
ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

# IMPROVING BRILL'S TAGGER LEXICAL AND TRANSFORMATION RULE FOR AFAAN OROMO LANGUAGE

ABRAHAM GIZAW AYANA

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

February, 2013

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

# IMPROVING BRILL'S TAGGER LEXICAL AND TRANSFORMATION RULE FOR AFAAN OROMO LANGUAGE

ABRAHAM GIZAW AYANA

**Signature of the Board of Examiners for Approval**

| Name | Signature |
|---|---|
| 1. Dr. Sebsibe Hailemariam, Advisor | _____ |
| 2. _____ | _____ |
| 3. _____ | _____ |

## Dedication

This thesis is dedicated to My Mother Ilfinesh Wirtu as well whose prominent support I will never forget in my life with which I was provided prior and throughout my course of study.

## Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to the almighty God. All of my efforts would have gone for naught if it had not been for his importunate help. Then I offer my sincerest thanks to my supervisor, Dr Sebsibe H/Mariam, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Master degree to his encouragement and effort, without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

I would also like to give special thanks to my family. My family members are always there to support me in every situation. On top all, three persons, my lovely mom, eldest brother, and my sister (Aynalem Gizaw) deserve very grateful thanks because without them I would not be who I am today.

Though it is difficult to mention the name of persons who gave me their hand while doing this thesis, it is necessary to mention those who gave me their precious time to read the thesis document, to share ideas, and gave me moral and material support. Dr. Fikadu Gemechu, Ato Solomon Shiferaw, Ato Tebaje Tadese, Ato Desalegn Nikola, Ato Mesay Yohannes, Ato Ehab Umer, and Ato Addisu Bole are few of them. I am very grateful to thank them for what they did.

Last but not least, I would like to sincerely thank all of my friends, colleagues, and classmates for their assistance, encouragement, and inspiration during this research.

# Table of Contents

# List of Figures

# List of Tables

## Acronyms and Abbreviations

| | |
|---|---|
| ANN | Artificial Neural Network |
| HMM | Hidden Markov Model |
| NLP | Natural Language Processing |
| AI | Artificial Intelligence |
| NLTK | Natural Language Toolkit |
| POS | Part of Speech |
| TEL | Transformational Error Driven Learning |

# Abstract

Natural Language Processing (NLP) refers to Human-like language processing which reveals that it is a discipline within the field of Artificial Intelligence (AI). However, the ultimate goal of research on Natural Language Processing is to parse and understand language, which is not fully achieved yet. For this reason, much research in NLP has focused on intermediate tasks that make sense of some of the structure inherent in language without requiring complete understanding. One such task is part-of-speech tagging, or simply tagging. Lack of standard part of speech tagger for Afaan Oromo will be the main obstacle for researchers in the area of machine translation, spell checkers, dictionary compilation and automatic sentence parsing and constructions.

Even though several works have been done on POS tagging for Afaan Oromo, the performance of the tagger is not sufficiently improved yet. Hence, this thesis has developed Afaan Oromo POS tagger to improve Brill's tagger lexical and transformation rule with sufficiently large training corpus. Accordingly, Afaan Oromo literatures on grammar and morphology are reviewed to understand nature of the language and also to identify possible tagsets. As a result, 26 broad tagsets were identified and 17,473 words from around 1100 sentences containing 6750 distinct words were tagged for training and testing purpose. From which 258 sentences are taken from the previous work. Transformation-based Error driven learning are adapted for Afaan Oromo part of speech tagging. Different experiments are conducted for the rule based approach taking 20% of the whole data for testing. A comparison with the previously adapted Brill's Tagger is made. The previously adapted Brill's Tagger shows an accuracy of 89.8% whereas the improved Brill's Tagger result shows an accuracy of 95.6% which has an improvement of 5.8%.

Hence, it is found that the size of the training corpus, the rule generating system in the lexical rule learner, and moreover, using Afaan Oromo HMM tagger as initial state tagger have a significant effect on the improvement of the tagger. Since there is only a few readymade standard corpuses, the manual tagging process to prepare corpus for this work was challenging and hence, it is recommended that a standard corpus is prepared.



**Keywords:** Afaan Oromo, POS tagger, NLP, Brill's Tagger

# CHAPTER ONE

# INTRODUCTION

## 1.1. Background

Computational linguistics is the study of Natural Languages by means of computational perspectives. It is meant to be able to identify an appropriate models that approach human performance in linguistic tasks. This can be achieved through various natural language processing methods for speech or text processing. Natural Language Processing (NLP) refers to Human-like language processing which is a discipline within the field of Artificial Intelligence (AI) [1].

Natural Language processing is one of the current hot research areas for scientists and academic researchers. The goal is to parse and understand natural language, which is not fully achieved yet. For this reason, much research in NLP has focused on preprocess and intermediate tasks that make sense of some of the structures inherent in language without requiring complete understanding. One such task is part-of-speech tagging, or simply tagging.

In sentences, all words can be labeled with their Part-of-Speech tag. These tags denote the grammatical function of the word in the sentence. Some simple, but well-known part of speech tags are for instance nouns, verbs, adjectives, adverbs and determiners. Part-of-Speech tagging makes sentences easier to parse by a computer, and is therefore a preprocessing step frequently used in text-processing systems [2]. Over the years there has been a lot of research to automate Part-of-Speech tagging, where a computer program tries to label each word with the correct Part-of-Speech tag.

Different methods have been used so far for POS tagging, such as Transformation-based learning, statistical learning using Hidden Markov models, statistical learning using Maximum Entropy models, Neural Networks, Support Vector Machines.

In this study, Brill's rule-based part-of-speech tagger is tested and adapted for Afaan Oromo Language. An algorithm of the original Brill's tagger is modified considering the nature of the language understudy. We have used this tool for part-of-speech tagging of Afaan Oromo words with the help of thesaurus and large training corpus.

## 1.2. Statement of the Problem

Very limited works have been done in the past in the areas of Computational linguistics in relation to African indigenous languages including major Ethiopian languages like Afaan Oromo.

Since Afaan Oromo is morphologically very productive, derivations and word formations in the language involve a number of different linguistic features including affixation, reduplication and compounding [6]. Obviously, these high inflectional forms and extensive derivational features of the language are presenting various challenges for text processing and information retrieval experiments in the language. In information retrieval, the abundance of different word forms and lexical variability may result in a greater likelihood of mismatch between the forms of a keyword in a query and its variant forms found in the document index database(s). In the context of machine translation, this may lead to a serious mismatch problem between query terms and citation forms of vocabulary entries found in the bilingual dictionaries that are commonly used for cross language information retrieval.

On the other hand, the amount of accessible electronic information has exploded in recent years thanks to the Internet and other related distributed international networks. Due to rapidly expanding use of the Internet for communication and dissemination of information throughout the world, electronic sources are now available in an ever-increasing number of languages. Users of such globally distributed networks (including digital libraries and World Wide Web) need to be able to access and retrieve any relevant information in whatever language and form it may have been recorded and stored [7]. Therefore, the need for automated natural language processing is increasingly important.

Lexical attributes, like syntactic (part-of-speech) and semantic attributes are in most cases, ambiguous in every language. Automatic resolution of ambiguity of these attributes can be achieved using different techniques; rule-based, statistical, Artificial Neural Network and their hybrids are some of them. Moreover, one linguistic feature is more influential in resolving ambiguity over the other feature. For example, knowledge of syntactic category can assist in smooth disambiguation of semantic category and vice versa [1]. Properly disambiguated syntactic and semantic properties of lexicon may significantly help us in word sense

disambiguation, text analysis, information retrieval, natural language understanding and speech processing etc.

As far as the researcher's knowledge is concerned, there is no readymade standard part of speech tagger for Afaan Oromo language. This exists as the main difficulty for researchers in the area of speech processing, spell checkers, dictionary compilation and automatic sentence parsing and constructions. All of these NLP research directions demand part of speech tagger during their preprocessing phase to attain the best performance [8].

Even though several works have been done in POS tagging for Afaan Oromo, the performance of the tagger has not sufficiently improved yet. The work in [9] is the first attempt to use a transformation based Error-Driven Learning (TEL) for Afaan Oromo POS tagger. The researcher recommended future work on improving the lexical and transformational rule to improve the performance of the POS tagger. Besides, the researcher found out that adding more training dataset can improve the performance of the tagger since the experiment was carried out on small scale dataset. Hence, the aim of this thesis is to improve Brill's tagger lexical and transformation rule for Afaan Oromo POS tagging with sufficiently large training corpus.

## 1.3. Objectives

### 1.3.1. General Objective

The general objective of the research is to enhance Brill's Tagger lexical and transformational rule for Afaan Oromo Language.

### 1.3.2. Specific Objectives

The specific objectives of the study are:

- To review related work and collect training dataset prepared for the same purpose.
- To adapt the Brill tagger Lexical rule
- To adapt the Brill tagger transformation rule
- To prepare more training dataset from untagged Afaan Oromo corpus
- To model TEL based POS for Afaan Oromo
- To develop prototype TEL based POS for Afaan Oromo language
- To test and analyze the performance of the model built

## 1.4. Methodology

### 1.4.1. Data Collection

This work is based on the finding of the previous Afaan Oromo POS tagging work [9], which found out that adding more training dataset can improve the performance of the tagger. Therefore, in this research, Afaan Oromo standard corpus with 17,473 words from around 1100 sentences containing 6750 distinct words were tagged for training and testing purpose. From which 258 sentences are taken from the previous work. About 26 broad tagsets were identified for tagging the corpus.

The Afaan Oromo balanced text corpus is collected randomly from different sources in a form of both hardcopy and softcopy. Those sources are considered to be under different domain or categories such as Afaan Oromo books, journals, publications, news, newspapers and previous research corpus. Accordingly, TV Oromia, Voice of America (Afaan Oromo service), Afaan Oromo FM radio, websites like www.oromiyaa.com (website of oromia regional state), www.gadaa.com ,www.qalbesa.wordpress.com, online journals and publications, books like Seenaa Oromo Jarraa 16ffaa, Yaadanii, Hawii, newspaper like Bariisa, Kallacha and previous Afaan Oromo POS tagging research corpus from the work of [8] and [9] are some of the main data source.

