ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUTE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

PART-OF-SPEECH TAGGING FOR AFAAN OROMO LANGUAGE

BY
GETACHEW MAMO

A THESIS SUBMITED TO THE SCHOOL OF GRADUTE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCINECE IN INFORMATION SCIENCE

ADDIS ABABA, ETHIOPIA
JANUARY, 2009

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUTE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SCIENCE

PART-OF-SPEECH TAGGING FOR AFAAN OROMO LANGUAGE

BY

GETACHEW MAMO

Signature of the Board of Examiners for Approval

	Name	Signature
1.	_____	_____
2.	_____	_____
3.	_____	_____
4.	_____	_____

Acknowledgment

First, I would like to thanks Almighty God for helping me to finalize my work peacefully.

My deepest gratitude goes to my advisor Dr. Million Meshesha for his critical comments on my work.

I would also thanks Ato Mesfin Getachew who has a great input in my study in providing relevant resources and leading me to the right track.

I am also grateful to my brothers, Dr. Mengesh Mamo and Ato Fituma Tefera, who have supported me in finance and materials during my stay in the postgraduate school.

I would also give my gratitude to my friends: Ato Behailu Getachew, Ato Dejen Alemu, Ato Eskindir Mesfin, Ato Teshome Alemu, Ato Zelalem Ragassa and Ato Haleluya Kifilu who have directly and indirectly contribute in my study with moral and material support.

Table of Contents

Abbreviations

POS	Part-of-speech
NLP	Natural Language Processing
LexProb	Lexical Probabilities
TransProb	Transitional Probabilities

List of Tables and Figures

Tables

Figures

Abstract

Most natural language processing systems use part-of-speech (POS) tagger as a separate module in their architecture. Specially, it is very significant for developing parser, machine translator, speech recognizer and search engines. Tagging is a process of labeling part-of-speech tags to words of a text such that contextual information can be obtained from word labels.

The main aim of this study is to develop part-of-speech tagger for Afaan Oromo language. After reviewing literatures on Afaan Oromo grammars and identify tagset and word categories, the study adopts Hidden Markov Model (HMM) approach and has implemented unigram and bigram models of Viterbi algorithm. HMM is a statistical approach which is used in this study for part-of–speech tagging for Afaan Oromo words in a given corpus. Unigram model is used to understand word ambiguity in the language, while bigram model is used to undertake contextual analysis of words.

For training and testing purpose 159 sentences (with a total of 1621 words) that are manually annotated sample corpus are used. The corpus is collected from different public Afaan Oromo newspapers and bulletins to make the sample corpus balanced. A database of lexical probabilities (LexProb) and transitional probabilities (TransProb) are developed from this annotated corpus. These two probabilities are from which the tagger learn and tag sequence of words in a sentence.

Java programming language is used to develop the tagger prototype based on the Viterbi algorithm with unigram and bigram models. It is also used to compute both lexical probabilities and transitional probabilities.

The performance of the prototype, Afaan Oromo tagger is tested using ten fold cross validation mechanism. The result shows that in both unigram and bigram models 87.58% and 91.97% accuracy is obtained, respectively. Based on experimental analysis, concluding remarks and recommendations are forwarded.

Chapter One

1. Introduction

1.1 Background

Natural language is any language that human beings use to communicate with each other. As stated in [1], natural language is not only the most important means of communication between human beings; it is also used over historical periods for the preservation of cultural achievements and their transmission from one generation to other.

Nowadays, natural language processing has been provoked by many scholars to understand its structural form through computational processes. Since most of human knowledge is recorded in linguistic form, computers that could understand natural language could access all these information [2].

The flood of digitized information has also been growing tremendously in this generation. This tendency will continue with the globalization of information sharing and with the growing importance of computer network at national and international level. This is one reason why the theoretical understanding and the automated treatments of communication processes based on natural language have such a decisive social and economic impact [1].

To make natural languages based communication processes and sharing of cultural values in digitized world, understanding of structural description of languages are one of the

fundamental issue of natural language processing. As indicated in [2], natural language system must use considerable knowledge about the structure of the language itself, including what the words are, how words combine to form sentence, what the words mean, how word meanings contribute to sentence meanings, and so on.

Natural language processing (NLP) is a field of study that deals with computing natural language with machine and provides a potential means of gaining access to the information inherent in the large amount of text available through intranet and the Internet.

According to [3], NLP is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things such as: multi- or cross-lingual information retrieval, including multilingual text processing and multilingual user interface systems, in order to exploit the full benefit of the World Wide Web (WWW) and digital libraries.

At the heart of any NLP task, there is the issue of natural language understanding. However, the process of building computer programs that understand natural language is not straightforward. As explained in [1], natural languages give rise to lexical ambiguity that words may have different meanings, i.e. one word is in general connected with different readings in the lexicon. Homograph, the phenomenon that certain words showing different morpho-syntatic behavior are identically written. For instance, the word ‘Bank’ has different meanings; Bank (= financial institute), Bank (= seating accommodation), etc.

In other words, words match more than one lexical category depending on the context that

they appear in sentences. For example, if we consider the word *miilaa 'leg'* in the following two sentences,

Lataan kubbaa *miilaa* xabata. 'Lata plays a football'.

Lataan *miilaa* eeraa qaba. 'Lata has long leg'.

In the first sentence, *miilaa 'leg'* takes the position of adjective to describe the noun kubbaa 'ball'. But in the second sentence, *miilaa* is a noun described by eeraa 'long'.

Besides ambiguity of words, inflection and derivation of the language are other reasons that make natural language understanding very complex. For instance, *tapha 'play'* contains the following inflection in Afaan Oromo language.

tapha-t	' she plays'
tapha-ta	'he plays'
tapha-tu	'they play'
tapha-ta-niiru	' they played'
tapha-chuu-fi	'they will play'

In the above particular context suffixes are added to show gender {–t, --ta}, number { –tu/--u} and future {--fi}.

To handle such complexities and use computers to understand and manipulate natural language text and speech, there are various research attempts under investigation. Some of these include natural language generation, machine translation, information extraction and

retrieval using natural language, text to speech synthesis, automatic written text recognition, grammar checking, and part-of-speech tagging. Most of these approaches have been developed for popular languages like English [3]. However, there are few studies for Afaan Oromo language. So, this study attempts to investigate the possibility of designing and developing an automatic part-of-speech tagger for Afaan Oromo natural language processing.

Part-of-speech (POS) tagging is the act of assigning each word in sentences a tag that describes how that word is used in the sentences. That means POS tagging assigns whether a given word is used as a noun, adjective, verb, etc. As Pla and Molina [4] notes, one of the most well-known disambiguation problems is POS tagging. A POS tagger attempts to assign the corresponding POS tag to each word in sentences, taking into account the context in which this word appears.

POS tagger finds the possible tags or lexical category for each word provided that the word is in a lexicon and guess possible tags for unknown words. It also chooses possible tags for each word that is ambiguous in its part-of-speech (syntactic disambiguation). Ambiguous words are very common in most languages. If a certain word is assigned more than one tag, this means that the word can have different meanings or functions in different contexts, for example, the Afaan Oromo word *diimaa* meaning *'red'* can be either a noun, adjective, or a verb.

Each POS tag is composed of the lexical category of the word (common noun, proper noun, adjective, etc.) and usually adds morphological information (number, gender, person, etc.). Normally, this set of POS tags must be previously defined by human expert for a language

(like Afaan Oromo) [23].

1.2 Justification and Statement of the Problem

Oromo (also known as Afaan Oromo) is one of the major languages that are widely spoken and used in Ethiopia [6]. Currently it is an official language of Oromia state (which is the largest region in Ethiopia). It is a common mother tongue for Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 34.5% of the total population according to the 2008 census [41].

With regard to the writing system, since 1991 Qubee (Latin-based alphabet) has been adopted and become the official script of Afaan Oromo [19]. Currently, Afaan Oromo is widely used as both written and spoken language in Ethiopia. Besides being an official working language of Oromia State, Afaan Oromo is the instructional medium for primary and junior secondary schools throughout the region and its administrative zones. Thus, the language has well established and standardized writing and spoken system. As a result the use of computers for processing information and storing huge amount of repositories with this language in governmental and non governmental offices and institutions in the region has been growing extremely. So, it is sound to process the linguistic form of the language with computers to understand and used the language for communication and transformation of knowledge in this digitized world.

To use computers for understanding and manipulation of Afaan Oromo language, there are very few researches attempted. These attempts include text-to-speech system for Afaan Oromo

[8], an automatic sentence parser for Oromo Language [9] and developing morphological analyzer for Afaan Oromo text [10].

There are also other related researches that were conducted on other local language. Especially on Amharic language, two researches were conducted on POS tagging by [5] and [11], but to the best of our knowledge there is no POS tagging research conducted for Afaan Oromo language.

Therefore, it is the aim of this study to develop POS tagger for Afaan Oromo so as to establish the base for future researchers who have interested in the area of machine translation, information retrieval and text summarization.

The output of POS tagger has many applications in many natural language processing activities [4]. Morpho-syntactic disambiguation is used as preprocessor in NLP systems. Thus, the use of a POS tagger simplifies the task of syntactic or semantic parsers because they do not have to manage ambiguous morphological sentences. Thus parsing cannot proceed in the absence of lexical analysis, and so it is necessary to first identify and determine part-of-speech of words.

It can also be incorporated in NLP systems that have to deal with unrestricted text, such as information extraction, information retrieval, and machine translation. In this modern world, huge amount of information are available on the Internet in different languages of the world. To access such information we need machine translator to translate into local languages. To develop a machine translation system, the lexical categories of the source and target languages

should be analyzed first since a translator translates, for example, nouns of the source language to the nouns of the target language. So, POS tagger is one of the key inputs in machine translation processes.

A word's part-of-speech can further tell us about how the word is pronounced. For instance, the word '*content'* in English can be a noun or an adjective. It is pronounced as '*CONtent'* and '*conTENT'* respectively. Thus, knowing part-of-speech can produce more natural pronunciations in a speech synthesis system and more accuracy in a speech recognition system [8].

All these applications can benefit from POS tagger to improve their performance in both accuracy and computational efficiency. So, it is worth to conduct a research and develop part-of-speech tagger for Afaan Oromo to simplify natural language processing and issues related to computational linguistics for Afaan Oromo language.

1.3 Objective of the Study

Details of the general and specific objectives of the research work are the following: -

1.3.1 General Objective

The general objective of the study is to investigate the possibility of designing and developing an automatic part-of-speech tagger for Afaan Oromo language.

1.3.2 Specific Objectives

In order to achieve the general objective, the research has the following specific objectives

- To review literatures and analyze documents concerning basic Afaan Oromo language word categories, its morphological properties and its sentences structure to derive the POS tags and the tagset for the language. And most common techniques suggested for the development of a POS tagger.
- Select and design the algorithm required for the development of a POS tagger.
- To study the type of lexicon required for Afaan Oromo POS tagging and design the appropriate lexicon.
- To develop a prototype, Afaan Oromo POS tagger.
- To test the prototype tagger and analyze its performance.
- To draw conclusions based on experimental results and recommend further research area in the future.

1.4 Research Methodology

The methods that are used for this study are literature review, algorithm design and development and testing for performance evaluation.

1.4.1 Review of Literatures and Discussion with Linguists

This method is used to understand the morphological property of Afaan Oromo language and its word classes. Literatures in the area of POS tagging are also reviewed, specifically that are based on Hidden Markov Model (HMM) approach in relation to Viterbi algorithms.

Discussion with experts and linguists are conducted to get more understanding in the area of word classes and morphological properties of Afaan Oromo language.

1.4.2 Algorithm Design and Implementation

HMM approach is adopted for the study because it performs the tagging process based on statistical information of words in a corpus. In other word, it does not need detail linguistic knowledge of the language as rule based approach [23].

As [40] described, the Viterbi algorithm is a dynamic programming algorithm that optimizes the tagging of a sequence, making the tagging much more efficient in both time and memory consumption. In a naïve implementation it would be calculated the probability of every possible path through the sequence of possible word-tag pairs, and then select the one with the highest probability. Since the number of possible paths through a sequence with a lot of ambiguities can be quite large, this will consume a lot more memory and time than necessary.

Since the path with highest probability will be a path that only includes optimal sub paths, there is no need to keep sub paths that are not optimal. Thus, the Viterbi algorithm only keeps the optimal sub path of each node at each position in the sequence, discarding the others. So, Viterbi algorithm is implemented for the study.

A sample lexicon, tagset, statistical databases and a prototype tagger are designed and developed to tag sentences in appropriate manner.

To implement this algorithm Java programming language is used. Java is relatively rich with

natural language processing inbuilt modules than other programming languages like C++ and visual basics. It is also freely available on the Internet.

1.4.3 Test and Evaluation

The prototype tagger is tested based on the sample test data prepared for this purpose. The performance evaluation is analyzed based on correctly tagged once by the prototype tagger. It is performed in two ways.

The first analysis is to evaluate the validation of the tagger with portion of the training set. Out of total data set 20% is used for testing and 100% (including test set) is used for training. Since the test set is part of the training set, it is used also to validate the manually tagged sample corpus. The test is done repeatedly through correcting manually tagged sample corpus until the satisfactory result is obtained.

The second performance analysis is using ten fold cross validation. Ten fold cross validation divides a given corpus in to ten folds. And nine folds are used for training and the tenth fold is used for testing. It provides an unbiased estimate of value of prediction error and preferred for small sample corpus [42].

1.5 Scope and Limitation of the Study

The scope of the research is limited to investigating the possibility of Hidden Markov Model approach to design automatic POS tagger for Afaan Oromo. The study is also limited to implement with both unigram and bigram models of Viterbi algorithm. Regarding the lexical

categories, the study focuses on broadly categorized part-of-speech of the language.

