB
Neeni PP
qofettida VR
ba NN
ubbaa NN
gede NN
ziiriyan AD
wottaasa VB
Neeni PP
Xufettaa NN
giyogee NV
misiliya NN
woykko CC
karttaa NN
meruwakka NNS
geetettees VB
Xufettay NN
xaafiyo VR
ba NN
demmanawu VT
maaddiya VR
ossuwa NN
. PU
bari PP
qofaa NN
kuuyyanawu VT
dosiya VR
asaassi NN
hagee NN
keehi AD
injje JJ
APPENDIX IV: Sample output tagged by TnT tool


%% lexicon : train.lex
%% ngrams : train.123
%% corpus : utest.txt
%% model : trigrams

%% sparse data : linear interpolation

%% lambda1 = 1.266094e-01   lambda2 = 3.787554e-01   lambda3 = 4.946352e-01

%% unknown mode: statistics of singletons

%% case of characters is significant

%% using suffix trie up to length 10

%% suffix backoff with theta = 1.577681e-01

%% Thorsten Brants, thorsten@coli.uni-sb.de

Mazggapiyoge PP
guugggiyappe NP
soridi VS
qoliyooga VB
Suuppaa JJ
qofata NNS
yafarada VS
bessa
Silimaa
be7iya
assay
wocamees
.
Tiikeetiya
gatee
al7o
.
Wuyigen
yedhdhiyo
wode
xanddaqi
giyabay
dares
Ha
xekkan
neeppe
APPENDIX V: CRF++ Template file used for tagging

# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-2,0]/%x[-1,0]/%x[0,0]
U06:%x[-1,0]/%x[0,0]/%x[1,0]
U07:%x[0,0]/%x[1,0]/%x[2,0]
U08:%x[-1,0]/%x[0,0]
U09:%x[0,0]/%x[1,0]

# Bigram
B