An incremental approach is used to prepare the tagged corpus. First, we took the 258 previously tagged Afaan Oromo corpus for training the Brill tagger. Then, this trained tagger takes untagged text as an input and tags the words based on the knowledge that it has acquired during the training and gives tagged text as an output. The output of the tagger is taken and given to the language professionals for correction and approval. After the corrected and approved tagged text is obtained, the corpus is updated which is used in turn for training of the final POS tagger model. This process is repeated until adding the corpus can have insignificant effect on the performance of the tagger.

A learning curve is used to analyse the effect of the size of the training corpus on the performance of the tagger. First, the tagger is trained on the 10% of the training corpus, which results in a small performance. Then, we added another 10% of the training corpus and saw a little increment on the tagger performance. The process continues until increasing the size of the

training corpus does not show significant improvement on the tagger performance. Therefore, the total size of the training corpus used for this research is said to be sufficiently large Afaan Oromo balanced corpus for improving the Brill's tagger performance.

### 1.4.2. Modeling

In this thesis work, Brill's Transformational Error Driven Models were experimented. Brill Transformation error driven learning approach is adapted for Afaan Oromo POS tagger in which rules are automatically learned from a manually annotated corpus. The rules are the important elements for annotating words in the TEL approach. TEL can use different types of initial state annotators. Default tagger is used for the original Brill's tagger. In this work, different initial state tagger is considered to be used. As a result HMM tagger is preferred to be an initial state tagger for the adapted TEL model. The HMM based tagger relies on the statistical property of words along with their part of speech categories. Such a statistical property can be distributional probability of words with tags which can be obtained during the training phase of the system. Both models, the rule based and HMM based taggers, have their own pros and cons. Improving Brill's tagger lexical and transformational rule is proposed for this thesis. A TEL that uses HMM tagger model as initial state tagger is exploited in this work.

### 1.4.3. Testing and Validation

A 10-fold cross validation method is used for training and testing of the system as it provides a better and reasonably acceptable result.

## 1.5. Tools and Techniques

In this study, we use several tools and techniques in order to achieve the desired goals. To customize Brill's tagging system for Afaan Oromo, Brill rule based tagger and NLTK are used. The tool incorporates the Original Brill Tagger, which is used for training and testing purpose. Python and Perl Programming languages are also used for coding as required.

## 1.6. Application of Results

The main goal of the study is to improve the Brill's tagger and transformation rules for Afaan Oromo. The study consists of the experimental work in automating NLP tasks. Hence, the result of this study will be useful for Afaan Oromo language processing tasks such as speech

recognition, information retrieval, information extraction, machine translation and others during their preprocessing phase. Moreover, the contribution of the result of this study in linguistic and computational linguistic work will make it very important.

## 1.7. Organization of the Thesis

The whole thesis is organized into Seven Chapters including the current one. The Second Chapter is all about literature review and related work. It describes the methods used so far for POS tagging and works that are done using Hybrid approach, combination of rule based and stochastic based. Chapter Three focuses on study of the nature, word class, sentence structure and tag set preparation of Afaan Oromo language. The Fourth Chapter deals with corpus preparation and design of Afaan Oromo POS tagger. Chapter Five deals with modifications and implementation of Brill's tagging algorithms for the architecture stipulated out in Chapter Four. Chapter Six mainly focuses on the experimental analysis of the part of speech tagger. Finally, the last chapter presents summary of the work done and future work.

# CHAPTER TWO

# LITERATURE REVIEW AND RELATED WORK

## 2.1. Literature Review

The main purpose of this Chapter is to give an overview of existing literature and methods used in the field of Natural Language Processing (NLP) with the special focus on part-of-speech tagging. Most language processing systems must recognize and interpret the linguistic structures that exist in a sequence of words [1]. This task is virtually impossible if all we know about each word is its text representation. Instead we want to be able to generalize over classes of words. These word classes are commonly named as Part-of-speech (POS). POS is a linguistic category of words that explains how word is used in a sentence [10]. Although different languages may have different classification schemes, some of the common lexical categories are Noun, Adjectives, Adverb, and Verb.

There are well-established sets of abbreviations for naming these classes, usually referred to as POS tags (For example. labels such as, NN for Noun, VV for Verbs and JJ for Adjectives). There is no standard representation for these parts of speech. Different researchers have used different symbols depending on the number of tag and morphological structure in the language under study. For example, Brown [1] uses VB for base form verb, WDT for wh-determiners, JJ for Adjectives, NN for singular proper noun and NNS for plural noun while others uses NN for all nouns, VV for verbs and ADJ for adjectives.

The collection of tags used for a particular task is known as a tagset. A corpus is a collection of texts from different areas such as newspaper text and scientific articles. A corpus in most cases contains extra information about every word such as its part-of-speech and morph-syntactic properties. Often the tagset is extended to include also morph-syntactic properties such as number and gender for nouns [11].

Conversely, to interpret words we need to be able to discriminate between different usages, such as a noun or as a verb. The process of classifying words in this way, and labeling them accordingly, is known as part-of-speech tagging, POS-tagging, or simply tagging [1]. We decide whether each word is a noun, verb, adjective, or whatever. The input to POS tagging is a

sentence or text of Natural Language and its output is a tagged word sequence. Here is an example of a tagged sentence:

**Abebe\NN is\VV a\AT clever\JJ student\NN**

There are essentially two sources of information for tagging. One way is to look at the tags of other words in the context of the word we are interested in. These words may also be ambiguous as to their part of speech, but the essential observation is that some part of speech sequences are common, such as AT JJ NN, while others are extremely unlikely or impossible, such as AT JJ VBP. This type of syntagmatic structural information is the most obvious source of information for tagging, but, by itself, it is not very successful because of many content words in a language can have various parts of speech, which results in loss of a lot of constraining information needed for tagging. These considerations suggest the second information source: just knowing the word involved (Lexical information) gives a lot of information about the correct tag.

Improper use of the input information remains the output of tagging with semantically incoherent reading. The tagging can also result in syntactically unlikely texts. Many words have more than one syntactic category. In tagging, we try to determine which of these syntactic categories is the most likely for a particular use of a word in a sentence. Tagging is also a problem of limited scope: instead of constructing a complete parse, we just fix the syntactic categories of the words in a sentence. For example, we are not concerned with finding the correct attachment of prepositional phrases.

Even though it is limited, the information we get from tagging is still quite useful. Part-of-speech tagging can be used in many applications such as machine translation, information retrieval, information extraction and grammar checking. Information extraction applications are using patterns for extracting information from text and often make reference to parts-of-speech in templates [12].

There are parts-of-speech which usually only contain a small number of words, such as conjunctions and prepositions, and these parts-of-speech are known as closed classes. It is not likely that new words will be added frequently to the closed classes. There are, however, closed classes which contain a large number of words such as numerals. The opposite of closed classes are the open classes containing parts-of-speech which usually have thousands of members and new members are added continuously, such as nouns and verbs. The distinction between closed

and open classes is relevant when handling unknown words, since these are more likely to belong to open classes.

One of the first distinctions which can be made among POS taggers is in terms of the degree of automation of the training and tagging process. Supervised taggers rely on pre-tagged corpora to help in the process of disambiguation. Unsupervised taggers on the other hand, use sophisticated computational methods to automatically induce word groupings (thus devising its own tagsets) from raw, untagged texts. Accordingly, it is able to calculate the probabilistic information needed by stochastic taggers or to induce the rules needed by rule based systems [13].

The unfortunate reality is that pre-tagged corpora are not readily available for the many languages and genres which one might wish to tag. Full automation of the tagging process addresses the need to accurately tag previously untagged genres and languages in light of the fact that manual tagging of training data is a costly and time-consuming process [3].

Throughout years a lot of different methods or approaches have been used to solve the lexical ambiguity of a word in a text. The most widely used are statistical methods, rule-based methods, and transformation-based learning and Artificial Neural Network methods.

### 2.1.1. Statistical Approach

Statistical NLP aims to do statistical inference for the field of natural language. Statistical inference in general consists of taking some data (generated in accordance with some unknown probability distribution) and then making some inferences about this distribution [1, 2, 13].

The statistical approach also called a stochastic approach includes frequency, probability or statistics. Any model which somehow incorporates frequency or probability, i.e. statistics, may be properly labeled as stochastic. Stochastic taggers exploit the power of probabilities and machine learning techniques in order to disambiguate and tag sequences of words [12, 13]. The simplest stochastic approach finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the un-annotated text.

An individual word conditional probability is the core idea for this approach. That is a given word W in a sentence is assigned a tag T based on certain condition, if the probability of the tag T of that given word W in that context is maximum. Tag with higher frequency is the one that will be assigned to the ambiguous word in the sentence. The frequency is the count of word with

particular tag divided by the total number of that tag usage. This can be represented in the following formula [1]:

$$P(W/T) = \frac{Countof(W,T)}{Countof(T)} \text{---------- (1)}$$

Where countof(W,T) is count of word(W) tagged with tag (T) and Countof(T) is count of tag (T) in the corpus.

An alternative to the word frequency approach is known as the N-gram approach that calculates the probability of a given sequence of tags. In this case, the $n^{th}$ word W is conditionally dependent on the previous n-1 entities (i.e. that can be words, tags or anything in the context). It determines the best tag for a word by calculating the probability that the word occurs with the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes. N-gram approach is just context dependency with the assumption n is limited. These are known as the Unigram, Bigram and Trigram models. Maximizing tag sequence alignment is another issue as HMM does.

The statistical approach is a non-deterministic approach in which the context defines the generality of the approach. As context increases its theoretical performance increases and vice versa. However, increasing the context demands huge collection of data which is not feasible in most cases. Context also defines dependencies of the tag of the word on its contextual information [1]. The problem with this approach is that it can come up with sequences of tags for sentences that are not acceptable according to the grammar rules of a language. This approach is not feasible because of the large number of parameters we would need. Even when considering more contexts, we had to smooth and interpolate since maximum likelihood estimates are not robust enough. Moreover, some changes in statistical methods may require re-annotation of the entire training corpus in the supervised statistical learning.