The main limitation while processing the study is the absence of readily available annotated corpus and word categories for natural language processing for the language. This would set a constraint on amount of sentences and broadly considering word categories that would be used for the study. Annotating corpus and specifying of word categories in detail are very tough and manually annotated by human experts and linguists, so that it needs much time and effort.

1.6 Application of the Study

This study becomes the groundwork for further researches in the areas of natural language processing in general and POS tagging in particular for Afaan Oromo language. Specially, it could be used as component part for application such as machine translation, spell checker, text summarization, information extraction and information retrieval.

Linguists and students in the area of Afaan Oromo can also apply the output of the study to identify word classes (part-of-speech) in a sentence automatically for language education.

1.7 Organization of the Thesis

The thesis is organized in five chapters. The first chapter discusses about statement of the problem, objectives, methodology, scope and limitation of the study. In chapter two, different issues regarding part-of-speech tagging are discussed. HMM and rule based approaches are reviewed. With special attention, HMM with respect to Viterbi algorithms are reviewed in

detail.

The third chapter is focused on word categories for Afaan Oromo. In the chapter, word categories and tagset that are implemented in chapter four are discussed and identified. The forth chapter deal with about implementation and performance analysis of the tagger. Sample corpus, lexical and transitional probabilities are processed and developed. Both bigram and unigram Viterbi algorithms are designed and implemented.

The last chapter, chapter five, presents conclusion based on the experiment that are done in chapter four and recommends potential NLP areas of researches for further study.

Chapter Two

2. Part-of-speech Tagging

2.1 Introduction

In natural language processing, part-of-speech (POS) tagging is the automatic annotation of words with grammatical categories, also sometimes called lexical categories. In other words, it corresponds to the identification of the words' part-of-speech in a sentence or corpus.

Part-of-speech tagging, or simply tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part-of-speech. It is a technique for deciding whether each word is a noun, verb, adjective, adverb, etc [18]. POS taggers have been applied to assign a single best POS to every word in a corpus. For example, the following is tagged sentence in Afaan Oromo Language.

Leenseen\NN kaleessa\AD deemte\VV 'Lense went yesterday'.

In the above example, words in the sentence, *Leensaan kaleessa deemte***,** are tagged with appropriate lexical categories of noun, adverb and verb respectively. The codes NN, AD, VV are tags for noun, adverb and verb respectively. The process of tagging takes a sentence as input, assigns a POS tag to the word or to each word in a sentence or in a corpus, and produces the tagged text as output.

A corpus, plural corpora, is a special collection of textual material collected according to a certain set of criteria. For example, the Brown corpus was designed as a representative sample

of written American English. Some of the criteria employed in its construction were to include particular texts in amounts proportional to actual publication [18]. It is a collection of naturally occurring texts chosen to characterize a state or variety of the language.

A corpus can be categorized as unannotated and annotated corpus. Unannotated corpus is a collection of raw texts that does not contain additional linguistic information or labels on words. For instance, the following sentence is unannotated or raw texts.

Leensaan baratuudha 'Lensa is a student'. In this sentence there is no additional linguistic information or other feature that describe words explicitly (*Leensaan, baratuudha*) in the sentence.

Annotated corpus is a collection of texts that are labeled by linguistic lexical categories or tags. This type of corpus explicitly describes about words based on contextual position in a sentence. *Leensaan/NN baratuudha/VBZ.* Words in the sentence are annotated. The word *'baratuudha'* is tagged with VBZ, a code that indicate third person singular. So that, we can simply identify what the words signify in a sentence based on its tag.

Annotated corpus has many applications in natural language processing. It enables to retrieve the frequency lists and indexes of various words or other structures of language with in it. Since corpus is available with explicitly linguistics information, data retrieval from the corpus can be easier and more than with unannotated data. It provides the most reliable source of data on language as it is actually used because it is naturalistic collection of data. It is an essential tool to develop part-of-speech taggers and parsers [22].

The annotation of corpus is done either manually or automatically. Manually annotating a corpus is more expensive, time talking and needs fair amount of handworks. It is assumed that manually annotated corpus is faultless. However, deep investigation shows that the annotated corpus is prone to error [22]. This is because; among annotators there may be disagreements while annotating a corpus. And even one annotator may not annotate the same sentence at different time in the same way. This creates inconsistence during annotation. Automatic annotation solves all limitations mentioned above in tagging corpus manually. Automatic annotator labels a large corpus within short period of time and in a consistence process.

There are many automatic annotators or taggers available for different languages [5, 24, 25, 39, 40]. However, this is not true for Afaan Oromo language. As it is mentioned in the previous chapter, tagging system are used as a preprocessor for many natural language processing tasks such as parsing, machine translation and information extraction. So, the study intends to fill the gap through investigating automatic part-of-speech for Afaan Oromo language. This chapter reviews HMM and rule based approaches used for part-of-speech tagging with more emphasis on HMM approach.

2.2 Approaches to Automatic Part-of-speech Tagging

A word's part-of-speech can not be identified simply by looking at the word from a given corpus. For instance, the word *'can'* in English has different part-of-speech in different context; it is a verb in the phrase *'he can'*, and a noun in the phrase *'a can'*. So a simple corpus lookup or a morphological analysis produces many words that are ambiguous, especially nouns and verbs.

There are two efficient approaches that have been established to solve such a problem [23]. The first approach uses rule constraint. Rules consider the left and right context of the word to disambiguate, that is, either discard or replace a wrong part-of-speech. Rules are symbolic and can be designed by hand or derived automatically from hand-annotated corpora. The second method is based on statistics (stochastic). Sequence statistics are automatically learned from hand-annotated corpora, and probabilistic models are applied that assign the most likely tags to words of a sentence.

2.2.1 Rule Based Approach

Rule based taggers use hand coded rules to determine the lexical categories of a word [2, 22]. Words are tagged based on the contextual information around a word that is going to be tagged. Part-of-speech distributions and statistics for each word can be derived from annotated corpora - dictionaries. Dictionaries provide a list of word with their lexical meanings. In dictionaries there are many citations of examples that describe a word in different context. These contextual citations provide information that is used as a clue to develop a rule and determine lexical categories of the word.

The earlier rule based algorithms for automatically assigning part-of-speech are based on two-stage architecture. The first stage uses a dictionary to assign each word a list of potential part-of-speech. For example, for the following English words we may get their potential part-of-speech as follows:

she: PP

promised: VBN,VBD

to: TO

back: VB, JJ, RB, NN

the: DT

bill: NN, VB

The code PP, VBN, VBD, TO, VB, JJ, RB, NN, DT are Personal pronoun, Verb-past participle, Verb-past tense, to, Verb-base form, Adjective, Adverb, Noun sing. or mass, Determiner respectively in Penn Treebank corpus [22].

The second stage used large lists of hand-written disambiguation rules to reduce down this list to a single part-of-speech for each word. For instance, Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP" [23].Where, PRP stands for personal pronoun.

The resent rule based algorithm is transformation based tagging (TBL); it is commonly known as Brill's rule based tagger [22]. Unlike earlier algorithm, it uses supervised learning technique that uses a pre-manually tagged training corpus. Brill's TBL algorithm has also stages to assign a word with appropriate tag. In the first stage, each word is assigned with its appropriate lexical category from annotated corpora. As a preprocessor, it assigns the most likely (frequent) tag to each word; for example, I/pro can/modal see/verb a/art bird/noun.

Brill's tagger then applies a list of transformations to alter the initial tagging. Transformations are contextual rules that rewrite a word tag into a new one. The transformation is performed only if the new tag of the word is legal, in the dictionary. If so, the word is assigned the new

tag. Transformations are executed sequentially and each transformation is applied to the text from left to right [2].

In English, change the tag from modal to noun if the previous word is an article. And the rule is applied to a sentence, *the/art can/noun rusted/verb*. Brill's rules tagger conforms to a limited number of transformation types, called templates. For example, the rule changes the tag from modal to noun if the previous word is an article, corresponds to template. The following table shows sample template that is used in Brill's rule tagger.

Rules	Explanation
alter(A, B, prevtag(C))	Change A to B if preceding tag is C
alter(A, B, nexttag(C))	Change A to B if the following tag is C

Where, A, B and C represent lexical categories or part-of-speech.

Table 2.1: Sample Template in Brill's Rule

For English language, there is no standard technique to deal with the unknown words in Brill's rule. But the baseline is to tag unknown words as nouns since it is the most frequent part-of-speech. Another technique is to use suffixes. The initial step tags unknown words as proper nouns for capitalized words and as common nouns for the rest [22].

2.2.2 Stochastic Approach

Most current part-of-speech taggers are probabilistic (stochastic). It is preferred to tag for a word by calculating the most likely tag in the context of the word and its immediate neighbors [24, 25].

The intuition behind all stochastic taggers is a simple generalization of the 'pick the most-likely tag for this word' approach based on the Bayesian framework. A stochastic approach includes most frequent tag, n – gram and hidden Markov model [22].

2.2.2.1 Most Frequent Tag

This is the simplest stochastic approach that finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag words in a sentence. For example, *P(verb|race)* can be estimated by looking at some corpus and saying "*out of all the times we saw 'race', how many are verbs?*", P(noun/race) "*out of all the times we saw 'race', how many are nouns?"* and so on for all lexical categories. And assign the category for a *'race'* that contains the highest probability. As [2] stated, this technique often obtains about a 90 percent success rate in English language, primarily because over half the words appearing in most corpora are not ambiguous.

However, there is a limitation with this approach when words come in sequence to form a sentence; it breaks the grammatical rule since it considers only the highest frequency of a category for a given word. For instance, all words in a sentence may be a noun or a verb. This is due to lack of the local context of a word in a sentence in which the word appears. N-gram is regarded as alternative solution for this problem. For instance, in English, if the word is preceded by the word *'the'*, it is much more likely to be a noun.

2.2.2.2 N-Gram

N-Gram is the technique to develop the local context of the sentence in which the word appears. It is one option to the word frequency approach. It calculates the probability of a given sequence of tags. It decides the appropriate tag for a word by calculating the probability that it occurs with in the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes [23]. These are known as the unigram, bigram and trigram models. The most common algorithm for implementing an n-gram approach for tagging new text is known as the Viterbi Algorithm [22]. Viterbi algorithm is discussed under chapter four, section 4.5.3.

2.2.2.3 Hidden Markov Model (HMM)

HMM is the statistical model which has been most used in POS tagging. The general idea is that, if we have a sequence of words, each with one or more potential tags, then we can choose the most likely sequence of tags by calculating the probability of all possible sequences of tags, and then choosing the sequence with the highest probability [26]. We can directly observe the sequence of words, but we can only estimate the sequence of tags, which is 'hidden' from the observer of the text; hence the term 'Hidden Markov Model' was given. A HMM enables us to estimate the most likely sequence of tags, making use of observed frequencies of words and tags (in a training corpus) [23].

The probability of a tag sequence is generally a function of:

- the probability that one tag follows another (n-gram); for example, after a determiner tag an adjective tag or a noun tag is quite likely, but a verb tag is less likely. So in a

sentence beginning with *the run…*, the word *'run'* is more likely to be a noun than a verb base form.

- The probability of a word being assigned a particular tag from the list of all possible tags (most frequent tag); for example, the word *'over'* could be a common noun in certain restricted contexts, but generally a preposition tag would be overwhelmingly the more likely one.

So, for a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes the following formula [23]:

P(word/tag) * P(tag/previous n tags)

Most frequent tag (likelihood) — N-gram (a prior)

As a result, HMM is the combination (product) of both most frequent tag and n-gram tag.

The use of HMM for tagging problem is nothing but finding the sequence of tags for a given sequence of words that maximizes the probabilities of the above formula, i.e., restated as $P(t_1,t_2 \ldots t_n / w_1, w_2 \ldots w_n)$.

$$bt_1^n = P(t_1, t_2, ..., t_n / w_1, w_2, ..., w_n) \quad \text{.....1}$$

Where, bt, P, t, and w represents the best sequence tags, probability, appropriate tag, word, respectively.

This equation is guaranteed to give us the best tag sequence. However, it is difficult to apply

directly in to operational since it needs too much data to estimate the result. So, it should be restated the equation using Baye's rule to transform into a set of other probabilities that are easier to compute [23]. Baye's rule stated as,

P(x/y) = (P(y/x) * P(x))/ P(y)

So, the Formula 1 is converted to Baye's rule as follows,

$$bt_1^n = (P(t_1,t_2,...,t_n) * P(w_1,w_2,...,w_n/t_1,t_2,...,t_n))/P(w_1,w_2,...,w_n) \quad \text{.....2}$$

Since we are looking for the most likely tag sequence for a sentence given a particular word sequence, the probability of the word sequence $P(w_1, w_2...,w_n)$ will be the same for each tag sequence so we can ignore it.

$$bt_1^n = P(t_1,t_2,...t_n) * P(w_1,w_2,...w_n/t_1,t_2,...t_n) \quad \text{.....3}$$

Statistics on sequences of any length are impossible to obtain, and at this point we need to make some approximations on Formula 3 to make the estimation tractable. As Allen [2] described, there are still no effective methods for calculating the probability of these long sequences accurately, as it would require far too much data. So, the probabilities can be approximated by probabilities that are simpler to calculate by making some assumptions.