Different models have been used for stochastic POS tagging such as Hidden Markov Model (HMM), Maximum Entropy model and some others. The statistical part of speech taggers that uses Markov Model captures contextual and Lexical information. In the lexical model, every word has a set of probable parts-of-speech. However, only using a lexical model is not enough to tag with high accuracy. We also need a model to capture the context of tags. In the contextual model, every part-of-speech is conditioned on its neighboring parts-of-speech.

### 2.1.1.1. Hidden Markov Model

Hidden Markov Model (HMM) is one of the most commonly used probabilistic models which depend on Markov Models. Markov models are state-space models that can be used to model a sequence of random variables that are not necessarily independent [14]. HMM is then nothing more than the probabilistic function of Markov process, a process which moves through the state to state to find optimal state sequence.

In general, Hidden Markov Model is defined by specifying the set of {S, V, A, B, π}, which are:

- S = the set of states = $\{S_1, S2... S_n\}$
- V = the output alphabet = $\{v_1, v_2... v_m\}$
- π(i) = probability of being in state qi at time t = 0 (i.e., in initial states)
- A = transition probabilities = $\{a_{ij}\}$, where $a_{ij}$ = Pr[entering state qj at time t+ 1 j in state qi at time t]: Note that the probability of going from state i to state j does not depend on the previous states at earlier times; this is the Markov property.
- B = output probabilities = $\{b_{i(k)}\}$, where $b$ = Pr[producing vk at time t j in state qj at time t]

This sequence of states is also called Markov Chain. Suppose that X = ($X_1$... $X_T$) is a sequence of random variables taking values in some finite set S = {$S_1$… $S_N$} the state spaces, then the Markov properties are:

- **Limited Horizon: -**That is, future elements of the sequence are conditionally independent of past elements, given the present element.

$$P(X_{t+1} = S_k/X_1 \dots X_t) = P(X_{t+1} = S_k/X_t) \text{---------- (2)}$$

- **Time invariant (stationary):-** The above (limited horizon) dependency doesn't change over time

$$P(X_{t+1} = S_k/X_t) = P(X_2/X_1) \text{---------- (3)}$$

X is then said to be Markov Chain or to have a Markov Properties. Markov Chain can be represented by a stochastic transition matrix, A:

$a_{ij} = P(X_{t+1} = S_j/X_t = S_i)$, where$a_{ij} \geq 0$, $V_{ij}$ and $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$

Markov model can be thought of as a (nondeterministic) finite state automaton. The Markov properties ensure that we have a finite state automaton, there are no long distance dependencies, and where one ends up next depends simply on what state one is in.

In Markov Model tagging, we look at the sequence of tags in a text as a Markov chain. That is, we assume that a word's tag only depends on the previous tag (limited horizon) and that this dependency does not change overtime (time invariance). For example, if a finite verb has a probability of 0.2 to occur after a pronoun at the beginning of a sentence, then this probability will not change as we tag the rest of the sentence (or new sentences). These are:

$P(ti/t1, \dots, tn) = P(ti/ti-1)$ (Limited Horizon)

$P(ti/ti-1)$ (Time invariant)

We use a training set of manually tagged text to learn the regularities of tag sequences. The Maximum likelihood estimate of the tag $t_k$ following $t_j$ is estimated from the relative frequencies of different tags following certain tag.

$$P(tj\ /\ tk) = \frac{C(tj,tk)}{C(tj)} \text{---------- (4)}$$

It turns out that for an HMM the intuitive relative frequency estimates are the estimates which maximize the probability of the training data. In practice, the task is to find the most probable tag sequence for a sequence of words, or equivalently, the most probable state sequence for a sequence of words (since the states of the Markov Model here are tags). We incorporate words by having the Markov Model emit words each time it leaves a state (i.e. symbol emission probability).

$$P(W/T) = P(wi/ti) \text{ Or} P\left(W/_T\right) = \frac{C(W,T)}{C(T)} \text{---------- (5)}$$

Use of a Hidden Markov Model to do part-of-speech-tagging, as we will define it, is a special case of Bayesian inference, a paradigm that has been known since the work of Bayes. The intuition of Bayesian classification is to use Bayes' rule to transform into a set of other probabilities which turn out to be easier to compute. Bayes' rule gives us a way to break down any conditional probability *P* (*W*|T) into three other probabilities [1, 14]:

$$P(T|W) = \frac{P(W|T)P(T)}{P(W)} \text{---------- (6)}$$

We can then substitute to get:

$$\hat{t}_{1\ldots n} = \ argmax_T(\frac{P(W|T)P(T)}{P(W)}) \text{ ---------- (7)}$$

We can conveniently simplify the equation by dropping the denominator$P(W)$. Why is that? Since we are choosing a tag sequence out of all tag sequences, we will be computing $\frac{P(W|T)P(T)}{P(W)}$ for each tag sequence. But $P(W)$ doesn't change for each tag sequence; we are always asking about the most likely tag sequence for the same observation W, which must have the same probability $P(W)$. Thus we can choose the tag sequence which maximizes this simpler formula:

$$\hat{t}_{1\ldots n} = \ argmax_T P(W|T)P(T) \text{---------- (8)}$$

To summarize, the most probable tag sequence ˆt1...n given some word string W can be computed by taking the product of two probabilities for each tag sequence, and choosing the tag sequence for which this product is greatest. The two terms are the prior probability of the tag sequence P (T)), and the likelihood of the word string$P(W|T)$:

$$\hat{t}_{1\ldots n} = \ argmax_T P(W|T)P(T) \text{---------- (9)}$$

Unfortunately the equation is still too hard to compute directly. HMM taggers therefore make two simplifying assumptions. The first assumption is that the probability of a word appearing is dependent only on its own part-of-speech tag; that it is independent of other words around it.

$$P(W|T) \approx \prod_{i=1}^{n} P\ (wi/ti) \text{---------- (10)}$$

The second assumption is that the probability of a tag appearing is dependent only on the previous tag, the bigram assumption (1st order Markov model):

$$P(T)\ \approx \prod_{i=1}^{n} P(ti/ti-1) \text{---------- (11)}$$

Plugging the simplifying assumptions into the results in the following equation by which a bigram tagger estimates the most probable tag sequence. So the final equation for determining the optimal tags for a sentence is:

$$\hat{t}_{1\ldots n} = \ argmax_T P(W|T)P(T) = argmax_T P(wi|ti)P(ti/ti-1) \text{---------- (12)}$$

We could evaluate equation above for all possible tagging of a sentence of length ***n,*** but that would make tagging exponential in the length n, of the input that is to be tagged. An efficient

tagging algorithm is the Viterbi algorithm, an algorithm of three steps: initialization, induction, and termination and readout respectively. The Viterbi algorithm is a dynamic programming algorithm that computes two probabilistic functions. The probability of the most probable path is the max of over all states and the final state is the state that maximizes it. The state path can be reconstructed by tracing back through the dynamic programming matrix [1, 14, and 15].

In a visible Markov model, we know what states the machine is passing through, so the state sequence or some deterministic function of it can be regarded as the output. HMM operate at a higher level of abstraction by postulating additional "hidden" structure, and that allows us to look at the order of categories of the output sequence [1, 2, 12, and 14]. In general HMM are useful:

- When one can think of underlying events probabilistically generating surface events.
- They are one of a class of models for which there exist efficient methods of training through use of the Expectation Maximization (EM) algorithm. EM algorithm allows us to automatically learn the model parameters that best account for the observed data.

One widespread use of this is tagging, assigning parts of speech (or other classifiers) to the words in a text. When HMM is taken to the application of POS Tagging, the hidden states are the POS tags (tagsets) and the sequences of words are the sequence of observations. The transition probability in POS tagging is the probability of moving from one tag to the next tag and the emission probability is the probability of getting a word $W_i$ being in tag $T_i$.

Nevertheless, what is important is whether we encode a process as a Markov process, not whether we most naturally do. One might think that, for $n \geq 3$, such a model is not a Markov model because it violates the Limited Horizon condition where we are looking a little further into earlier history. But we can reformulate any model as a Markov model by simply encoding the appropriate amount of history into the state space. In general, any fixed finite amount of history can always be encoded in this way by simply elaborating the state space as a cross product of multiple previous states. In such cases, we sometimes talk of an $n^{th}$ order Markov model, where n is the number of previous states that we are using to predict the next state. Thus, an n-gram model is equivalent to an $(n-1)^{the}$ order Markov model [1].

### 2.1.2. Maximum Entropy Model

The Maximum Entropy Model (MEM) is based on the principle of Maximum Entropy, which states that when choosing between numbers of different probabilistic models for a set of data, the most valid model is the one which makes fewest arbitrary assumptions about the nature of the data [1]. The term, maximum entropy here means maximum randomness or minimum additional structure. It exploits some of the good properties of transformation-based learning and Markov model tagging. It allows flexibility in cues used to disambiguate words. The outputs of the maximum entropy tagging are tags and their probabilities.

The maximum entropy framework finds a single probability model consistent with the constraints of the training data and maximally agnostic beyond what the training data indicates. The probability model for MEM is defined over (H, X, T). The probability model is taken over a space H * T, where H is the set of environments in which a word appears or "histories" and T is the set of possible POS tags. Maximum entropy model specifies a set of features from the environment for tag prediction [16]. The features remind us transformation rules in transformation based learning. The model's probability of a history h together with a tag t is defined as:

$$P(h,t) = \pi\mu \prod_{j=1}^{k} \alpha_j^{fji(h,t)} \text{---------- (13)}$$

Where $\pi$ is a normalization constant, $\{a1 \ldots ak\}$ are the positive model parameters, $\{f\ 1 \ldots fk\}$ are known as features, where $fj(h,t)$ is in {0, 1} and each parameter aj corresponds to a feature$fj$.

A typical environment is specified as

$$hi = \{wi, wi+1, wi+2, wi-1, wi-2, ti-1, ti-2\} \text{---------- (14)}$$

Where, h stands for environment, w for word, t for tag and i for index. The above equation is for the $ith$ word, $wi$ whose preceding two words are $w_{i-1}$ and $w_{i-2}$ and the succeeding two words are $wi+1$ and$wi+2$, and the previous two tags are $ti-1$ and$ti-2$.

Given the environment, a set of binary features can be defined. Following is the $jth$ feature and is on or off based on environment properties.

$$fj(hi, ti) = \{1\ if suffix(wi) = \text{ing} and ti = PastPartV \text{----------} (15)$$

$$\{0\ otherwise$$

That is, the feature mentioned above will be on (i.e. 1) if the suffix of the word in question is ing and the tag is past participle and will be off (i.e. 0) if not. Features are generated from feature templates. For the above feature, the template is

$$Xisasuffixofwi, |X| < 5\ ANDti = T$$

Where X and T are variables

A set of features and their observed probabilities are extracted from the training set. Generalized Iterative scaling method is then used to create the maximum entropy model consistent with the observed feature probabilities. Now we get the model trained.

Like Markov model tagging, most probable tag sequence according to the probability model is built. Beam search is used for this purpose, keeping n most likely tag sequences up to the word being tagged. Unlike Markov model approach, there is a great deal of flexibility in what contextual cues can be used.

## 2.2. Rule-Based Approach

Rule-based part-of-speech tagging is the oldest approach that uses hand-written rules for tagging. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. Disambiguation is done by analyzing the linguistic features of the word, its preceding word, its following word and other aspects. For example, if the preceding word is article then the word in question must be noun. This information is coded in the form of rules. The rules may be context-pattern rules or as regular expressions compiled into finite-state automata that are intersected with lexically ambiguous sentence representations.

Rule-based taggers try to assign a tag to each word in a sentence using a set of hand written rules. Rules are based on knowledge of the specific language which may consist of a large number of morphological, lexical and syntactical information. These rules can be obtained manually that are handcrafted by linguistic professionals or through machine learning. In handcrafted rules the set of rules must be properly written and checked by human experts. The

Machine learning rule is the result of transformation-based learning, which will be discussed later in the following section.

Rule-based taggers generally consist of two phases. The first phase is concerned with getting all possible tags of each word of the sentence and the second phase is concerned with identification of the correct tag by using some hand written rules. It is the core component of the rule-based tagger. All words are given unique codes based on their grammatical word categories. Lexicon contains words and the corresponding part of speech tags taken from the tag set. The tagger starts its processing by first looking up each word in the lexicon with its corresponding part of speech.

The rules are called tag changing rules that provide information about appropriateness of a given tag based on the context. These rules could specify, for instance that a word following a determiner and an adjective must be a noun. Rule can be contextual rules, which are predefined rules based on context, or it can be lexical rules that help the tagger to make reasonable guess. Contextual rules modify the tag of the word based on the surrounding words whereas lexical uses morphological behavior of the word itself.

Typical rule based approaches use contextual information to assign tags to unknown or ambiguous words. These rules are often known as context frame rules. As an example, a context frame rule might say something like: If an ambiguous/unknown word X is preceded by a determiner and followed by a noun, tag it as an adjective.

DET - X - n = X/JJ---------- (16)

In addition to contextual information, many taggers use morphological information to aid in the disambiguation process. One such rule might be: if an ambiguous/unknown word ends in an -ing and is preceded by a verb, label it a verb (depending on your theory of grammar, of course).

Some systems go beyond using contextual and morphological information by including rules pertaining to such factors as capitalization and punctuation. Information of this type is of greater or lesser value depending on the language being tagged. In German for example, information about capitalization proves extremely useful in the tagging of unknown nouns [17].

Rule based taggers most commonly require supervised training; but, very recently there has been a great deal of interest in automatic induction of rules. One approach to automatic rule induction is to run an untagged text through a tagger and see how it performs. A human then goes through

the output of this first phase and corrects any erroneously tagged words. The properly tagged text is then submitted to the tagger, which learns correction rules by comparing the two sets of data. Several iterations of this process are sometimes necessary.

A criticism of rule-based taggers is the amount of effort necessary to write the disambiguation rules. For the rule based tagger, much time is spent to develop a rule-set. In addition, the rules in rule-based systems are usually difficult to construct and typically not robust. It is quite easy (and most often needed) to devise a small set of rules that make sense, even in a broader syntactic context. But tuning those systems to achieve good performance is a laborious and dedicated task.

### 2.2.1. Transformation-Based Approach

Transformation-based tagging combines the benefits of both rule-based and probabilistic Approach. It picks the most likely tag based on the training corpus and then applies a certain set of rules to see whether the tag should be changed to anything else. This approach will introduce several errors. The next step is to correct as many errors as possible by applying transformation rules that the tagger has learned. It saves any new rules that it has learnt in the process, for further use. One example of an effective tagger in this category is the Brill Tagger.

One of the strengths of this method is that it can exploit a wider range of lexical and syntactic regularities. In particular, tags can be conditioned on words and on more contexts. Transformation-based tagging encodes complex interdependencies between words and tags by selecting and sequencing transformations that transform an initial imperfect tagging into one with fewer errors [3]. The training of a transformation-based tagger requires an order of magnitude fewer decisions than estimating the large number of parameters of a Markov model.

Transformation based learning usually starts with some simple solution to the problem. Then it runs through cycles. At each cycle, the transformation which gives more benefit is chosen and applied to the problem. The algorithm stops when the selected transformations do not add more value or there are no more transformations to be selected. This is like painting a wall with background color first, then paint different color in each block as per its shape or so. TBL is best suitable for classification tasks.

In TBL, accuracy is generally considered as the objective function. So in each training cycle, the tagger finds the transformations that greatly reduce the errors in the training set. This

transformation is then added to the transformation list and applied to the training corpus. At the end of the training, the tagger is run by first tagging the fresh text with initial-state annotator, then applying each transformation in order wherever it can apply.

Transformation-based tagging has two key components:

- A specification of which error-correcting transformations are admissible
- The learning algorithm

As input data, we need a tagged corpus and a dictionary. We first tag each word in the training corpus with its most frequent tag that is what we need the dictionary for. The learning algorithm then constructs a ranked list of transformations that transforms the initial tagging into a tagging that is close to correct. This ranked list can be used to tag new text, by again initially choosing each word's most frequent tag, and then applying the transformations.

The learner is given allowable transformation types. A tag may change from X to Y if the previous word is W, the previous tag is Ti and the following tag is Tj, or the following word is W.

A transformation consists of two parts, a triggering environment and a rewrite rule. Rewrite rules have the form t1 -> t2 meaning replace tag t1 by tag t2. Examples of the type of transformations that are learned given the triggering environments are shown in table 2.1.

*Table 2.1 Examples of some transformations learned in transformation-based tagging.*

| ***Source tag*** | ***Target tag*** | ***Triggering environment*** |
|---|---|---|
| NN | VV | previous tag is TO |
| VVP | VV | one of the previous three tags is MD |
| JJR | ADV | next tag is JJ |
| VVP | VV | one of the previous two words is *n't* |

The first transformation specifies that nouns should be retagged as verbs after the tag TO (the word 'to'). Later transformations with more specific triggers will switch some words back to NN (e.g., school in go to school). The second transformation in table 2.1 applies to verbs with identical base and past tense forms like *cut* and put. A preceding modal makes it unlikely that they are used in the past tense. An example for the third transformation is the retagging of more in more valuable player.

The first three transformations in table 2.1 are triggered by tags. The fourth one is triggered by a word (words like don't and shouldn't are split up into a modal and n't). Similar to the second transformation, this one also changes a past tense form to a base form. A preceding n't makes a base-form more likely than a past tense form.

Word-triggered environments can also be conditioned on the current word and on a combination of words and tags (The current word is $wi$ and the following tag is $tj$). There is also a third type of transformation in addition to tag-triggered and word-triggered transformations. Morphology - Triggered Transformations offer an elegant way of integrating the handling of unknown words into the general tagging formalism. Initially, unknown words are tagged as proper nouns (NNP) if capitalized, as common nouns (NN) otherwise. Then morphology-triggered transformations replace NN by NNS if the unknown word's suffix is -s these transformations are learned by the same learning algorithm as the tagging transformations.

The learning algorithm of transformation-based tagging selects the best transformations and determines their order of application. Initially we tag each word with its most frequent tag. In each iteration of the loop, we choose the transformation that reduces the error rate most, where the error is measured as the number of words that are miss-tagged in tagged corpus. We stop when there is no transformation left that reduces the error rate by more than a pre specified threshold. This procedure is a greedy search for the optimal sequence of transformations.

We also have to make two decisions about how to apply the transformation. First, we are going to stipulate that transformations are applied from left to right to the input. Secondly, we have to decide whether transformations should have an immediate or delayed effect. In the case of immediate effect, applications of the same transformation can influence each other. Brill implements delayed-effect transformations, which are simpler [1, 7].

An interesting twist on this tagging model is to use it for unsupervised learning as an alternative to HMM tagging. As with HMM tagging, the only information available in unsupervised tagging is which tags are allowable for each word. We can then take advantage of the fact that many words only have one tag and use that as the scoring function for selecting transformations. Brill describes a system based on this idea that achieves tagging accuracies of up to a remarkable result for an unsupervised method. What is particularly interesting is that there is no overtraining.

Advantages of Transformation Based Learning

1. Small set of simple rules that are sufficient for tagging is learned.
2. As the learned rules are easy to understand, development and debugging are easier.
3. Interlacing of machine-learned and human-generated rules reduce the complexity in tagging.
4. Transformation list can be compiled into finite-state machine resulting in a very fast tagger. A TBL tagger can be even ten times faster than the fastest Markov-model tagger.
5. TBL is less rigid in what cues it uses to disambiguate a particular word. Still it can choose appropriate cues.

Disadvantages of Transformation Based Learning

1. TBL does not provide tag probabilities.
2. Training time is often intolerably long, especially on the large corpora which are very common in Natural Language Processing.

### 2.2.2. Artificial Neural Network Approach

An Artificial Neural Network is a method of information processing that is motivated by the way biological nervous systems process information. The most important feature of this paradigm is the structure of the information processing systems. The system is composed of a large number of highly interconnected processing elements (neurons) working in harmony to solve specific problems. These units are highly interconnected by directed weighted links; associated with each unit as an activation values. Though this connection, this activation is propagated to other units. The interconnections of the neurons follow specific network architecture [18].