Each of the two expressions in the Formula will be approximated (assumed) as follows. The first expression ($P(t_1,t_2...,t_n)$), the probability of the sequence of tags, can be approximated by a series of probabilities based on a limited number of previous tags. The most common

assumptions use either one, two or three previous tags. The unigram only looks at a single tag for a word. The bigram model looks at pairs of tags (or words) and uses the conditional probability that a tag t_1 will follow a category t_i-$_1$, written as $P(t_1 \mid t_i\text{-}_1)$. The trigram model uses the conditional probability of one tag (or word) given the two preceding tags (or words), that is, $P(t_i \mid t_i\text{-}_2\ t_i\text{-}_1)$.

The first assumption is Markov assumption that simplifies first factor in Formal 3: a tag only depends on a fixed number of previous tags (in the case of bigram it depends on a single previous tag).

$$P(t_1,t_2...,t_n) \approx \prod_{i=1}^{n} P(t_i / t_{i-1}) \quad \text{.....4}$$

The second assumption is independent assumption that simplifies the second factor (P $(w_1,w_2...,w_n / t_1,t_2...,t_n)$) from Formula 3: words are independent from each other, that means, a word's identity only depends on its own tag.

$$P(w_1,w_2...,w_n / t_1,t_2...,t_n) \approx$$

$$\prod_{i=1}^{n} P(w_i / t_i) \quad \text{.....5}$$

So, the final formula that is applied in Viterbi algorithm can be the multiplication of Formula 5 and Formula 4:

That means , the Formula 3 (bt_1^n = $P(t_1,t_2...,t_n)$ * P $(w_1,w_2...,w_n / t_1,t_2...,t_n)$) is simplified as :

$P(t_1,t_2...,t_n)$ is assumed as $\prod_{i=1}^{n} P(t_i / t_{i-1})$ ……..…..Markov assumption.

$P(w_1,w_2...,w_n / t_1,t_2...,t_n)$ is assumed as $\prod_{i=1}^{n} P(w_i / t_i)$ …..Independent assumption.

So that, the following formula is the applicable formula, that is used for bigram model.

$$bt_1^n = \prod_{i=1}^{n} (P(w_i / t_i) P(t_i / t_{i-1})) \quad \text{……..……6}$$

As it is discussed above, there are two kinds of probabilities in Formula 6. Tag transition probabilities $p(t_i|t_{i-1})$ and word likelihood probabilities $p(w_i|t_i)$. The combination of these two probabilities is called Hidden Markov Model.

Tag transition probabilities $p(t_i|t_{i-1})$ would be computed by counting the total number of tag t_{i-1} and t_i that are coming together in annotated corpus divided by total number of tag t_{i-1}.

$$p(t_i|t_{i-1}) = count(t_{i-1}, t_i)/count(t_{i-1})$$

For instance, the probability of NN given JJ in a given corpus is:

$$P(NN/JJ) = count(NN, JJ)/ count(JJ)$$

Word likelihood probability $p(w_i|t_i)$ also counts annotated corpus in the same manner as we do for the transition probability, as follows:

$$p(w_i|t_i) = count(t_i, w_i)/count(t_i)$$

It means that the likelihood probability of word w_i given tag t_i is the total number of tag w_i and t_i that are coming together in annotated corpus divided by total number of tag t_i.

For instance, the probability of *can* given NN is

$$P(can/NN) = count(NN, can)/count(NN) .$$

2.2.3 Stochastic versus Rule Based Approach

Automatic part-of-speech tagging is an area of natural language processing where statistical techniques have been more successful than rule-based methods [26, 29, 31]. The appeal of stochastic techniques over traditional rule-based techniques comes from the ease with which the necessary statistics can be automatically acquired and the fact that very little handcrafted knowledge need to be built into the system. Stochastic taggers have obtained a high degree of accuracy without performing any syntactic analysis on the input.

The stochastic approach has some especially desirable properties [38]:

- Robustness. Stochastic approach can be easily modeled by the so called n-gram model, and the inference calculation is simple arithmetic.
- Easy in obtaining the required knowledge from large set of corpora. For example, a large set of probabilities can be estimated using large on-line tagged corpora.
- Efficiency. Algorithms for the n-gram model are linear with respect to input length.
- Compatibility with the idea behind our probabilistic information retrieval model. This compatibility makes it possible to take advantage of our past experience in developing the tagger.

In contrast, the rules in rule-based systems are usually difficult to construct and are typically not very robust. Rule-based morphological analyzers employing a hand-crafted lexicon and a

hand crafted connectivity matrix.

2.3 Related Literature

In this section, researches that are conducted in the area of part-of-speech tagging for other natural languages are reviewed. The significance of the section is to see the trend in processing of natural languages and adopt crucial tools and techniques for the study.

2.3.1 Part-of-speech Tagging for English Text

This research reports on the implementation and empirical comparison of three supervised stochastic tagging approaches (Unigram Model, Hidden Markov Model and Viterbi algorithm) [36]. The data set used for training and validation is the tagged version of the Brown Corpus from the Penn Treebank 3 (PTB) corpus. The PTB collection contains 2,499 news stories from over three years of the Wall Street Journal.

Part-of-speech tagging model consists of three parameters in Hidden Markov Model approach.

1. Initial state probabilities. This is a vector that quantifies the probability of the first hidden state (tag) in a sentence. Since the first word in a sentence has no predecessor, it follow the idea of assuming the most frequent tag that the word has been observed with in the training data as the most probable tag for this word.
2. Sate transition probabilities. It is stored in a transition matrix; quantify the likelihood of observing one hidden state given the previous hidden state.
3. State emission probabilities. This is stored in a confusion matrix, specify the probabilities of observing a particular state (word) while the HMM is in a certain

hidden state.

Viterbi algorithm was used to decode the hidden sequence of tags for a given sequence of observation of words in a sentence. The essential intuition behind the Viterbi algorithm and its main advantage are the reduction of the complexity of examining every full path through a trellis by recursively finding partial probabilities for the most likely path from one state to the next.

The Viterbi algorithm requires three steps for searching and identifying one complete and most probable route through the trellis: Initialization, Induction and termination and path (most likely tag sequence) readout (by backtracking).

Viterbi algorithm was examined with respect to accuracy in three ways. The total accuracy implementation of Viterbi approximation was considered as (a) partial probabilities, back pointers and backtracking, (b) initial, transition and confusion probabilities, and (c) probabilities of single words.

Point (a) represents the heart of the Viterbi algorithm. It is the combination of the forward algorithm, back pointers and backtracking that enables Viterbi to find the globally best path through a trellis. This point is referred as Viterbi.

Point (b) represents the key idea of HMM, which is contained in the initialization and part of the induction step of the Viterbi algorithm. It is suggested that the numeric difference between points (a) and (b) represents the accuracy gain that Viterbi can provide over the mere concatenation of tags that are chosen as the maximal products out of all combination of state

transition and emission probabilities between subsequent words. In short, the difference between point (a) and (b) represents the difference between globally maximal searches and locally maximal search solutions. Here it is defined that local maximum is the outcome of induction that excludes partial probabilities from computation. This point is referred as HMM.

Point (c) resembles the initialization and initial state probabilities disregarding the impact of a word's predecessor on a word's POS and not making use of the relaxed independence assumption among words in UM. In HMM and Viterbi, the computation of point (c) applies to every first word in a sentence as well as one word sentences. This point is referred as the Unigram Model (UM).

In order to determine the accuracy of Viterbi (globally maximal solution), HMM (locally maximal solution), and UM (probabilistically maximal solution), multiple ten-fold cross validations were used. The validation was implemented by first randomly breaking the full corpus into ten partitions. Then, nine folds were used for training. Finally, all tags removed from the remaining fold used the three algorithms to tag the data in the tenth fold, compared the automatically assigned tags to the original labeling of the tenth fold, and recorded all deviations as errors. This procedure was repeated ten times and the error rates were averaged. As a result, 93.49%, 93.25% and 88.48% were reported for Viterbi, HMM and UM respectively.

2.3.2. Part-of-speech Tagger for Amharic

One of the researches conducted in the same approach with the study in local language is part-

of-speech tagger for Amharic [5]. Viterbi algorithm bigram model is used to develop a tagger prototype and visual basic programming language for implementing the prototype. Manually tagged sample corpus (290 Amharic words) was used for training and testing the prototype.

The training set is to develop transitional and lexical probabilities from which the tagger learns to tag sequence of words in a sentence. The test set is used to evaluate the performance analysis of the prototype tagger.

The prototype tagger has been evaluated in two ways. In the first performance, the tagger prototype was trained with training set and tested with the same training set. And for the second performance, 90% and 10% of the total data set has been used for training and testing respectively. As a result, Mesfin [5] reports 97% and 90% from the performance analysis of the tagger prototype with training set and test set respectively.

2.3.2. Part-of-speech Tagger for Arabic

As indicated in [33], since there were no annotated corpus and tagset for Arabic language, the researcher started from scratch from developing corpus for part-of-speech tagging for the language. Statistical and rule based methods were used in combination to develop parts of tagger for the language. Rule base method has been used to determine unknown words in the corpus. Stemmer for the language was used to identify suffixes of unknown words. Stemming is the process of removing all of a word's affixes to produce the stem or root. Affixes assist to tag the unknown words in the corpus. And for words in the corpus, statistical method has been used to disambiguate ambiguous words.

A corpus of 50, 000 words were extracted from different newspapers to drive annotated lexicon. 40% of the corpus has been used for testing and 60% used for training purposes. The performance analysis of the tagger has been trained and tested. So that, it has been reported, the statistical tagger has performed an accuracy of 90%.

2.3.3. Part-of-speech Tagger for Bangali

This paper [37] describes a work on building Part-of-Speech (POS) tagger for Bengali. Bangali is one of Morphologically Rich Languages. Supervised and semi-supervised bigram Hidden Markov Model (HMM) and Maximum Entropy (ME) based stochastic taggers were used to perform the task. Word's suffix information has been used for the unknown words.

The training data includes manually annotated 3625 sentences (approximately 40,000 words) for both supervised and semi supervised HMM and ME model. Models have been tested on a set of randomly drawn 400 sentences (5000 words) disjoint from the training corpus. The accuracy of supervised and semi supervised HMM and ME are 77.26%, 77.16% and 84.56% respectively.

2.3.3. Part-of-speech Tagger for Romanian

This research also implemented Hidden Markov Model for Part-of-Speech Tagging using Romanian Corpora [34]. Second order (trigram model) Viterbi algorithm has been used to implement the tagger. The basic algorithm is fairly straight-forward: at first, the tagger looks

up the dictionary for all possible tags that the current token can have, together with their respective lexical probabilities(i.e., the probability distribution of the possible tags for the word form). This is then combined with the contextual probability for each tag to occur in a sequence preceded by the two previous tags. The tag with the highest combined score is selected.

The corpus used in the experiments and evaluation reported was made of the integral texts in two books: Orwell's 1984 and Plato's The Republic. The amounts of corpus from each source are 117910 and 136960 respectively.

The training and testing processes has been done three times. The first training was done on 90% of “1984“, the second on 90% of “The Republic” and the third on the concatenation of the texts used in the first two (90% of each of the two books). The resulting language models were used to test the corresponding unseen 10% of the texts. As a result, 97.82%, 96.10%and 95.63% performance analysis were reported.

As discussed above, most part-of-speech tagging were conducted in HMM approach since it needs no detail knowledge of a language. Once the corpus, which represents a language, is identified and annotated with experts in the field, the rest process is the matter of computing the frequency probabilities and contextual probabilities of words in the corpus. And labeling words with their appropriate word categories based on information from these probabilities.

2.4. Conclusion

This chapter discusses about POS tagging. The chapter focuses on techniques of HMM that are implemented in chapter five. The next chapter discusses the word classes for Afaan Oromo language. The word classes are very significant in the study since it is the base to identify tagset that is implemented in chapter four.

Chapter Three

3. Word Classes of Afaan Oromo Language

3.1 Introduction

Words are traditionally grouped into equivalence classes called part-of-speech (known as word classes, morphological classes, or lexical tags). In traditional grammars there were generally only a few part-of-speech (noun, verb, adjective, preposition, adverb, conjunction, etc.) [23]. Whereas, currently there are many more word categories added for natural language processing in general and part-of-speech tagging in particular to identify words in sentences with their specific identities. For example, for English language, there are 36 word categories identified in "The Penn Treebank tagset" for natural language processing tasks [2].

This chapter identifies word classes (part-of-speech) for Afaan Oromo language for the purpose of the study. The identification of these word classes is based on the syntactical (structural) rule of the language in a sentence. This is because the role (whether a word is a noun, a verb or any other category) of a word is identified in a sentence based on their contextual position in a sentence.

As indicated in [14, 19], Afaan Oromo language has all the lexical categories of words known to exist. These are noun, verb, adjective, adverb, pronoun, preposition, conjunction and introjections. These categories are characterized based on lexical meanings of words. In other word, these word categories are classified based on their semantic coherence rather than considering their positional role or synthetic meaning in a sentence.

However, resent scholars argue that the classification of word categories should be characterized through out the text based on contextual position they occupy in a sentence [14, 15, 23].

Word classes do have tendencies towards semantic coherence (nouns do in fact often describe 'people, places or things', and adjectives often describe properties), but semantically categorizing words is not used for definitional criterion of word categories for part-of-speech tagging [23]. So the identification of word categories for part-of-speech tagging is based on their structural position in a sentence. For instance, the following two sentences indicate how much the syntactical position of words in a sentence changes the function (word class) of words.

Dabalaan mana *citaa* ijaare. ‘Debela built a thatched house’.
Dabalaan *citaa* fiduu dhaqe. ‘Debela has gone to bring thatched’.

In the first sentence, the word *citaa ‘thatched’* has taken the position of adjective to describe the types of house, where as in the second sentence it has placed the position of noun. So, even if the word is considered as noun based on its lexical meaning, it can also be categorized in other categories based on its contextual position in the sentence.