Artificial Neural Networks consist of three layers namely an input layer, hidden layer and output layer. The input layer which is connected to the hidden layer represents the raw information that is fed to the network as an input so that it can learn and adapt properties. The middle layer, so called hidden layer, connected to the output layer is determined by the activities of the input unit and the weights on the connections between the input and hidden units. The output layer represents the result of the learning properties from the input layer and hidden layer.

Taking the ANN approach to the application of part of speech tagging, first preprocessing activities are performed before dealing actually with the ANN based part of speech tagger. Such

preprocessing activities can be tokenizing, feature extraction like POS information, word information, POS category and order information etc. The results of the preprocessing activities are given to the input layer of the network from which the network can learn pattern. As mentioned above, the input layer is connected to the hidden layer and in this layer different algorithms like error back-propagation algorithm, an algorithm based on an error-correction learning rule specifically on the minimization of the mean squared error which is a measure of the difference between the actual and the desired output, can be used for training the system [19].

This technique of tackling the problem of assigning part of speech tags to words has some disadvantages compared to the HMM and rule based approaches. Some of these are: The HMM method assigns the sequence of tags for the sequences of words in the entire sentence i.e. it takes the due consideration of the sentence structure. The same thing is true with rule based approaches which tend to take the sentence structure and generally the linguistic patterns into consideration [12].

### 2.2.3. Hybrid Approach

Hybrid Approach as its name implies takes the benefits of the different approaches described above, combine and apply them to certain POS tagging problems. This can be the combination of statistical approach and rule based approach or transformation based approaches. The performance and accuracy of using hybrid approach is usually better than using single approach because the hybrid approach takes the advantages from the different approaches to improve the performance of the system.

## 2.3. Related Work

So far we have discussed different approaches on POS tagging. This section describes the review of related work on part of speech tagging. Different languages are considered including Afaan Oromo, which is the focus of this study. Various researchers in the field of computational linguistics have used different approaches and methods in order to solve the part of speech tagging problems within particular languages.

The first work on Afaan Oromo language part of speech tagging, which uses statistical approach with Hidden Markov Model [8], was done in Addis Ababa University in 2009. The researcher has collected 159 Afaan Oromo sentences (with 1621 distinct words) from different sources and

used 17 tag sets to annotate these sentences. He divided these sentences into training set and test set. The HMM based Afaan Oromo part of speech tagger was trained on the training set in order to compute and store the lexical and contextual probabilities of the words in the training set. The tagger then took untagged Afaan Oromo text as an input and tokenized the sentences into words before actually assigning the part of speech tags sequence. After this, each token in the sentence is assigned with a correct part of speech tag sequence that is done using unigram and bigram models of the Viterbi algorithm by taking the knowledge from lexical and contextual probabilities gained during the training session. The researcher has tested the performance of the tagger and got an accuracy of 87.58% and 91.97% for the unigram and bigram models respectively. As a first attempt for Afaan Oromo language, the result of the study is promising and considered to be the base for future work.

A transformational error driven learning approach was used in the work of [9] in Addis Ababa University in 2010 for Afaan Oromo language. In this work, the researcher has adapted the Brill Transformational error driven learning with some modifications on the tagger template. This approach can be considered as a hybrid (rule based + stochastic) as it learns rules automatically during the training that makes it rule based and uses supervised learning method (it uses part of speech tagged training corpus) to get the statistical property of words like lexical probability and contextual probability from the training corpus that makes it stochastic approach.

The researcher has used 233 sentences (1708 distinct words) of Afaan Oromo language which he divided into training set and testing set. He used 18 tagsets to tag the 233 sentences. He conducted different experiments to test the performance of the tagger for both the original and modified Brill tagger. Accordingly, he has got 80.08% accuracy for the modified Brill tagger. The performance of the original Brill tagger on Afaan Oromo text was found to be 77.64%. The performance improvement for the modified tagger, as the researcher has stated, is due to the adjustment made for the Afaan Oromo learning template. The small size of corpus and tagset can be considered as the limitations of the study.

The work of [20] using a hybrid approach; HMM tagger combined with rule based tagger in 2010 is the first for Tigrigna part of speech tagging. It uses 36 broad tag sets and 26,000 words from around 1000 sentences containing 8000 distinct words were tagged for training and testing purpose. The researcher used raw Tigrigna text to first tag by the HMM tagger; afterwards the

rule based tagger is used as a corrector of the HMM tagger. Viterbi algorithm and Brill Transformation-based Error driven learning are adapted for the HMM and Rule based taggers respectively. Taking 25% of the whole data for testing, the accuracy of the HMM and rule based approaches 89.13% and 91.8% respectively whereas, the hybrid model's performance is 95.88%. Hence, the result obtained shows that the hybrid of the two taggers outperforms the individual taggers.

In POS tagging, using a hybrid approach (Rule-based and neural network) for Amharic language [21], neural network output anomaly was corrected by rule-based approach. Back propagation algorithm and Brill transformation based learning method are adapted for development of Amharic tagger. Relatively, a corpus with large amount of data is used to train and test the tagger. The experiment result of this work indicates that 91% and 94% accuracy for rule-based and neural network tagger respectively. But the result reached 98% when the experiment was conducted on the hybrid tagger.

In the work on improving Brill's POS tagger for an Agglutinative Language [22], Brill's rule based POS tagger is tested and adapted for Hungarian. It is shown that the present system does not attain as high accuracy for Hungarian as it does for English (and other Germanic languages) because of the structural difference between these languages. The tagger has the greatest difficulties with part-of–speech belonging to open classes because of their complicated structure. It is also shown that the accuracy of tagging can be increased from approximately 83% to 97% by simply changing the rule generating mechanisms, namely the lexical templates in the lexical training module.

Resolving POS Ambiguity in the Greek language using Learning techniques [23] is another research area on POS tagging. It investigates the use of Transformational-Based Error-Driven learning for resolving part-of-speech ambiguity in the Greek language. The aim is not only to study the performance, but also to examine its independence on different thematic domains. Results are presented here for two different test cases: a corpus on "management succession events" and a general-theme corpus. The two experiments show that the performance of this method does not depend on the thematic domain of the corpus, and its accuracy for the Greek language is around 95%.

The work of [17] used Rule-based approach with Brill Tagger for German language. It has shown how the tagging performance improves with increasing corpus size. According to the work, training over a corpus of only 28,500 words results in an error rate of around 5% for unseen text. In addition, it has demonstrated that the error rate can be reduced by looking up unknown words in an external lexicon, and by manually adding rules to the rule set that has been learned by the tagger. The researchers thus obtained an error rate of 2.79% for the reference corpus to which the manual rules were tuned. For a second general reference, corpus lexical-lookup and manual rules leaf to an error rate of 4.13%.

## 2.4. Summery

Natural Language processing is one of the current hot research areas for scientists and academic researchers. There are different approaches for natural language processing. Such as Transformation-based learning, statistical learning using Hidden Markov models, statistical learning using Maximum Entropy models, Neural Networks, Support Vector Machines and hybrids of them. All of these methods have their own pros and cons.

Natural languages is ambiguous in many cases. Therefore, researchers have been using different modification techniques to adopt the methods with the structure of languages for better performance. It is also shown that it is difficult to capture detail knowledge of tag transformation of the language without using sufficient size of corpus.

As it is mentioned in Chapter One, this research is an extension of the work done in [9], which uses TEL for Afaan Oromo. Accordingly, Brill's TEL part-of-speech tagger is tested and adopted for Afaan Oromo Language. An algorithm of the original Brill's tagger is modified considering the nature of the language understudy. We have used this tool for part-of-speech tagging of Afaan Oromo words with the help of thesaurus and large training corpus.

# CHAPTER THREE

# AFAAN OROMO LANGUAGE AND TAGSET PREPARATION

## 3.1. Introduction

In the previous chapter some of literatures on the techniques of part of speech tagging were described. Review of related works on the area was also discussed. This chapter mainly deals with the structure and word classification of the language understudy, in this case Afaan Oromo. Afaan Oromo is an official language of Oromia regional state in Ethiopia.

The Oromo tribe makes up a significant portion of the population of Ethiopia. According to the 2007 Ethiopian census, over 37% of Ethiopian claim Oromo descent, making them the largest ethnic group in the country [24].The Oromo nation has a single common mother tongue and basic common culture. The ethnic origin of this group is assumed as Southern part of East Africa who moved northward into Ethiopia in the 16$^{th}$ Century. They are found in almost all regions of Ethiopia, including the neighboring countries such as Kenya and Somalia [7].

The Oromo language, Afaan Oromo, is an Eastern Cushitic language of the Afro-Asiatic language family, and the most widely spoken of the forty or so Cushitic languages. Afaan Oromo is spoken over a geographically wide expanse that includes Ethiopia, Kenya, and parts of East, South and North Africa. The language has recently begun to use the Latin alphabet, which is relatively best suited for transcription of Afaan Oromo [25].

Afaan Oromo is considered as one of the five most widely spoken languages from among the approximately one thousand or so languages of Africa [26]. Taking into consideration the number of speakers and the geographic area it covers, Afaan Oromo, most probably rates second among the African indigenous languages. It is the third most widely spoken language in Africa, after Arabic and Hausa. According to the 2007 Ethiopian census, Afaan Oromo has about 34%speakers of the total population. Perhaps not less than two million non-Oromo speak Afaan Oromo as a second language [24]. In fact Afaan Oromo is a lingua franca in the whole of Ethiopian Empire except for the northern part. It is a language spoken in common by several members of many of the nationalities like Harari, Anuak, Barta, Sidama, Gurage, etc., who are neighbors to Oromo.