In any language, there are customized standard word orders in a sentence to transfer their message as complete as possible. A word order is a syntactic device that signals relationship between words in a sentence. For instance, in English and France, the order of words is subject – verb – object/complement. Whereas, in Amharic and Japan languages, the order of words is

subject – object – verb [27]. As a result, the order of words in a sentence is a clue to identify word categories.

As described in [19], in Afaan Oromo the order of words in a sentence is subject – object – verb, that commonly known as, subject-verb agreement. Let's illustrate it in the following sentence of Afaan Oromo.

Dabalaan muka mure. 'Dabala cut a tree.
In this sentence, *Dabalaan* is a subject, *muka 'tree'* is an object and *mure 'cut'* is a verb.

As indicated above, if words do not be arranged in their appropriate place in a sentence, their message will also be vague or has no meaning at all. For instance, while the sentence *Leeliseen hiriyashee faana dhufte 'Lelise came with her friend'* is correctly written following the order *subject – object – verb* of the language, the sentence *Faana dhufte Leeliseen hiriyaashee 'with came Lelise with friend'* written in the order *verb – subject – object* which do not follow the right order of words.

Morphological derivations of the language are also another factor to identify categories of words. In the following words, we can see the change of word categories from one class to another due to derivation.

dheeraa 'long' ---------- dheerina 'height' → from adjective class to noun class
furdaa 'fat' ---------- furdisuu 'make fat' →from adjective class to verb class

Accordingly, this chapter identifies the lexical categories of words in Afaan Oromo based on their contextual position they occupy in a sentence and follow the same pattern to tag corpus manually for implementation purpose. Identification of lexical categories here is very significant since the discussion of the following chapter depend up on these identified word categories.

Resent studies categorize Afaan Oromo words in to five classes which includes nouns, verb, adverb, conjunction and adoption. Pronoun and adjectives are categorized under noun. Preposition and postpositions are categorized under adoptions [14, 19]. Since the study is going to develop a system that labels each word in a sentence with its unique category and subtypes in a category, the section discuses all eight categories (those indicated by traditional grammarians) and numerals. So in the following sections, major lexical categories (noun, pronoun, verb, adjective, adverb, preposition, conjunction, numerals and introjections) of Afaan Oromo are discussed. Writing system and punctuation in the language also discussed to briefly overview the alphabet and punctuation system of the language.

3.2 The Writing System and Punctuation Marks of Afaan Oromo Language

Currently, Afaan Oromo is commonly written with a modified Latin alphabet called Qubee Afaan Oromo, which was formally adopted in 1991 [19]. Qubee Afaan Oromo aligned with so many countries that use the Latin script including English. One obvious advantage of this is that the adaptability to computer technology which gives alphabetic writing. There are 36 basic alphabets in Qubee Afaan Oromo, 26 consonants and 10 vowels. All consonants and vowels of Afaan Oromo were discussed in detail in [20].

Afaan Oromo has the typical Southern Cushitic set of five short and five long vowels, indicated in the orthography by doubling the five vowel letters. The difference in length is contrastive, for example, *lafa 'earth', laafaa 'soft'*. Gemination is also significant in Afaan Oromo. That is, consonant length can distinguish words from one another, for example, *badaa 'bad', baddaa 'highland'*.

Punctuation marks in Afaan Oromo follow the same punctuation pattern used in English and other languages that follow Latin writing system.[1]

As it is mentioned above, identification of word categories is very significant for the study since the main objective of the study is automatically labeling words with their appropriate categories (word classes) in corpus.

3.3 Word Categories

Here under the various word categories in Afaan Oromo language is discussed.

3.3.1 Noun Categories

A noun is any word which names a person, place, thing, idea, animal, quality, or activity [22]. Since lexical classes like noun are defined functionally (morphological and syntactically), some words for people, places, and things may not be nouns, and conversely some nouns may not be words for people, places, or things. So, based on its contextual position, nouns are

[1] see [9] for more detail description

commonly found at the beginning of a sentence in Afaan Oromo. For instance, words that are italic in the following sentences are nouns.

Diimaan filannoo kooti. ‘Red is my preference’.

Sareen manatti olixxe ‘ Dog enters into a house’

[14] described nouns of Afaan Oromo in a sentence as follows. How noun is described? Noun is forms which can take the position of *hoolaa ‘sheep’* in the following sentences.

Hoolaa bituun gaaridha ‘buying sheep is good’.

akka hoolaa ‘like sheep’.

hoolaa kana ’this sheep’.

In this structure, hoolaa ‘sheep’ may be replaced by farda ‘hourse’, mana ‘house’, muka ‘tree’, etc.

Words that are categorized as noun in a sentence could be a subject or an object. Subject mostly comes at the beginning whereas object comes after an object and in front of a verb in a sentence. Let’s see the position of subjects and objects in the following sentences.

Gamadaan qotee bulaadha ‘Gamada is a farmer’.

Leelisaan qotiyyoo lama qaba. ‘Lelisa has two oxen’.

In the above two sentences, *Gamadaan* and *Leelisan* are subjects and *qotiyyoo ‘oxen’* is an object since the word *qotiyyoo* found in front of the verb *qaba* and after the subject *Leelisaan*.

In Afaan Oromo, nouns are also characterized in singular and plural numbers. Singular nouns are nouns which are quantified and not add any suffixes in the language. For instance, *farda 'hourse', re'ee 'goat', nama 'man/women', mana 'house'* and so on. All of them indicate single entity or animal or person. Quantified plural nouns in Afaan Oromo can be formed in two ways. The first way is, by adding numerals after singular nouns. For example, *farda lama 'two horse', re'ee sadii 'three goat', nama shan 'five men'* and so on. Plural nouns are also formed through the addition of suffixes in the language. The most common plural suffix is *–oota, -ota, - wwan, -een (a)an, -lii,* and *lee*. Look the following table for examples.

Singular	Plural
Sangaa 'ox'	Sang**oota** 'oxen'
Barsiisaa 'teacher'	Barsiift**ota** 'teachers'
Mootii 'King'	Moot**ota** 'Kings'
Mucaa 'child'	Mucoo**lii** 'children'
Gaangee 'mule'	Gaango**lii** 'mules'
Maatii 'family'	Maatii**wwan** 'families'
Hojii 'work'	Hojii**lee** 'works'
Waggaa 'year'	Waggaa**wwan** 'years'

Table 3.1: Plural forms of nouns that are formed from suffix

Like most other Afro-Asiatic languages, Afaan Oromo has two grammatical genders, masculine and feminine, and all nouns belong to either of the two.

As described in [15, 19, 38], a small number of nouns pairs for people (masculine and feminine), however, end in *-eessa (m.)* and *-eettii (f.),* as do adjectives when they are used as nouns: *obboleessa 'brother', obboleettii 'sister', dureessa 'the rich one (m.)', dureettii 'the rich one (f.)'*. Grammatical gender normally agrees with biological gender for people and animals; thus nouns such as *abbaa 'father', ilma 'son',* and *sangaa 'ox'* are masculine, while nouns such as *haadha 'mother'* and *intala 'girl/daughter'* and *sa'a 'cow'* are feminine. However, most names for animals do not specify biological gender.

Nouns in Afaan Oromo also formed through derivation from other classes. For example, the following words are nouns derived from other classes:

Furdaa 'fat' – Furdina 'fatness' = > from adjective to noun
Eeraa 'long' – Eerina 'height' = > from adjective to noun
Qaba 'have' – Qabeenya 'wealth' = > from verb to noun

Nouns in Afaan Oromo are also grouped into proper nouns and common nouns as the same as other languages like English. Proper nouns, like *Nagaraa 'Negera', Jimmaa 'Jimma',* and, *Rabbi 'God'* are names of specific persons or entities. Their first letter is usually capitalized.

Common nouns are divided into count nouns and mass nouns. Count nouns are those that allow grammatical enumeration; that is, they can occur in both the singular and plural (*sangaa 'ox' /sangoota 'oxen', mana 'house'/maneen 'houses'*) and they can be counted (*sangaa tokko 'one ox', sangaa lama 'two ox'*). Mass nouns are used when something is conceptualized as a homogeneous group. For instance, *xaafii 'tef', bishaan 'water'* and *qileenssa 'wind'*.

3.3.2 Pronoun Categories

Pronoun is a word that is used instead of a noun or noun phrase [42]. Afaan Oromo pronouns are also used in place of nouns as pronouns of other languages. As nouns, pronouns characterize based on number and gender. For instance, *ishee 'her', isa 'him', isaan 'they'* are some. There are types of pronouns in the language based on their function and meaning in a sentence. These are personal pronouns, possessive pronouns, demonstrative pronouns, and reflexive and reciprocal pronouns. Personal pronouns are *ani 'I', ati* (singular) *'you', inni 'he', isiin/ishiin 'she', isin* (plural) *'you', nuy/nuti 'we'* and *isaan 'they'* are used in the subject position of nouns in the sentences. For instance, the following two sentences describe the fact.

Caalaan kaleessa dhufe 'Chala came yesterday.'
Inni kaleessa dhufe 'he came yesterday'.

In the above two sentences, *Inni 'he'* replaces the subject *Caalaa*, they have the same meaning.

Personal pronouns that replaces objectives in sentences are *ana 'me', si* (singular) *'you' isin* (plural) *'you', isa 'him', ishee 'her', nu 'us'* and *isaani 'them'*. For instance,
Sareen baratootatti dute 'Dog barks at students'.
Sareen *isaanitti* dute 'Dog barks at them'.

In this example, *isaan 'them'* replaces the object *baratoota 'students'*.

Possessive pronouns also one parts of Afaan Oromo pronoun that indicates the ownership of

something in a sentence. Possessive pronouns includes *koo/kiyya 'mine'*, *kee*(singular) *'yours'*, *(i)saa 'his'*, *(i)shii 'her'*, *keenya 'ours'*, *keessan*(plural) *'yours'* and *(i)saanii 'their'*. For example,

Mani barumsa kun kan *keenya* 'this school is ours.

Konkolaataan *saa* ni cabe 'his car was broken'.

Abbaan shee dhufee ture 'her father came'.

Demonstrative pronouns used to refer objects mentioned earlier or which are always present in the speaker's mind. Like English, Afaan Oromo makes a two-way distinction between proximal ('this, these') and distal ('that, those') demonstrative pronouns and adjectives. Some dialects distinguish masculine and feminine for the proximal pronouns. Unlike in English, singular and plural demonstratives are not distinguished, but, as for nouns and personal pronouns in the language, *case* is distinguished. The following table indicates base and nominative forms.

Case	Proximal ('this, these')	Distal ('that, those')
Base	Kana [tana (f.)]	San
Nominative	Kuni [tuni (f.)]	Suni

Table 3.2: Case and Nominal form of Demonstrative Pronoun

Afaan Oromo has two ways of expressing reflexive pronouns ('myself', 'yourself', etc.). One is to use the noun meaning 'self': *of(i) or if(i)*. This pronoun is inflected for *case* but, unless it is being emphasized, not for person, number, or gender: *isheen of laalti 'she looks at herself'* (base form of *of*), *isheen ofiif makiinaa bitte 'she bought herself a car'* (dative of *of*). The other possibility is to use the noun meaning *mataa 'head'*, with possessive suffixes: *mataa koo 'myself'*, *mataa kee 'yourself (s.)'*, etc.

Afaan Oromo has also a reciprocal pronoun *wal 'each other'* that is used like *of/if*. That is, it is inflected for case but not person, number, or gender: *wal jaalatu 'they like each other'* (base form of *wal*), *kennaa walii bitan 'they bought each other gifts'*.

3.3.3 Verb Categories

The verbs are words or compound of words that expresses action, a state of being and/ or relationship between two things [23]. In its most powerful and normal position, it is found at the end of the sentence in Afaan Oromo. Consider the following examples:

Lamiin kaleessa *dhufe.* 'Lami came yesterday'.
Lamiin hin *dhukubsata*. 'Lami was sick'.
Lamiin farda *bite*. 'Lami bought a horse'.

In the above examples, all words that are italic are verbs. They appear at the end of the given sentences.

Intransitive, transitive and auxiliaries are major subcategories of verbs in Afaan Oromo [9,14].

Intransitive verbs are verbs which do not take object or complement in a sentence. For example,

makiinichi *cabe* ‘the car broke’.

Isaan kaleessa *dhufani* ‘They came yesterday’.

Aliin ni *fiiga* ‘Ali is running’.

In the above sentences, all italic words are intransitive verbs. They appear at the end the given sentences and do not transfer message from subject to complement.

Transitive verbs are verbs which transfer message to complements (objects). For instance,

Tuluun burcukoo *cabse* ‘Tulu broke a glsaa’.

Caalaan muka *mure* ‘Cala cut a tree’.

In the above two sentences, *cabse ‘broke’* and *mure ‘cut’* are transitive verbs since they interrelated subjects and objects in the sentences.

The other main subtype of verb categories is auxiliary verbs. Auxiliary verbs are verbs that support the main verb used in a sentence. Some auxiliary verbs in Afaan Oromo are indicated in the following examples.

Isheen baratuu-*dha* ‘She is a student’.

Leeloon ingineera *ta’e* ‘Lelo becomes an engineer’.

Inni mana barumsa deemee *ture* ‘he was going to school’.

Hojii kana hojjachuu *qabda* ‘you have to work this job’.

All words in italic are auxiliary verbs in Afaan Oromo language.

3.3.4 Adjective Categories

Adjectives in a sentence modifies nouns to denote quality of a thing; that is, it specifies to what extent a thing is as distinct from something else [17]. For example,

Aliin *guracha* ‘Ali is black’

inni *jarsa* ‘he is old’.