## 3.2. Afaan Oromo Phonemes

Phoneme represents the smallest unit of natural languages. The Afaan Oromo phoneme includes twenty-eight consonants, five short and five long vowels. The difference in length is contrastive, for example, *lafa* 'earth', *laafaa* 'light'. Geminates (double consonants) are tolerated, diphthongs (double discrete vowels) do not occur, and consonant clusters are attested although highly restricted. Afaan Oromo has another glottalzed phone that is more unusual, an implosive retroflex stop, "dh" in Oromo orthography, a sound that is like an English "d" produced with the tongue curled back slightly and with the air drawn in so that a glottal stop is heard before the following vowel begins [26].

The syllable structure of Afaan Oromo can be schematized as follows: CV (V) (C), where C is a variable for 'consonant', V is a variable for 'vowel', VV represents a long vowel, and items in parentheses are optional.

## 3.3. Afaan Oromo Sentence Structure

Afaan Oromo follows Subject-Object-Verb (SOV) format. But because it is a declined language (nouns change depending on their role in the sentence), word order can be flexible, though verbs always come after their subjects and objects. Nouns precede modifiers, articles, pronouns, and case markers. Verbs follow their noun phrase arguments and occasionally their modifiers [5].

For a sentence of the language to be meaningful, it should follow the proper standard word order. If not the sentences may convey vague meaning or totally lose their meanings. Understanding of the structure of sentences can help us to know the relationship between words, which in turn lets us to categorize them correctly.

## 3.4. Afaan Oromo Word Classes

In this work, we discuss the basic Afaan Oromo word classes, which are standard for most linguists. This classification depends on the word's contribution and meaning in a sentence. These are noun (Maqaa), pronoun (Bamaqaa), adjective (ibsamaqaa), verb (Xumura), adverb (IbsaXumura), conjunction (Walqabsiistota), preposition (Durduuba) and Interjection (Rajjeeffannoo). These eight classes of words represent Afaan Oromo part of speech.

The three Afaan Oromo word classes, Noun, Verbs and Adjectives, are in the category of open classes and the rest namely Pronouns, Adverbs, Prepositions and Conjunctions, are in the category of closed classes. Interjections are words without syntactic functions. In this categorization, interjections are not considered as part-of-speech/word class.

### 3.4.1. Afaan Oromo Noun (Maqaa)

Afaan Oromo nouns are words used to name or identify any of categories of things, people, places or ideas or a particular one of these entities. Whatever exists, we assume, can be named, and that name is a noun. Many (but not all) Oromo nouns inflect for gender (masculine, feminine), while all inflect for number (singular – specific vs. non-specific, plural) and case (nominative, accusative, dative, genitive, instrumental, locative, ablative, and vocative) [26].

As in other languages like English, nouns in Afaan Oromo have different types or classes. There are proper and common nouns, collective nouns, and concrete and abstract nouns.

Proper nouns are nouns that represent a unique entity (like a specific person or a specific place) while Common nouns are nouns which describe an entire group of entities (examples would be the nouns of village or women). Proper nouns as a general rule are capitalized in Afaan Oromo language. Common nouns as a general rule are not capitalized. Examples from Afaan Oromo include:

- Calaan Finfinee deeme. “Chala went to Finfine” (proper nouns)
- Obbo Magarsaan, hayyuu dha. “Mr Magarsa is a wise man.” (Common noun)

Collective nouns name groups consisting of more than one individual or entity. The group is a single unit, but it has more than one member. Examples include "matii", "koree", "ummata","waldaya", and “saba”.

Concrete nouns refer to their ability to register on your five senses. If you can see, hear, smell, taste, or feel the item, it's a concrete noun. Abstract nouns on the other hand refer to abstract objects such as ideas or concepts, like the nouns *gowwummaa* “foolishness” or *hiyyummaa* “poorness”.

Nouns in Afaan Oromo can be used in two different places in a sentence. The first one is when the noun comes in the first positions it takes suffix:-ni, -n, ykn or –i. For example in the following sentences, the underlined words are nouns:

- Namni milla lama qaba. “A man has two legs.”
- Konkolatan sun dhufe. “That car came.”

The second position of noun in a sentence is when it comes at the middle, in front of verb. For Example in the following sentences:

- Konkotlataan Tolaa dhufe. “Tola’s car came”
- Namni saree guddisa. “A man grew a dog”

### 3.4.2. Afaan Oromo Pronoun (Bamaqaa)

In linguistics and grammar, a pronoun is a word or forms that substitute for a noun. It is a particular case of a pro-form. This is used in place of noun in most cases when a noun is pre stated. For Example in the following sentences,

- Inni fira kooti. “he is my relative”
- Isaan durata’aa wajjirichaati. “he is a leader of the office”

In Afaan Oromo, there are different categories of pronoun, which includes Personal pronoun (Bamaqoota Matayyaa), Reflexive and reciprocal pronouns (Bamaqoota Walummaa fi Bamaqoota Ofiiffee), Possessive pronoun (Bamaqoota Abbummaa), Interrogative pronoun (Bamaqoota gaaffii).

Oromo uses plural pronouns (isin and isaan) also as the polite/formal pronouns. Mostly, one uses the polite form when talking to/about older and respected members of the community. In many areas of southern Oromia, ati “you” is rarely used (and considered rude) and only the polite form of “you”, isin, is used. Table 3.1. Shows Oromo Personal Pronouns

***Table 3.1 Oromo Personal Pronouns***

| **Subject Pronouns** | | **Direct Object Pronouns** | |
|---|---|---|---|
| **Afaan Oromo** | **English** | **Afaan Oromo** | **English** |
| Ani | I | Na | Me |
| nuti, nu'i | We | Nu | Us |
| Ati | you | Si | You |
| Isin | you (pl.) | Isini | you (pl.) |
| Inni | he, it | Isa | him, it |
| Isheen | she | Ishee | Her |
| Isaan | they | Isaani | Them |

Like English, Afaan Oromo uses different forms of personal pronouns to indicate their role in the sentence. While “he” and “him” may refer to the same person, English uses “he” for subjects and “him” for objects. Afaan Oromo has several forms for all nouns, including pronouns, though for now we will only deal with the subject (nominative) and direct object (accusative) forms [26]. Consider the following examples.

- **Isheen isa jaalatti** ‘She likes him’
- **Inni ishee jaalata** ‘He likes her’
- **Nuti isa binna** ‘We buy it’
- **Ati na dhageessa**? Or more commonly **Na dhageessaa?** ‘Do you hear me?’

### 3.4.3. Afaan Oromo Adjective (IbsaMaqaa)

Adjective is used to describe noun in a sentence. Afaan Oromo adjective form a very small class in the language and inflect for gender and number. Nouns are typically used attributively to achieve the effect of adjectival modification.

***Table 3.2: Afaan Oromo Adjectives***

| Descriptive | Possessive | Interrogative | Quantitative | Numbers and Rank |
|---|---|---|---|---|
| Guddaa<br>Dheeraa<br>Diima | Keenna<br>Kee | Maalii<br>Akkamii<br>Kam | Hedduu<br>Mara | Tokko<br>Lama<br>Tokkoffaa<br>Lamaffaa |

Adjectives in Afaan Oromo have a gender; they are either feminine (ending with –u) or masculine (ending with –a). They are following the nouns they describe. For example in the sentences:

**Tolasaan Diimadha**‘Tolasa is red.’ and **Intalli bareeddun heerumte** ’beautiful girl is married’. Diima ‘red’ and bareeddu ‘beautiful’ is an adjective that describes Tolasaan and Intalli (which are nouns) respectively.

### 3.4.4. Afaan Oromo Verb (Xumura)

Verbs are the type of word class that stand to show action. In Afaan Oromo verbs always come at the end of each sentence. An Oromo verb consists minimally of a stem, representing the lexical meaning of the verb, and a suffix, representing tense or aspect and subject agreement. For

example, in dhufne 'we came', dhuf- is the stem ('come') and -ne indicates that the tense is past and that the subject of the verb is first person plural.

As in many other Afro-Asiatic languages, Oromo makes a basic two-way distinction in its verb system between the two tensed forms, past (or "perfect") and present (or "imperfect" or "non-past"). Each of these has its own set of tense/agreement suffixes.

Verbs inflect for person, gender, number, tense-aspect, mood, and voice. Tense marking does not play a major role in the language – the language divides events in time in two ways: complete (perfective/past) and incomplete (progressive – involving the present or future). Compound tenses are possible and are formed with a variety of auxiliary verbs. Several grammatical moods are attested: indicative, interrogative, imperative, and jussive (a directive mood that signals a speaker's command, permission, or agreement) [5].

### 3.4.5. Afaan Oromo Adverbs (Ibsa Xumura)

Adverbs are any word that explain or modify verbs. Adverbs form a large class of expressions in the language and bear case morphology. These can be adverbs of time, place, manner, frequency etc. For example, in the following sentences:

- **Tolaan suuta deema**. This means 'Tolaa walks slowly'; suuta 'slowly' is an adverb (adverb of manner).
- **Jallaneen bor deemti**. This means 'Jallanee will go tomorrow.' *bor* 'tomorrow' is an adverb (adverb of time).
- **Kananisaan yeroo hundaa ni mo'ata**. This means 'Kananisa wins every time.' *Yeroohundaa* every time is an adverb (adverb of frequency).
- **Gammachuun achi deeme**. This means 'Gamachu went there' achi 'there' is an adverb (adverb of place).

### 3.4.6. Afaan Oromo Conjunction (Wal qabsiistu)

A word that can be used to join or connect two phrases, clauses and sentences is known as a conjunction. Conjunction can be divided into coordinating and subordinating conjunctions. Coordinating conjunctions are used to connect two independent clauses. Mostly, these conjunctions are used when the speaker needs to lay emphasis on the two sentences equally.

Some of these conjunctions in Afaan Oromo include garuu ‘but’, moo ‘or’, kanaafuu ‘therefore’, ‘haata umalee‘ however/so’, ta’u illee ‘even though’ etc. Consider the following example.

- **Tolaan kaleessa dhufe garuu nu bira hin bule**. ‘Tola came yesterday but he didn’t spend the night with us’
- **Adii fi gurracha** ‘white and black’
- **Obbo Magarsaan har’a yookiin bor dhufu** ‘Mr. Magarsa will come today or tommorow’

### 3.4.7. Afaan Oromo Preposition (Durduube)

Whereas conjunctions usually link more complete thoughts together, prepositions link nouns to other parts of the sentence. In Afaan Oromo, both prepositions and postpositions exist; however, the use of postpositions is preferred and occurs with a higher frequency than the use of prepositions. Some common prepositions and postpositions are listed in Table 3.3.