Aliin qotee bulaa *jabaadha* ‘Ali is strong farmer’.

In the above sentences, all words (*guracha ‘black’, jarsa ‘old’*, and *jabadha ‘strong’*) are adjectives.

Adjectives in Afaan Oromo sentences can be formed from original adjectives and compound words. For instance, as mentioned above, *guracha ‘black’, diimaa ‘red’, eeraa ‘long’* and so on are original adjectives. *Humna dhabeessa ‘weak’, bifa dhabess ‘ugly’, simboo qabeessa ‘handsome’* are some of the adjectives formed from compound sentences.

Adjectives also characterize number and gender in Afaan Oromo. The following examples indicate adjectives in numbers and gender.

Singular	Plural	Feminine	Masculine
Guracha	Gurachota	Guracha	Gurattii
Dheeraa	Dheerota	Dheeraa	Deertuu
Jarjaraa	Jarjaroota	Jarjaraa	Jarjartuu
Ko'eessa	Ko'eeyyii	Ko'eessa	Ko'eettii

Table 3.3: Plural and, Feminine and masculine forms of adjective

3.3.5 Adverb Categories

Adverbs are words which are used to modify verbs. In Afaan Oromo, adverbs often precede verbs that they modify. In the language, adverbs could be categorized as adverbial time, adverbial place and adverbial condition. Adverbial time includes words like amma *'now'*, *yoom 'when'*, *har'a 'today'*, *kaleessa 'yesterday'*, *bor 'tomorrow'*, *iftaan 'after tomorrow'* and the like. The following examples show words of adverbial time in the language.

Fayyisaan *kaleessa* dhufe 'Feyisa came yesterday'
Inni *har'a* deeme 'he goes today'

In the above sentences, *kaleessa 'yesterday'* and *har'a 'today'* are adverbial time in the language.

Adverbial place responses the question 'where?' in a sentence. Words includes *achi 'there'*,

gubbaa 'above', jidduu 'middle', bira 'together', gadi 'below' and the like are some of adverbial places in the language. For example,

Aliin *mana* jira 'Ali is at home'

Aliin *diidaa* hojjata 'Ali is working outside'.

Isheen *iddoo* hoteelaa jirti 'she is at hotel'.

Mana 'home', diida 'outside' and iddoo 'at' are adverbs of place in the above sentences.

Adverbial manner is one sub categories of the verb in a sentence that response the question 'how?' The following examples illustrate use of some adverbial manner in sentences.

Suuta deemi 'go slowly'

Inni *qajeelcheetu* hojjeta 'he works well'.

Isheen *bay'ee* cimtuudha barumsa sheetti 'she is very strong in her education'.

Suuta 'slowly', qajeelchee 'well' bay'ee 'verb' and *cimtuu 'strong'* are adverbial manners in above sentences.

3.3.6 Preposition Categories

In constituent structures, their position may precede or follow the category with which they form a syntactic unit. On the bases of this, they may be referred to by the less general terms, prepositions or postpositions [14]. Some preposition and postposition in the language includes *akka 'as', eegasii 'since', hamma 'until', gara 'to', gadi 'below', oli 'above', irra 'on'* and so

on. For instance, let's see the following sentences.

Aliin *gara* hospitaalaa adeeme 'Ali goes to hospital'.

Aliin buna mana saa *duuba* dhaabe 'Ali planted coffee at the back of his house'.

In the first sentence, *gara 'to'* is a preposition precedes *hospitaala 'hospital'* and in the second sentence, *duuba' back'* is the postposition. It comes after *mana 'house'* with which it forms a syntactic unit.

3.3.7 Conjunction Categories

Conjunctions are words that join words, phrases or sentences [42]. It is categorized in to coordinating conjunctions and subordinating conjunctions. Coordinating conjunctions in Afaan Oromo can join two main clauses that a user wants to emphasize equally. These conjunction includes *akkasumas ' besides/in addition to', garuu 'but', haata'u malee 'however', kanaafuu 'so/therefore', ta'ulee 'eventhough'* and so on. For instance,

Aliin mana ijaarateerra *garuu* konqolata hinqabu 'Ali built a house but he has no a car'.

Ati magaala *moo* badiyaa irra jalata 'which do you like city or rural'.

In these two sentences, *garuu 'but'* and *moo 'or'* are coordinate conjunctions. In the first sentence, two sentences (*Aliin mana ijaarateerra* and *konqolata hinqabu*)are joined by coordinate conjuction *garuu 'but'*. In the second sentence, two words (*magaala* and *badiyaa*) are joined by *moo 'or'*.

Subordinate conjunctions are used to join main clause with subordinate clause. They includes *otoo 'if', yoo 'as if', yennaa 'when', hamma 'until', erga 'after', dura 'before'* and so on. For example, in the sentences,

Yennaa Abdii dhufe Loomeen laqanshee nyaatti 'When Abdi came Lome was eating her lunch'.

Hamma ani dhufutti kana hojjadhuu na'eegi 'until I come do this job'.

Otoo haalaan qo'atee qormaata ni dabarta 'if you work hard, you will pass the exam'.

Erga inni dhufe hojjette 'she did it after he came'.

In the last sentence of above given examples, subordinate conjunction *erga 'after'* joins main clause *isheen hojjette'she did'* and subordinate clause *inni dhufe 'he came'*.

3.3.8 Numeral Categories

Numerals include all elements which refer to quantity or amount. They can be cardinal and ordinal numbers. As Baye[14] indicated, commonly the position of numerals follows the category with which they form a syntactic unit. The following example illustrates both cardinal and ordinal numbers in Afaan Oromo.

Among cardinals, *sadii 'three', afur 'four', dhibba 'hundred'* are some examples. Numerals in Afaan Oromo also formed in compound words. They are put separate like in English. *Dhibba lama 'two hundred', kuma tokko' 'one thousand'* and so on are some examples.

Ordinal number is a number that refers to the position to something in a series [42]. In Afaan Oromo ordinal number takes the suffix *–ffaa*. For example, *Tokkoffaa 'first'*, *lamaffaa 'second'*, *sagalaffaa 'ninth'* and so on.

3.3.9 Introjections Categories

Introjections in the language are words that have unique functions to express emotion, sudden surprise, pleasure, sadness, and so on[19]. Afaan Oromo has many words of introjections includes *oo* 'to express emotion when some new things happen', *ah* 'to order to be silent when some new sound is heard from outside', *wayyoo*, 'to express sadness' *waa* 'when someone give warning to somebody else', and so on.

3.4 Tags and Tagset for Afaan Oromo

The previous section (3.3) gave broad descriptions of the kinds of lexical classes that Afaan Oromo words fall into. In this section we identify and describe the actual tags and tagset used in part-of-speech tagging for the study. Tags are labels that add more information on each word in sentences [22]. Tagset is a collection of tags that are identified and used for developing a prototype for the study.

Since there is no readily available tag sets for the language, the tag sets for the study are identified and developed in this section. Identifying and developing detail of tag set needs human experts and time consuming. Therefore, for the present study, broad category of tagset is developed.

3.4.1 Noun Tags

As described in section 3.3.3, nouns in Afaan Oromo characterize numbers (singular and plural form) and genders (masculine and feminine). Besides, they are also categorized in to proper and common nouns. But, in the present study, all these subtypes of nouns are assigned NN without distinction as indicated in the following examples.

Dablaan/NN kara mana/NN keenyaa darbe ‘Dabala passed through our home’.

Jaldeessi/NN boqqollo/NN ba’ee baleesse ‘Monkey destroyed maize very much’.

Manni/NN keenyi karaa duradha ‘Our home is on a road’.

Months of the year in Afaan Oromo are all assigned the tag NN. Abdiin Amajjii/NN keessa dhalate ‘Abdi was born in January’. Days of the week of the language are also assigned NN when they are used as nouns. For instance,

Dilbanni/NN guyyaa boqonaati ‘Sunday is a day of rest’.

Directions (*baha ‘east’, dhiha ‘west’, kaabafi ‘north’ kibba ‘south’*) in Afaan Oromo and initials in names (*Obbo ‘Mr’, Aadde ‘Mrs’, Durbee ‘Miss’*, and so on) are also assigned NN as indicated in the following examples.

Wallaggaan karaa lixa/NN Oromiyaatti argamti ‘Wellega is found at the west of Oromia state’.

Obbo/NN Gamadaan dura ta’a ganda keenyaatti ‘Mr Gamada is the leader of our kebele’.

Durbee/NN Leenseen kaleessa walga’hii haalaan gaggeesite ‘Miss Lensa led the assembly in a

good manner'.

There are special subtypes of noun tagset in the language that are joined with other categories like in Amharic [5]. These are noun words joined with postpositions (unlike in Amharic joined with preposition) and conjunctions. For nouns that are not separated from postpositions are assigned NP. Nouns that are not separated from conjunctions are assigned NC. The following examples illustrate those special subcategories of noun in sentences.

Leensaan konkolataan/NP gara mana barumsa dhaqte 'Leensa went to school by car'.

Leensaafi/NC Nagaraan mana barumsa dhaqani 'Lensa and Negara went to school'.

In the first sentence the noun *konkolata 'car'* and the postposition *n 'by'* joined together as (*konkolataa* **- *n***). There are many such kinds of words in Afaan Oromo as shown below.

Fardaan 'on horse back'

Lukaan 'on foot'

Uleen 'by stick'

Leensaaf 'for her'

In the second example, the noun *Leesaa 'Lensa'* and conjunction word *fi 'and'* are joined together as one word. The following are also more examples of words that are assigned NC.

Tulunis/NC mana barumsa dhaqe 'Tulu also has gone to school'.

Dameenis/NC ta'e Biiftuun barataniiru 'Besides Dame, Bifftuu also learned'.

Nis 'besides' and *fi 'and'* are common conjunction words that are do not separated from nouns and pronouns in the language.

3.4.2 Pronoun Tags

All subtypes of pronouns (personal pronouns, possessive pronoun, demonstrative pronouns, reflexive pronouns and reciprocal pronouns) as described in section 3.3.2, are assigned PP. For instance,

Isheen/PP gara mana barumsa dhaqte 'she has gone to school'.
Isaan/PP kaleessa dhufani 'They came yesterday'.
Leensaan gara isaa/PP dhaqte 'Lesa goes to him'.
Hojii kana uni/PP hojachuu qabna 'We have to work this job'.
Kuni/PP kan keenya/PP 'This is ours'.

As noun, pronouns are also characterized by numbers and gender, which can be categories of pronouns in to subtypes. All of them are assigned PP without discrimination. There are also special subtypes of pronouns similar to nouns in the language. These are pronoun words that do not separate from postposition and conjunction. For such kinds of words special tags are assigned. For all words that do not separate from postposition PS is assigned and for those words that do not separate from conjunction PC is assigned. Let's see them in the following examples.

Hojiin isheetiin/PS hojetame sirri miti 'Works done by her are not good'.

Kaleessa inniifi/PC isheen wliin turani ‘Yesterday he and her were together’.

Innis/PC ta’e isheen hojii hin qabani ‘he or she is idle’.

In the first sentence of given examples, the phrase isheetiin ‘by her’ formed from two words ishee ‘her’ (pronoun) and tiin ‘by’ (postipostion). The phrase is not purely pronoun or postposition. So that special tag PS is assigned.

3.4.3 Verb Tags

In addition to nouns, verbs are one of the major constituent in a sentence. Verbs in Afaan Oromo are categorized into two major categories; namely, main verbs and auxiliary verbs. For the purpose of this study all subtypes of main verbs (intransitive, transitive and so on) are assigned VV. As described above, verbs in the language characterize number, gender and tenses. All of them are known as VV without any discrimination for the present study. The following examples indicate how subtypes of main verbs in the corpus are labeled.

Inni mana ijaare/VV ‘He built a house’.

Inni rafaa/VV jira ‘he is sleeping’.

Isaan Leensaa faana mana barumsa dhaqani/VV ‘They have gone to school with Lensa’.

For all attributes and values of attributes of auxiliary verbs AX are assigned. The following examples illustrate some of the words that are assigned AX in the study.

Leeloon injineera ta’e/AX ‘Leeloo becomes an engineer’.

Inni Amarikaanii dhaqeetu ture/AX ‘he was in America’.

Hojii kana haalaan hojjachuu qabnaAX ‘We have to work this work’.

Ishee konkolataa oofuu dandeessi/AX ‘She can drive a car’.

Dabalaan mana kana jiraataa/AX ture/AX ‘Dabala used to live this house’.

3.4.4 Adjective Tags

For the purposes of the study, all sorts of adjectives in Afaan Oromo are assigned JJ. That means it does not differentiate the specific features of adjectives (gender, number and their values as indicated in section 3.3.4). In the following sentences adjectives are indicated that are assigned JJ.

Dabaloon furdaadha/JJ ‘Dablo is fat’.

Dabaloon mana guddaa/JJ ijaare ‘Dabalo built a big house’.

Lamiin mana citaa/JJ keessa jiraata ‘Lami lives in thatched house’.

Inni sangaa guracha/JJ qaba ‘he has a black ox’.

Leenseen dheertuudha/JJ ‘Lense is long’.

For adjective words that do not separate from conjunctions JC are assigned as indicated in the following examples.

Inni furdaafi/JC eeraadha ‘he is fat and long’.

Isheen guraatiifi/JC gabaaduudha ‘she is black and short’.

Numerals in the language also function in place of adjectives in sentences. For such kinds of words JN are assigned. The following examples illustrate such words in sentences.

Kitaabicha qarshii dhibbaan/JN bite ‘I bought the book in hundred birr’.