***Table 3.3 Afaan Oromo Preposition and Postpositions***

| **Postpositions** | **Prepositions** |
|---|---|
| **ala** — out, outside | **gara** — towards |
| **bira** — beside, with, around | **eega, erga** — since, from, after |
| **booda** — after | **haga, hanga** — until |
| **cinaa** — beside, near, next to | **hamma** — up to, as much as |
| **dur, dura** — before | **akka** — like, as |
| **duuba** — behind, back of | **waa'ee** — about, in regard to |
| **irra** — on | |
| **irraa** — from | |
| **itti** — to, at, in | |
| **jala** — under, beneath | |
| **jidduu** — middle, between | |
| **keessa** — in, inside | |
| **malee** — without, except | |

### 3.4.8. Afaan Oromo Introjections (Raajii)

These are words that are used to express emotions, pleasure, sorrow or suddenly happening situations. Introjections have their own word expressions in many languages. Afaan Oromo’s introjections include *ishoo* for happiness *wayyoo* for sadness *ah* for silent event or situation happened.

### 3.4.9. Afaan Oromo Numeral (Lakkoobsa)

Numerals include words that refer to number or quantity of something. It can be cardinals such as sadii (three), afur (four) which come after the noun they modify, so that “two mangoes” is “mangoo lama”, just as “five birr” is “qarshii shan” and 200 is “dhibba lama”. Ordinal numbers are formed by adding the suffix -ffaa or -affaa to the number. Fractions can be expressed by saying the numerator as a cardinal number and then the denominator as an ordinal number. For example: ½ “tokko-lammaffaa”.

When the same number is repeated, it applies to all items. Thus, “lama lama” means “everything is two (birr)”. Two numbers said together indicate amount of birr for number of items, as in “lama sadii” for “two (birr) for three (items)”.

## 3.5. Afaan Oromo Tags and Tag sets

In the previous sections, the broad categories of nine Afaan Oromo word classes are explained. In this section, the actual tags used in this thesis work are discussed. Tags are the labels used for adding more information concerning the lexical category of each word in a sentence and tagsets are the collection of the tags used for developing the Afaan Oromo part of speech tagger.

The tagsets that are discussed below are classified as a basic class and subclasses of the basic class where noun, pronoun, verb, adjective, preposition, conjunction, adverb, interjection are considered to be the basic classes. In addition, numeral and punctuation are also included as basic classes in the process of identifying the tagsets.

Since part of the corpus prepared for this study was adopted from [9] corpus, tagset selected for study is also based on the work of [9]. Around 26 tagsets were identified in this work. 18 of the tagsets are adopted from the work of [9]. In order to consider more morphological structure and lexical category of the language, 8 (eight) additional tags are introduced in this research. The identification of the tags is made by taking 11 word classes namely: noun, pronoun, verb, adjective, adverb, preposition, conjunction, numerals, punctuation, introjections, and negation as the basic tags and others are derived from combination of or these basic classes. List of all Afaan Oromo tags is shown in Table 3.4 below

***Table 3.8: Afaan Oromo Tags set***

| ***S/NO*** | ***Basic category/tag*** | ***Derived category/tag*** | ***Description*** | ***Example*** |
|---|---|---|---|---|
| 1. | Noun | NN | Noun | Nyaata |
| 2. | | NPROP | Proper noun | Hundee<br>Caalaa<br>Lalisa |
| 3. | | NC | Noun + conjunction | Jaallannee-ffii |
| 4. | | NP | Noun + Preposition | Arsii-tti |
| 5. | Pronoun | PP | pronoun | Isii<br>Isa<br>Isaan |
| 6. | | PS | Preposition + pronoun | Isii-tti |
| 7. | | PC | Pronoun + conjunction | Isii-fi |
| 8. | | PREF | Reflexive pronoun | Ofii, walii |
| 9. | | PD | Demonstrative pronoun | Kuni, suni |
| 10. | | PDPR | Preposition + demonstrative pronoun | Suni –s |
| 11. | Verb | VV | verb | Barresite<br>Dubise |
| 12. | | AX | Auxiliary | Ta'e<br>Rafe |
| 13. | | VC | Verb + conjunction | Beekti<br>Beekna<br>Hinbeeknu |
| 14. | Adjective | JJ | adjective | Guddaa<br>Cimaa |
| 15. | | JC | Adjective _ conjunction | Qaloo –fii |
| 16. | | JP | Preposition + adjective | Gudda –tti |
| 17. | Adverb | ADV | adverb | Dilbata, wixata |
| 18. | | ADVPREP | Preposition + adverb | Dilbata –rraa |
| 19. | | ADVC | Adverb + conjunction | Dilbataa-fii |
| 20. | Preposition | PR | Preposition | -tti, oolee |
| 21. | Conjunction | CC | conjunction | Fii, yookan |
| 22. | Numerals | ON | Ordinal number | Tokko, lama |
| 23. | | JN | Cardinal Number | Tokkoffaa, saffaa |
| 24. | Punctuation | PUNC | Punctuation | .,!? |
| 25. | Interjection | II | Interjection | Ishoo, wayyoo |
| 26. | Negation | NG | Negation | Hin |

The 8 (eight) additional tags (highlighted on Table 3.8) that are added in this work are: NPROP for proper noun, PREF for reflexive pronoun, PD for demonstrative pronoun, PDPR for demonstrative pronoun joined with proposition, VC for verbs with conjunction, JP for adjective joined with preposition, ADVPREP for adverb with preposition, ADVC for adverb with conjunction.

# CHAPTER FOUR

# DESIGN OF AFAAN OROMO POS TAGGER

## 4.1. Introduction

The process of labeling each word with its part of speech tagger is an important component of a natural language processing (NLP) system. Part of speech tagging is often used as a prerequisite for more complex NLP applications such as information extraction, syntactic parsing, machine translation or semantic field annotation etc. Afaan Oromo POS tagging is a method of assigning a specific Afaan Oromo part of speech tag to each word in Afaan Oromo sentence to disambiguate the function of that word in the specific context.

In this chapter, a detail description of design issues and techniques of the Afaan Oromo POS tagger are dealt with. Moreover, the design of the Brill transformation based error driven learning for Afaan Oromo language is discussed. Some of the modification on the initial state tagger and rule template of the tagger are also mentioned. Next the generalized learning algorithm for TEL of Afaan Oromo POS tagger is explored. Finally, the architecture of the modified Brill tagger is presented.

## 4.1. Approaches and Techniques

Part of speech tagging is the process of assigning part of speech labels to a sequence of words in a sentence. This problem can possibly be tackled using different approaches. Such approaches are rule based and statistical which have their own pros and cons. As far as part of speech tagger with possibly a higher performance of tagging is desired, there is a need to take the advantages from two or more different approaches thereby remedying the shortcomings of the approaches.

The statistical approach as its name implies extracts the statistical properties of words in the training phase to label words with their correct part of speech. One of the most widely used models in the statistical approach for part of speech tagging is the Hidden Markov Model. The Brill tagger is a transformational error driven approach that uses the hybrid of the statistical and rule based approach for tagging.

As it is mentioned in Chapter One, this work is an extension of the work done in [9], which uses TEL for Afaan Oromo Tagger. The researcher has tried to customize the original Brill tagger for

Afaan Oromo with a bit modification. Even though the performance of the modified Brill tagger is better than the default, it has also got varies drawbacks. Most of the words are incorrectly assigned to a single tag (noun), the initial state tagger is assign for untagged Afaan Oromo texts. Moreover, the transformational rules were trained on a very small training corpus that lacks knowledge to generalize and perform proper change of tags based on the learnt rule.

Thus, this research is designed to mitigate the limitation of the work done in [9] by doing the following amendments that the researchers believe will enhance the performance of the TEL POS for Afaan Oromo. The first one is to use sufficiently large corpus to train the transformation rules so as to capture detail knowledge of tag transformation of the language. The second is to replace the initial state annotator in the Brill tagger with HMM based POS tagger. This would have the following impacts that improve the Brill tagger lexical and transformational rules for Afaan Oromo.

1. The initial state annotator almost will have the appropriate tag of each lexicon in the given corpus and hence will improve performance as it minimizes the wrongly assigned initial tags to words.
2. The transformation rule requires less knowledge to make corrective actions and hence the required knowledge would easily be captured from the corpus.

## 4.2. Designing Transformation-based Error-Driven learning

In 1992, Eric Brill introduced a POS tagger that was based on rules or transformations as he calls them. The tagger is an extension of a rule based approach. The Brill's transformation based learning is the framework of Transformation-based Error-driven Learning (TEL) [27]. The name reflects the fact that the tagger is based on transformations or rules, and learns by detecting errors.

Brill's rule based tagger learning also known as transformation based error driven learning is a framework that is based on rules that can be learned by detecting errors occurred in the previous steps. It has two phases: the initial state tagger and the learning phase. The TEL tagger takes untagged corpus as an input and the initial state tagger assigns the likely tag for the words in the untagged corpus which then results in a new and temporary corpus as an output. The learning phase takes two input data namely the temporary corpus tagged by the initial state tagger and the goal corpus, manually tagged corpus assumed to be correct, which is used for comparison against

the temporary corpus during rule derivation. The temporary corpus passes through the learner iteratively to derive Brill transformations. In each iteration, the learner derives a new rule, afterwards a comparison with goal corpus is done and the rule that improves the annotation is considered as a Brill transformation and the temporary corpus is updated. The learning phase continues until no rules that can improve the tagging of temporary corpus (through comparison with the goal corpus) can be derived. By this process, the learner produces an ordered list of rules which can be applied for tagging untagged texts.