Baratoota sadii/JN qofaatu dhufe har’a ‘Only, three students come today’.

3.4.5 Adverb Tags

Adverbs in Afaan Oromo have three major subcategories (adverbial time, adverbial manner and adverbial place) as discussed in section 3.4.5. But for the purpose of the study all adverbial are assigned AD without sub distinctions as indicated in the following examples.

Dabalaan suuta/AD deema ‘Debela is going slowly’.

Dabalaan amma/AD deeme ‘Debela has gone now’.

Leensaan kaleessa/AD deemte ‘Lensa went yesterday’.

Inni amaan/AD booda/AD hin dhfu ‘he does not come after this time’.

Hunduma durseetu\AD dhufe ‘He came before all’.

Days of the weeks are assigned AD when they are used as adverb in a sentence as shown in the following examples.

Isheen gaafa Jimaataa/AD dhufti ‘she will come on Friday’.

Dilbata/AD Dilbata/AD waaqeffannaan deema ‘I go worshipping every Sunday’.

3.4.6 Preposition Tags

As it is discussed in section 3.4.1 and 3.4.2 some words of prepositions in Afaan Oromo are not separated from nouns and pronouns. For example, in a sentence, *hojiin isheetiin hojjatame*

'works done by her' the word *isheetiin 'by her'* is formed from two words *ishee 'her'* and *tiin 'by'*. So, it is difficult to assign PP (pronoun) or PR (preposition) because they are joined together as one word. For the study NP and, PS are assigned for postpositions that are not separated from nouns and pronouns respectively as indicated under 3.4.1 and 3.4.2.

For all other prepositions that are do not join with other words PR are assigned as indicated below.

Leeloon erga/PR dukanaa'ee dhfe 'Lelo came after noon'.

Inni gara/PR mana yaalaa deeme 'he has gone to health center'.

Qonnaan bultotni waa'ee/PR tekinoloji qonnaa baratani 'Farmers have learned about farming technology

Inni alatti/PR bahe 'he outs to the field'.

3.4.7 Numerals Tags

The major categories of numerals in Afaan Oromo are cardinal and ordinal numerals. Cardinal numerals are almost functioning in place of adjectives. So, it is threatened under adjective tagset and assigned JN. The other major category of numerals in the language is ordinal numerals and all are assigned ON as shown in the following examples.

Leeliseen kutaa sheeti tokkoffaa/ON baate 'Lelise stood first in her class'.

Jimaani guyyaa shanaffaadha/ON turban keessaa 'Friday is the fifth day in the week'.

3.4.8 Conjunction Tags

As similar to prepositions, conjunctions also have characteristics of joining with other words of categories, specially, with nouns, adjectives and pronouns in the language. For instance, consider the following sentence.

Lamiifi Leensaan dhufani 'Lemi and Lensa have come'.

In this sentence, the word *Lamiifi* formed from *Lamii* and *fi 'and'*. These problems are treated as discussed in section 3.4.1. And conjunction words that do not separate from pronouns and adjectives are treated under section 3.4.2 and 3.4.3 respectively.

For all separated conjunctions CC are assigned. The following examples illustrate words that are assigned CC in a corpus.

Oromiyaan saba baa'ee akkasumas/CC qileensaa gaarii qabdi 'Besides many people Oromia has suitable whether condition.

Yoo bokkaan robe malee/CC margi hin margu 'without rain grass can not grow'.

Waan/CC isheen hamtuu taateef namni ishee hin jaalatu 'Since she has bad behavior, people do not like her'.

Akkuma/CC inni mana seeneen isheen mana baate 'As soon as he enters home, she went out'.

3.4.9 Introjections Tags

All introjections in Afaan Oromo are assigned II. The following sentences show how

introjections in the language are labeled with II.

Oh/II! Nama dhiba ‘ oh it is incridable’.

Wayyoo/II! ‘Sorry’

3.4.10 Punctuation Tags

All forms of punctuations in Afaan Oromo are assigned PN. Such punctuations includes ?, ., !, =, >, < and so on.

3.5 The Tagset for Afaan Oromo

The following table gives summary of tag sets that are used in the implementation of the study.

Tag	***Description***	***Example***
NN	A tag for all types of nouns that are not joined with other categories in sentences.	Dabalaa, mana, sa’a
NP	A tag for all nouns that are not separated from postpositions.	Leeloon, konkolataan
NC	A tag for all nouns that are not separated from conjunctions.	Jiraafi, re’eefi
PP	A tag for all pronouns that are not joined with other categories.	Ishee, isaan, isa

PS	A tag for all pronouns that are not separated from postpositions.	Isheetiin, isaaniin
PC	A tag for all pronouns that are not separated from conjunctions.	Isheefi, innis
VV	A tag for all main verbs in sentences.	Mure, qote, reebe
AX	A tag for all auxiliary verbs.	Ta'a, dha
JJ	A tag for all adjectives that are separated from other categories.	Furdaa, dheeraa, gababaa
JC	A tag for adjectives that are not separated from conjunction.	Dheeraafi, gababaas
JN	A tag for numeral adjectives.	Sadii, shan
AD	A tag for all types of adverbs in the language.	Kaleessa, turban
PR	A tag for all preposition/postposition that are separated from other categories.	Eega, gara
ON	A tag for ordinary numerals.	Tokkoffaa, shanaffaa
CC	A tag for all conjunctions that are separated from other categories.	Kanaafuu, haata'u malee
II	A tag for all introjections in the language.	Ah!, wayyoo
PN	A tag for all punctuations in the language.	., <,>

Table 3.4: Tagset for Afaan Oromo

3.6 Conclusion

In this chapter, word categories for Afaan Oromo language are discussed. Since there are no prepared word categories for the language, seventeen broad word categories are identified for this study. It is difficult to prepare detail specific word categories in limited time and resources since it needs human experts in the field. Word categories that are identified here are bases for implementation of the study. The next chapter emphasizes on preparation of sample corpus and implementing of the study.

Chapter Four

4. Implementation and Performance Analysis

4.1 Introduction

The previous chapter discusses the general word categories and tagset for Afaan Oromo language which are implemented in this chapter.

As it is discussed in previous chapter in detail, seventeen tags have been identified for the purpose of the study. For instance, all subtypes of nouns in the language are assigned NN except nouns joined with prepositions/postpositions and conjunctions that are assigned NP and NC respectively. So, the implementation of the study is based on these seventeen tags that have been identified as a tagset.

Besides the absence of tagset for the language, lack of corpus that represents the language also other great issue in the study. So that preparation of corpus is one concern that is discussed in this chapter because corpus is also one of the basic components of the study. The chapter also discusses about all the preprocessing steps to develop the tagger including developing two major databases (lexical probabilities and transition probabilities that are discussed conceptually in chapter two) that are used in the implementation of Viterbi algorithms.

The following Figure shows the general design and tagging processes of Viterbi algorithm.

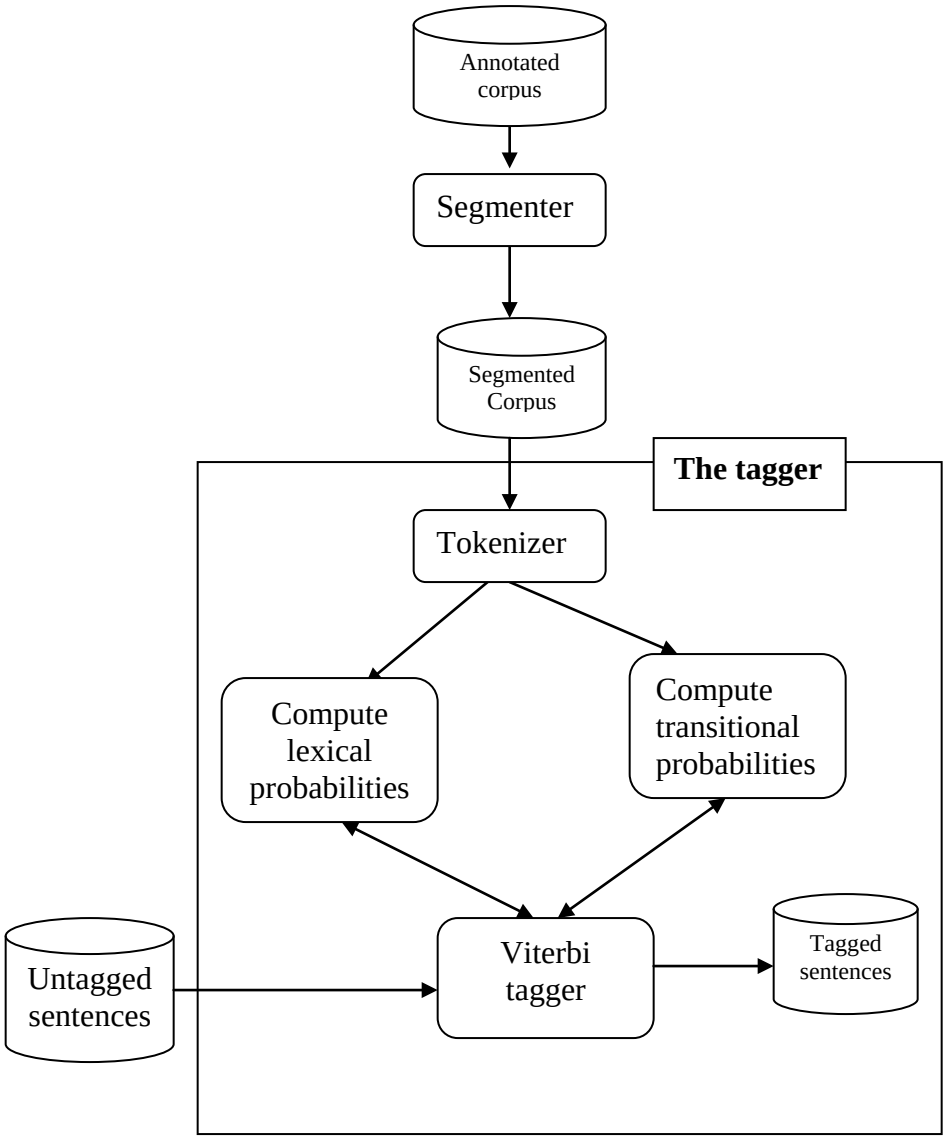

Figure 4.1: The Graphical Representation of Implementation Processes

The tagger starts tokenizing segmented corpus that is segmented by sentence segmenter. Segmented corpus is a corpus that is segmented into sentences. Then two probabilities; namely, lexical probabilities and transitional probabilities are computed. These probabilities are from which the tagger learns to tag sequence of words in sentences. That means the tagger gets the highest probability of category for a given word from lexical probabilities and its bigram information from transitional probabilities to identify the appropriate tag for the word

in a sentence. Finally, it labels the appropriate tag and writes the tagged sentences to secondary storages.

The detail processes are shown in the following section.

4.2 The Sample Corpus and the Manual Tagging Process

A corpus is a collection of texts or speech stored in an electronic machine-readable format [22]. Untagged corpus, as the name indicates, is a collection of raw texts that are not annotated with their appropriate part-of-speech categories. For supervised training and performance analysis and computing probabilities of words we need a tagged corpus.

Balanced corpus is needed to process natural language processing tasks like part-of-speech tagging. Balanced corpus is a corpus that represents the words that are used in a language. As indicated in [22], texts collected from a unique source, say from scientific magazines, will probably be biased toward some specific words that do not appear in everyday life. Such types of corpuses are not balanced corpus so that they are not appropriate for many natural languages processing in general and parts speech tagging in particular except for special purposes.

However, developing a balanced corpus is one of the difficult tasks in NLP research because it requires collecting data from a wide range of sources: fiction, newspapers, technical, and popular literatures. As a result it requires much time and human effort.

For this particular study, corpus was collected from different popular Afaan Oromo newspapers (Bariisaa, Bakkalcha Oromiyaa and Oromiyaa) and bulletins (Qabee and Oromiyaa) to balance the corpus. For many part-of-speech tagging researches, for other languages, newspapers and bulletins used to collect balanced corpora [33, 34]. Newspapers, bulletins and public magazines are considered as consisting different issues of the community: social, economical, technological and political issues. So that they are a potential source for collecting balanced corpus for natural language processing tasks.

The collected corpus for this study has been manually tagged by experts especially by linguists in the field. The tagging process is based on the identified tagset and corpus that is manually tagged, considering contextual position of words in a sentence. This tagged corpus is used for training the tagger and evaluates its performance. The total tagged corpus consist 159 sentences (the total of 1621 tokens).

4.3 Evaluation Procedures

In order to evaluate the tagger, first, the tagger is trained with the training set and run on the untagged 20% of the training set and then the output of the tagger is compared with manually tagged portion of this training set. The tagger errors are corrected and this step is repeated until the satisfactory result is obtained.

In the next evaluation mechanism, ten-fold cross validations are performed to determine the accuracy of the tagger [36]. The validation is implemented by first stratified randomly breaking the full corpus into ten partitions. Then, nine folds are used for training. And all tags

are removed from the remaining fold and then used the algorithms to tag the data in the tenth fold, compared the automatically assigned tags to the manually labeled of the tenth fold, and recorded all deviations as errors. This procedure is repeated ten times and the accuracy rates are averaged.

According to Jurafsky and Martin [23], taggers are, typically, evaluated by running the Gold Standard test and/ or comparing the results to the manually tagged test. The Gold Standard represents the performance measure - the accuracy rates for Viterbi by determining the portion of tagged words that the tagger and a human-labeled validation set agree.

4.4 Lexicon Analysis

This section discusses the actual data preparation processes carried out to create the probabilities of words for the tagger. The two basic probabilities for the tagger are lexical and transition probabilities.

The main lexicon words are stored in a hash table, where the slots in the table are the word-structures themselves, not pointers to word-structures. This design saves one indirection for every look-up, and it saves one pointer for each word. Moreover, the size of the table is the same as the number of words, which means no memory loss.

4.4.1 The lexicon

The following lexicon is prepared from which the two probabilities are developed for the analysis of the data set.

	NN…	PP…	VV…	JJ…	AD…	Total
mana	2	0	0	0	0	2
meeshaa	1	0	0	1	0	2
nama	2	0	0	1	0	3
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
ofii	0	4	0	0	0	4
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
tokko	1	0	0	2	0	4
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
yaadadhe	0	0	2	0	0	2
yeroo	0	0	0	0	9	9
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Total	334	100	351	226	81	1621

Table 4.1: Sample of Lexicon

The lexicon in Table 4.1 is parts of the entire corpus from which the lexical and transitional probabilities are derived. The rows show the count of each token with each category. And the column shows the count of each category with each token in the corpus identified for the study.

As indicated in the Table, for example, the count of the word *nama 'man'* is two as a noun and one as an adjective; the number of the word *yeroo 'time'* is nine as adverb and so on in the corpus.

The total identified categories for the study are seventeen categories. Among these categories, Table 4.2 shows some of the count of the categories in the corpus.

Category	Count of category
Noun	334
Pronoun	100
Verbs	226
. . .	. . .
Total	1621

Table 4.2: Sample Count of Categories in the Corpus

4.4.2 The Lexicon Probabilities

The lexical probabilities have been estimated by computing the relative frequencies of every word per category from the training annotated corpus. All statistical information, that enables to develop probabilities, are derived automatically from a hand annotated corpus (the lexicon).

The statistical information that is derived from the lexicon is:

- Cn – The number of word tokens.
- C(w,t) – Occurrences of word w tagged with tag t.
- C(t) – Occurrences of the tag t.
- C(t1,t2) – Occurrences of the tag bigram t1,t2.

The contextual information from sequences of words is obtained from immediately preceded

category using bigram model.

In the lexicon probabilities, each word (w_i) occurrences tagged with tag (t_i) is counted and divided by the counted number of occurrence of the tag (t_i).

In the Table 4.3 below, for instance, the lexical probability of the word *Oromoon* tagged with *NN* is calculated as:

C(Oromoon, NN) = 7

C(NN) = 334

So, P(Oromoon\NN) = C(Oromoon, NN)/C(NN)

= 7/334

= 0.0206

Where, C and P are count of and Probability, respectively.

Table 4.3 shows sample lexical probabilities' of words in the lexical probabilities hash table, named **LexProb**. The calculation is performed based on Formula 5 that was discussed in chapter two, section 2.2.2.3. In LexProb hash table, the keywords are the tokens tagged with their appropriate code of categories and their results are the probabilities of the tokens given categories.

Word with given lexical category	Probability
P(Oromoon\NN)	0.0206
P(jedhaman\VV)	0.0052
P(kabajaa\AD)	0.02174
P(ayyaanichaafi\NC)	0.11111
P(amma\AD)	0.04348
P(yeroo\AD)	0.10869

Table 4.3: Sample Lexical probabilities

4.4.3 The Transitional Probabilities

In transitional probabilities, the information of one part-of-speech category preceded by other categories is developed from training lexicon corpus. For this study, bigram is used. Bigram considers the information of the category (t_{-1}) preceded the target category (t). That means, $P(t\backslash t_{-1})$, where t is – part-of-speech category.

As similar to lexical probabilities, transitional probabilities are computed using Formula 4 discussed in chapter, section 2.2.2.3 as follows:

$P(t\backslash t\text{-}1) = C(t\text{-}1, t)/C(t)$

For exemple, C($S) = 157

C(NN,$S) = 79

P(NN\$S) = C(NN, $S)/C($S)

= 79/157

= 0.5032

Bigram category	Probability
P(NN\$S)	0.5032
P(VV\$S)	0.0063
P(NN\VV)	0.1538
P(NN\PN)	0.0063
P(JJ\NN)	0.2695
P(JJ\$S)	0.1465
P(PP\NN)	0.1018

Table 4.4: Sample Transitional probability

As it is indicated in Table 4.4, the code $S indicates occurrence of the words at the beginning of the sentence. It is observed that noun and verb words occurrence at the beginning of the sentence is 0.5032 and 0.0063 respectively. This shows that in all training sample corpus noun

words register the highest probability of occurrence, while verbs register the least probability of occurrence at the beginning of the sentences. This confirms the structure of sentences construction in Afaan Oromo is Subject – Object (or/and complement) – verb as it is discussed in chapter three in detail.

In TransProb hash table, the key words are assigned by two joined categories. The first two character symbols represents the target category (t) and the second two character symbols represents the category immediate precedes the target category (t_{-1}).

4.4 Part-of-speech Algorithms

In this study, preprocessing algorithms including sentence splitter and word tokenizer, and Viterbi algorithms are used.

4.5.1 The Sentence Splitter

The manually tagged corpus is broken down into sentences to get the contextual information of words in sentences. Specifically, this is required to develop one of the basic probabilities for the tagger, transitional probabilities, which are developed from bigram categories ($t_i \backslash t_{i-1}$). Figure 4.2, shows the algorithm that segments the corpus in to sentences.

```
Given a corpus
char last = '.' or '?' or '!'
while(corpus !=null){
String string
while(string.length()-1 ! = last){
        concatenate   strings}
break the string}
```

Figure 4.2: Sentence Splitter Algorithm

4.5.2 The Tokenizer

The tokenizer class takes the output of the SentenceSplitter and break in to tokens. Tokens are every words and punctuation marks in the corpus. This enables to treat tokens individually and to prepare statistical information of each token in the corpus to process probabilities of tokens and categories.

The tokenizer makes ready the corpus for two analyzer classes that build knowledge bases for the tagger. The first class, LexProbAnalyzer, builds lexical probabilities and the second class, TransProbAnalyzer, in other side builds transitional probabilities from tokenized corpus.

Given line of strings, the algorithm tokenize as follows:

```
Given sentences
While(sentences !=null)
String PUNCT ="!\"$%^&*()_+=#{}[];:'‘/?,. \t\n";
while (string ! = PUNCT) do{
   concatenate characters in the string
}
break the string
}
```

Figure 4.3: Tokenizer Algorithm

4.5.3 The Tagger

The tagger learns from the two probabilities to label appropriate tag to each word in sentences. The tagger for the study is developed from Viterbi algorithm of hidden Markov model. Viterbi algorithm performs tagging processes in three steps [2]:

- the initialization step
- the iteration step and
- the sequence step

The problem is given word sequence w_1, . . ., w_T, lexical categories L_1, . . ., L_N, lexical probabilities PROB(w_t\$L_i$), and bigram probabilities PROB(L_i\$L_j$), find the most likely sequence of lexical categories C_1, . . ., C_T for the word sequence.

4.5.3.1 Initialization Step

The initialization step initializes array variables, SEQSCORE and BACKPTR, as shown in Figure 4.4. SEQSCORE array variable is the variable that temporary holds probabilities of

words in a sentence that is going to be tagged. It is initialized with the product of probabilities of categories at the beginning of a sentence (Li\$) and the beginning of word's probability tagged with given categories ($w_1\backslash L_i$) in the sentence. BACKPTR is used to hold the index of the highest probability of category for a given word.

```
for i = 1 to N do
    SEQSCORE(i,1) = PROB(w1\Li) * PROB(Li\$)
    BACKPTR(i,1) = 0
```

Figure 4.4: Initialization Step in Viterbi Algorithm

4.5.3.2 Iteration Step

In iteration step, for all the rest of words in the sentence, it looks up lexical probabilities of possible tags for the word. This is then combined with the contextual probability for each tag to occur in a sequence preceded by the one previous tag. The tag with the highest combined score is selected and its index is stored in BACKPTR as shown in Figure 4.5. This iteration is continued for all sequence of words in a sentence.

```
for t = 2 to T
  For i =1 to N
    SEQSCORE (i,t) MAXj=1,N(SEQSCORE(j,t-1) * PROB(Li\Lj))*
PROB(wt\Li)
     BACKPTR(i,t)  = index of j that gave the max above
```

Figure 4.5: Iteration Step in Viterbi Algorithm

4.5.3.3 Sequence Identification Step

Sequence Identification Step finally does the process of tagging for each word depending on the information of BACKPTR variable as indicated in Figure 4.6. It processes through iterating the BACKPTR that holds the pointer of appropriate category for each word in the sentence.

```
C(T) = i that maximizes SEQSCORE(i,T)
for i = T-1 to 1 do
    C(i) = BACKPTR(C(i+1), i+1)
```

Figure 4.6: Sequence Identification Step in Viterbi Algorithm

4.5 Performance Analysis of the Tagger

In order to test the performance of the tagger the following experiments are carried out. The first experiment is done to validate the tagger how much it performs on portion of the training set. Out of the total data set 20% is used for testing and 100% (including the test set) is used for training as discussed in chapter one, section 1.4.3.

The second experiment is done to validate the actual performance of the tagger with test set that is not included in training set. In this experiment ten fold cross validation is used. Out of total data set, 90% is for training and the remaining 10% is used for testing as discussed in chapter one, section 1.4.3.

4.6.1. Performance Analysis with Portion of Training Set

Table 4.5 indicates the performance of the tagger which is tested on 20% of the total training set. The purpose of using this validation method is to evaluate the human errors while tagging the sample corpus manually besides validating the tagger. Accordingly, after correcting human and spelling errors, the performance of the tagger shows 282 words (96.58%) correctly tagged words.

Correctly tagged words	**Incorrectly tagged words**	**Accuracy in percent**
282	10	96.58%

Table 4.5: Validating the Tagger with 20% Test Set

Incorrectly tagged words are analyzed through manually tagged corpus. Most of them are tagged incorrectly from nouns to adjectives and vise versa. For instance, the word *biyya 'country'*, *Afaan 'language'* and *yeeyyii 'wolf'* are wrongly tagged as adjectives. As it is discussed in chapter two, categories of words in sentences characterize their position. Specially, some noun words take the position of adjectives and vise versa in different context of sentences.

The other errors are due to human errors while preparing the manually tagged corpus. For instance, the word *taanaan 'if so'* is with two words that are not separate in Afaan Oromo and given the tag code CC for the study, but it is wrongly assigned with auxiliary verb code (AX) and the word *waanti 'anything'* is wrongly tagged with preposition instead of pronoun. Such

kinds of problems are handled through reexamining the corpus and correcting wrongly tagged words according to their positional function in sentences.

The other problem that causes for wrongly tag assignment is spelling error. For the same word there are different spellings in training set and test set. These problems can be solved by developing standardized corpus for the language.

4.6.2 Performance Analysis with Separate Test Set

In this performance analysis, the tagger is repeatedly trained and tested following ten fold cross validation as shown in the following Table.

test sets	**Correctly tagged Words**	**Incorrectly tagged Words**	**Accuracy in percent**
1	99	16	86.07%
2	115	6	95.04%
3	119	8	93.70%
4	110	10	91.67%
5	114	10	91.93%
6	120	8	93.75%
7	126	10	92.65%
8	119	8	93.70%
9	126	12	91.30%
10	115	13	89.84%
Average accuracy			**91.97%**

a

Correctly tagged Words	**Incorrectly tagged Words**	**Accuracy in percent**
93	22	80.87%
106	15	87.60%
110	17	86.61%
108	12	90%
106	18	85.48%
115	13	89.84%
125	11	91.91%
113	14	88.98%
119	19	86.23%
113	15	88.28%
		87.58%

b

Table 4.6: The Accuracy of the Bigram(a) and Unigram(b) Model Tested by Test Sets

The algorithms of the tagger are tested with a corpus of 146 Afaan Oromo words in average in each test set and that is trained on the training set of 1315 words, and the result of each test are compared with a copy of the test set that is hand annotated. As a result, the results of the experiments for both bigram and unigram algorithms show an accuracy of 91.97% and 87.5% correctly tagged words in average respectively.

With this corpus, the distributions of accuracy performance in both models are not as far from each other. The maximum variation in the distribution of bigram and unigram models is 8.97 and 11.04 respectively. If the corpus is standardized, this variation will reduce since standardized corpus consist relatively complete representative of words for the language and fair distribution of words in training set and test are observed.

In bigram model, the statistical accuracy is performed more than unigram model. Bigram model uses probability of contextual information besides the highest probability of categories given a word in a sentence to tag the word. The difference accuracy rate from bigram to unigram is 4.39% with this dataset. This indicates, contextual information (the position in which the word appear in sentence) affects the determination of word categories for Afaan Oromo.

4.6 Conclusion

Even if the study is the groundwork in part-of-speech tagging for Afaan Oromo language, it is believed that the proposed framework is a promising approach to deal with the tagging

problem for the language. With this corpus it shows statistical significance. In Table 4.6, the accuracy of unigram and bigram algorithms indicates 87.58%.and 91.975, respectively So, that the result indicates us the bigram model of Viterbi algorithm is the promising approach in part-of-speech tagging for Afaan Oromo language.

Since the knowledge bases are automatically constructed, there is no human error in preparing the statistical information from which the tagger learns to identify appropriate tag for words in sentences.

Since the sparse data, many words in the test set are not found in the training corpus, and this is not handled by the tagger. These words, as a matter of fact, tagged with the tag UN, which means that the tagger was not able to find any possible tags for that word. Using standardized and large amount of corpus and the combination of stochastic and rule based can solve the problem.

Chapter Five

5. Conclusions and Recommendations

5.1 Conclusions

The study is used as groundwork for parts of tagger development for Afaan Oromo language. Since there is no corpus and tagsets for natural language processing of the language, sample corpus and tagset are designed and developed for the study.

Word sense disambiguation experiments using lexical and syntactic features are conducted. Unigram and bigram models of Viterbi algorithms and 1462 annotated Afaan Oromo words excluding punctuation marks are used to implement the tagger. Both unigram model and bigram model show reasonably good accuracies and that the part-of-speech of the word immediately following the target word (bigram) is particularly useful in disambiguation as compared to individual part-of-speech features (unigram). Their difference indicates a significant variation (4.39%).

Both models do not try to tag unknown words in any way; it just tags them as UN, unknown. Incorporating of rule based approach handles the problems of this problem since it enables to guess unknown words based on their suffixes besides their surrounding words. The study also does not include number, gender and person of further subcategories of main categories in the language.

Despite the limitations of the taggers presented in this study, the results presented herein can function as a first benchmark for future research on part-of-speech tagging of Afaan Oromo language. Furthermore, the inclusion of other model of Viterbi algorithm such as trigram model and other approach includes rule based may improve the performance of the system.

5.2 Recommendations

As mentioned in section 5.1, this work is the groundwork in part-of-speech tagger for Afaan Oromo language. Obviously, there are further works that should be considered by future researchers in the area of natural language processing in general and part-of-speech tagging in particular.

- The accuracy and effective processing of natural language processing that need annotated data sets are depend up on standardized and sufficient amount of corpus. Lack of standardized annotated corpus is one of the limitations in the language. So, the responsible agents should take in to consideration to prepare it. The standard corpus is not only for Afaan Oromo but also should be prepared for other local languages as well.

- This study performs with baseline (unigram) and first order of Markov Model (bigram) that takes two sequences of words in a sentence in to consideration. Therefore, researchers and experts can repeat the same study with the same method and approach with huge corpus and in second order Markov Model (trigram) that considers three sequences of words in a sentence.

- There are also other approaches that are different from Hidden Markov Model which can be applied for similar problem. So, experts and researchers can develop the same title for the language in rule based and neural networks to compare the result with the performance of this study.

- Currently, many part-of-speech tagger have done in combination of both hidden Markov Model and Rule based approach to get more accurate result and to handle rare words and unknown words. So, to make full fledged the tagger, experts and researchers can do in combination of Hidden Markov Model and Rule based tagging approach.

- The same approach and method can be repeated for developing parts-of-speech tagger for other local languages, like Wolaita, Tigrigna, etc.

- This study does not consider genders and numbers of word categories in the implementation. So researchers and experts can work by including these further sub-categorization of word classes.

Reference

[1] Hermann Helbig. Knowledge representation and the semantics of natural language. Springer-Verlg Berlin Heidelberg, Germany, 2006

[2] James Allen. Natural language Understanding. The Benjamin/Cummings Publishing company, Redwood City, Canada, 1995

[3] Gobinda G. Chowdhury. Natural Language Processing: Department of Computer and Information Sciences, University of Strathclyde, Glasgow G1 1XH, UK, http://www.cis.strath.ac.uk/cis/research/publications/ papers/strathcis_publication_320.pdf

[4] Ferran Pla and Antonio Molina. Natural Language Engineering: Improving part of-speech tagging using lexicalized HMMs_ 2004. Cambridge University Press, United Kingdom, 2004

[5] Mesfin Getachew. Automatic part-of-speech tagging for Amharic language an experiment using stochastic Hidden Markov Approach. MSc. Thesis. School of Graduate Studies, Addis Ababa University, 2001.

[6] Abara Nefa. Long Vowels in Afaan Oromo: A Generative Approach. M.A. Thesis. School of Graduate Studies, Addis Ababa University, 1988.

[7] Kula K. T., et al. Evaluation of Oromo-English Cross-Language Information Retrieval. In IJCAI 2007 Workshop on CLIA, Hyderabad (India), 2007.

[8] Morka Mekonnen. Text to speech system for Afaan Oromo. MSc Thesis. School of Graduate Studies, Addis Ababa University, 2001.

[9] Diriba Magarsa. An automatic sentence parser for Oromo language. MSc Thesis. School of Graduate Studies, Addis Ababa University, 2000.

[10] Asseffa W/Mariam. Developing morphological analysis for Afaan Oromo text. MSc Thesis. School of Graduate Studies, Addis Ababa University, 2000.

[11] Yenewondim Biadgo. Application of multilayer perception neural network for tagging part-of-speech for Amharic language. MSc Thesis. School of Graduate Studies, Addis Ababa University, 2005.

[13] Chris Biemann. Obtaining semantic relations from text corpora in an unsupervised way, MULTILINGUA Fellowship, University of Leipzig, Germany, 2005

[14] Baye Yimam. The pharse structure of Ethiopian Oromo. University of London, School of Oriental studies, Ph. D. Dissertation, London, 1986.

[16] Mengesha Rikitu. How to read Oromiffa and use its grammar. Magic press, London, 1993.

[17] Noah Webster. Webster's new twentieth century dictionary of English language. 2nd edition, New York, USA, 1979.

[18] Christopher D. Manning and Hinrich schutze. Foundations of Statistical Natural Language Processing. The MIT Press Cambridge, Massachusetts London, England, 1999

[19] Gumii Qormaata Afaan Oromo. Caasluga Afaan Oromo. Komoshinii Aadaafi Tuurizimii Oromiyaa, 1996

[20] Wakshum Mekonen. Development of a stemming Algorithm for Afaan Oromo text, MSc Thesis School of Graduate Studies, Addis Ababa University, 2000

[21] Sandipan Dandapat, et al. A Hybrid Model for Part-f-Speech Tagging and its Application to Bengali, 2004

[22] Pierre M. Nugues. An Introduction to Language Processing with Perl and Prolog. Springer-Verlag Berlin Heidelberg, Germany, 2006

[23] Daniel Jurafsky and James H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice-Hall, Inc., 2000.

[24] Sandipan Dand et al. Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario. Department of Computer Science and Engineering Indian Institute of Technology Kharagpur, 2007

[25] Frank Van Eynde. Part-of-speech Tagging and Lemmatisation for the Spoken Dutch Corpus, Center for Computational Linguistics Maria-Theresiastraat 21 3000 Leuven, Belgium, 2000

[26] Roger Garside and Nicholas Smith. A Hybrid Grammatical Tagger: CLAWS4, http://ucrel.lancs.ac.uk/papers/HybridTaggerGS97.pdf

[27] Gerard Despate. Introduction to Language Lingustics. Unpulished, 1972

[28] Catherine Griefnow Mewis. A Grammatical sketch of written Oromo. Germeny, 2001

[29] Eric Brill. A Simple Rule-Based Part-of-speech Tagger. Department of Computer Science University of Pennsylvania Philadelphia, Pennsylvania 19104, 1992

[30] Christer Samuelsson Atro Voutilainen. Comparing a Linguistic and a Stochastic

Tagger. Lucent Technologies Bell Laboratories .Murray Hill, NJ 07974, USA, 1988.

[31] Tomoyoshi Matsukawa , Scott Miller etal. Example-Based Correction of Word Segmentation and Part-of-speech Labeling. Tomoyoshi Matsukawa , Scott Miller, and Ralph Weischedel BBN Systems and Technologies Cambridge, MA 02138, England. http://www.aclweb.org/anthology-new/H/H93/H93-1045.pdf

[32] Steven Bird et al. Introduction to Natural Language Processing. 2007 http://nltk.org

[33] Shereen Khoja. APT: Arabic Part-of-speech Tagger. Computing Department, Lancaster University Lancaster LA1 4YR, UK. http://archimedes.fas.harvard.edu/mdh/arabic/NAACL.pdf

[34] Chao-Huang Chang and Cheng-Der Chen. HMM-based Part-of-Speech Tagging for Chinese Corpora. E000/CCL, Building 11, Industrial Technology Research Institute Chutung, Hsinchu 31015, Taiwan, R.O.C. http://www.aclweb.org/anthology-new/W/W93/W93-0305.pdf

[35] Jana Diesner. Part-of-speech Tagging for English Text Data. School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213

[36] Sisay Fissaha. Part-of-speech tagging for Amharic using Conditional Random Fields. Informatics Institute, University of Amsterdam Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, 2005

[37] Sandipan Dandapat, et al. Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario. Department of Computer Science and Engineering Indian Institute of Technology Kharagpur India 721302. Proceedings of the ACL 2007 Demo and Poster Sessions,

Prague, 2007

[38] R. J. Hayward and Mohammed Hassan . The Oromo Orthography of Shaykh Bakri Saṗalō Bulletin of the School of Oriental and African Studies, University of London, 1981. http://en.wikipedia.org/wiki/Oromo_language.

[39] Jinxi Xu, et al. The design and Implementation of a part-of-speech tagger for English, 1994

[40] Simon STÅHL. Part-of-Speech Tagger for Swedish, Computer Science, Lund University, 2000

[41] Census report: Ethiopia's population now 76 million. December 4th, 2008. http://ethiopolitics.com/news

[42] Yoshua Bengio and Yves Grandvalet. No Unbiased Estimator of the Variance of K-Fold Cross-Validation. Dept. IRO, Universit´e de Montr´eal C.P. 6128, Montreal, Qc, H3C 3J7, Canada, 2004

http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html

[43] A S Hornby. Oxford Advanced Learner's Dictionary. Sixth edition, Oxford University press,2000

Appendix

Sample manually annotated corpus:

GaheeJJ dubartootiNN baadiyyaaJJ wabiiJJ soorataaNN mirkaneesuufVV qabataafiJC urteessaaJJ ta'eAX cimsuudhaafVV qaamoleenNN dhimmiJJ ilaalatuVV xiyeeffataniiAD hojjechuuVV akkaPR qabanAX ibsameVV .PN KunuunsiJJ qabeenyaNN uumamaafiJC eegumsiJJ naannawaaNN wabiiJJ midhaanNN nyaataaJJ mirkaneessuufVV shooraNN olaanaaJJ akkaPR gumaachuVV ittigaafatamaanJJ abbaaNN TaayitaaNN eegumsaJJ NaannawaaNN ibsameVV .PN KunPP kakuuVV OromoonNN qabudhaAX .PN GuyyaanNN kunPP sadarkaaJJ adduyaattisNC ta'eeAX sadarkaaJJ biyyaaNN keenyaattiJJ yerooAD jalqabaatiifAD kabajameeraVV .PN kanPR boriitiifAD hinAX yaadinaaVV kanPR har'aAD ta'uAX hinAX beekamuutiAX .PN MirgaNN jireenyaafiNC guddinaJJ daa'immaniifNP .PN IjjiNN utooCC ilaaltuuVV axxiffachuunVV hinAX danda'amuVV .PN AsheetaafiNC bareedaNN biraPR hinAX darbanVV .PN QorannaafiNC qo'annaaNN baraAD baraAD dheeraanJJ boodadhaPR qaroominniNN har'aaNN kanPR argameVV .PN BakkiPR SeerriNN hinAX jireettiVV waantiPP kabajamuJJ hinAX jiruAX .PN tarsiimoofiNC teekinkootaNN jedhamanVV ittiPP dabaladhuVV .PN BoftiNN kanPR suunfatuVV arrabaNN isaaniPP .PN TisiisaNN gammoojjiiJJ dhukkubaNN beeyldootattiJJ daddbarsituVV dhabamsiisuufisVV keemikaalliNN farraJJ tisiisaaNN biifameeraVV .PN MeePR dubbiiNN kanaPP xiqqoJJ qabatamaaVV gooneeVV haaPR ilaalluuVV .PN KanPR waanPR ofiiPP kabajuVV ofifillePP kabajaJJ argataVV .PN FayyaanNN waanPR hundaJJ caalaJJ .PN QananiisaanNN dorgommiiVV eegalePR walPR irraaPR hinPR kutuVV .PN

Sample test sets

Yoo wal hin lolan waraana of harkaa qabu taanaan ofirraa garagalchanii wal rukutu malee ittiin wal waraanuun safuu
Bifti qonnaa kanaa boodaa akka geedaramuuf jiru bu'awwan qorannoo saayinsii addeessu
Waajirichi dhaabbilee 23 keessatti jijjiirama hojii qoratee hojjeessuuf sochiirra akka jiru beeksisaniiru
Bishaan jireenyaafi waan barbaachisaa guddaa akka ta'e beekamaadha
Waggaa 12 booda dubartii jalqabaa sanyii gurraachota kessa pirezidaantii yuunvarstii taatee hojjate
kan duulee hin beekne hidhataa bula
Waanni cimaan ammoo ciminasaatiin yoo itti fufe kan maqaan isaa tolee mul'atu
Amma gara maaraguutti ce'uu keetii of jabeessi
Akka walii galaatti malaammaltummaan qaama kennuufi fudhatu gidduutti waan raawwatuuf dhiibbaa inni biyya irraan gahu hubannee dhabamsiisuuf motummaa
Bofti baayyee soch'u Tisiisa jiran keessaafi alaa farra tisiisaa ti
Fayyaan aadde Faantuu ammamuu matatii mul'ate meeshaa dhiigaa hoteela Giyoonitti gorsa bal'aa kenna
Tisiisa qammooJJii ilalluu dura darbuu kanaan barbaachisaadha

Sample lexical probabilities in LexProb hash table:

Sample transitional probabilities in TransProb hash table:

Declaration

This thesis is my original work and has not been submitted as a partial requirement for a degree in any university

Getachew Mamo
January, 2009

The thesis has been submitted for examination with my approval as university advisor.

Million Meshesha (PhD)