The learning phase of the Brill tagger has two sub-phases: the lexical rule learner, as its name implies, it derives lexical rules which are used for tagging unknown words and the contextual rule learner, learns the context of a word in a sentence and derives contextual rules which are used for the improvement of the accuracy of the tagger. Hence, the TEL is used twice in the Brill's tagger learning phase: in the lexical rule learner and in the contextual rule learner. The lexical and contextual rule learners use two corpora, the goal corpus and the temporary corpus. Then the goal of the tagger is to change the tags of the temporary corpus step by step in the learning phase to make it similar with the goal corpus as much as possible. The framework of the original Brill TEL tagger is shown in Figure 4.1.

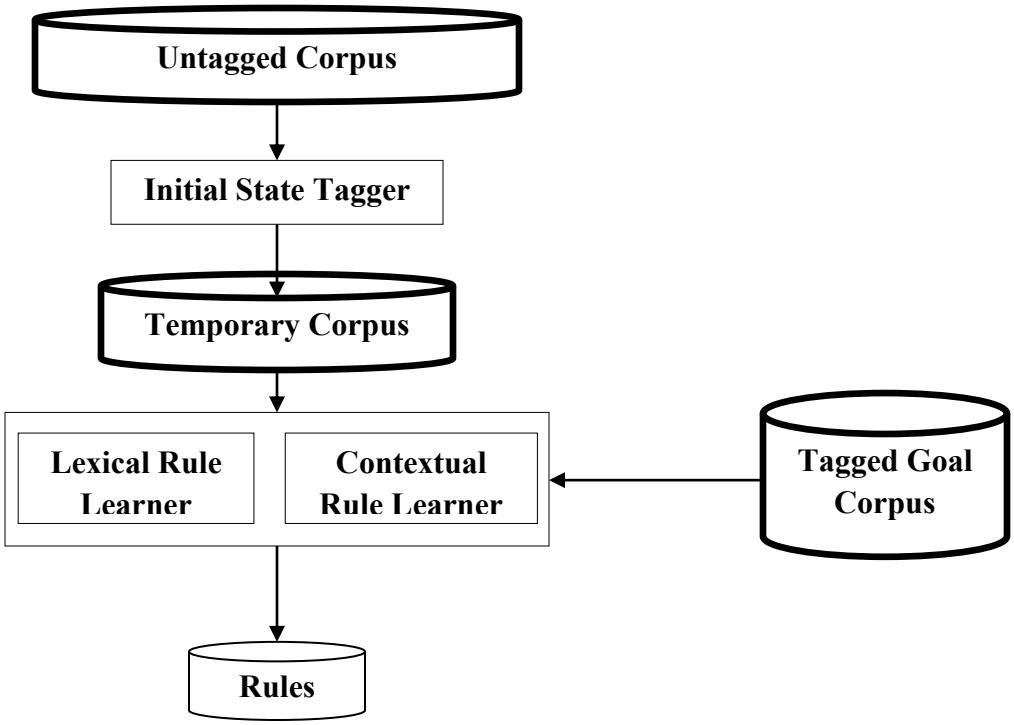


***Figure 4.1: Original Brill's Transformational Error-driven learning***

The TEL begins with an un-annotated text as input which passes through the initial state annotator'. The initial state tagger component takes untagged corpus as an input and tags this untagged corpus in some fashion. The initial state tagger can be chosen to be n-gram taggers (like unigram, bigram or trigram), default tagger, and/or sophisticated taggers etc. The Original Brill's tagger uses Default tagger, which assigns a specific open class tag (noun in most cases) for all words in the corpus. The choice of the tagger depends on the final performance of the overall Brill TEL tagger.

In this thesis, HMM tagger is used as initial state tagger. HMM tagger assigns the most probable part of speech tag for each word by calculating the word emission probability and tag transition probability. The framework of the adapted Brill TEL tagger for this work is shown in Figure 4.2.

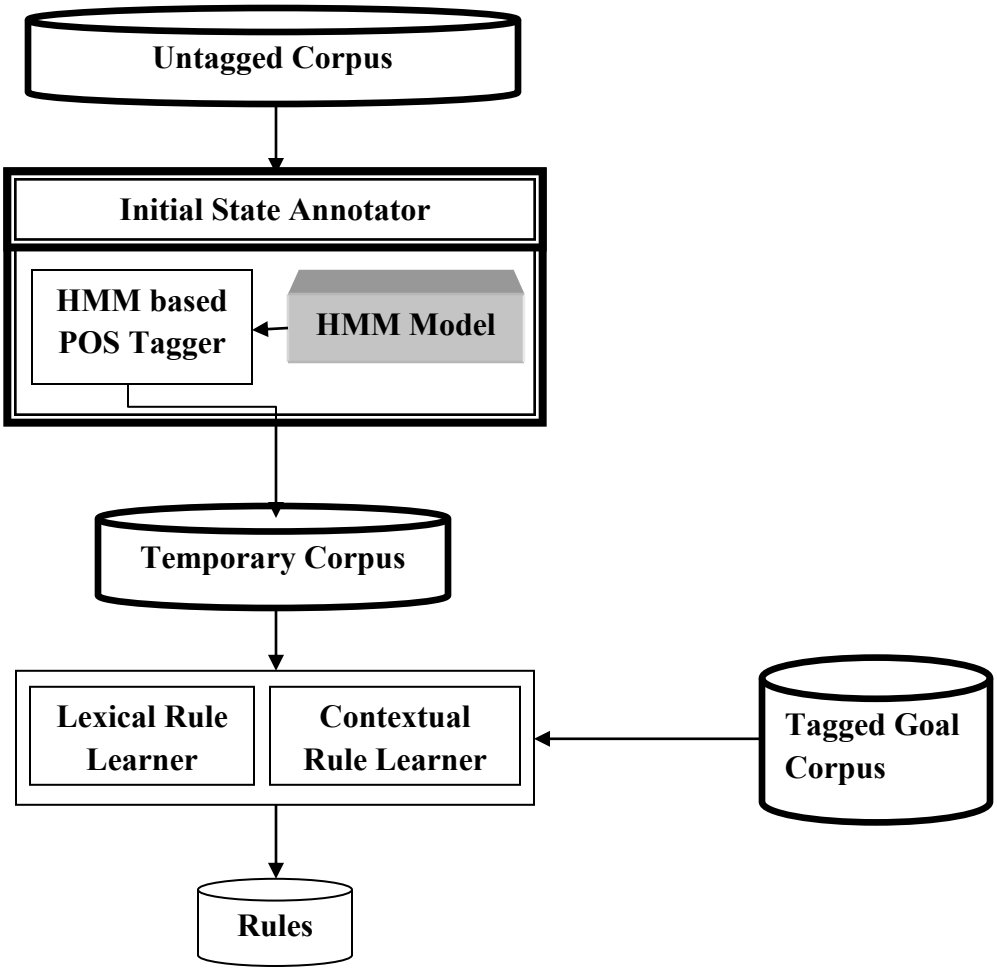


***Figure 4.2: Adapted Brill's Transformational Error-driven learning***

HMM is the statistical model which is mostly 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 [8]. 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. 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).

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:

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

Where "word" is words in a sentence, "tag" is the POS tag for particular word

The optimal sequence of part of speech tags for a given sequence of words in an input sentence to be tagged can be found using the Viterbi algorithm [1, 15]. The Viterbi algorithm is a dynamic programming algorithm that finds the optimal path in the tagging process. It reduces the complexity of the HMM core issue, finding the best part of speech tag sequence for a given sequence of words in the input sentence, to polynomial time and the algorithm is linear in the number of words to be tagged [14]. In simple terms, the Viterbi algorithm calculates the probability of all possible paths of the word tag pairs in the input sentence. Afterwards, it will select the path of the word tag pair with the highest probability to be the best path [1, 13]. It uses

the lexical and contextual probabilities obtained from the lexical and contextual model to find the best path.

The output of the initial state annotators is a temporary corpus (i.e. unannotated corpus that is tagged with the initial stated tagger), which is then compared to a goal corpus (i.e. manually annotated training corpus). For each time the temporary corpus is passed through the learner, the learner produces one new rule, the single rule that improves the annotation the most compared with the goal corpus, and replaces the temporary corpus with the analysis that results when this rule is applied to it. By this process the learner produces an ordered list of rules and saves them for the next process.

### 4.2.1. Rules

A rule is one component of the Brill TEL tagger that consists of two parts: a condition (the trigger and possibly a current tag), and a resulting tag. The rules are instantiated from a set of predefined transformation templates. The rule can be categorized as lexical rule or contextual rule depending on the information content of the rule. Whether they are lexical or contextual rules, they contain uninstantiated variables and are of the form:

If Trigger, then change the tag X to the tag Y

Or

If Trigger, then change the tag to the tag Y *where X and Y are variables*

The interpretation of the first type of the transformation template is that if the rule triggers on a word with current tag X then the rule replaces current tag with resulting tag Y. The second one means that if the rule triggers on a word (regardless of the current tag) then the rule tags this word with resulting tag Y. The set of all permissible rules (PR) are generated from all possible instantiations of all predefined templates. The set of permissible rules are generated during the learning process by the two main components of the learning phase, the Lexical rule learner and the contextual rule learner. The Brill tagger rule component stores the rules, Brill transformation Templates from the learning phase. Putting it altogether, the rule component is the component that handles the output of the learning phase namely the contextual rule and lexical rule learners.

### 4.2.2. Learning Phase

The Brill tagger learning phase, as can be seen also from Figure 4.3, has two sub components namely the lexical rule learner and contextual rule learner. A brief description of these subcomponents is given in the following sub sections.

#### 4.2.2.1. The Lexical Rule Learner

The goal of the lexical rule learner is to derive a set of all permissible rules that can produce the most likely tag for any word in the given input text of a specific language, i.e. the most frequent tag for the word in question considering all texts in that language. The problem is to determine the most likely tags for unknown words, given the most likely tag for each word in a comparatively small set of words.

The lexical rule learner component uses statistical methods to find the most likely tag of a word.The rule generating process takes an initially tagged temporary corpus (TC0) which can be tagged by the initial state tagger, and finds the rule in PR which gets the best score when applied to TC0. A best score for a rule means that the temporary corpus produced when applying the rule gives an annotation closer to the goal corpus. And this rule can be called as R1. Then R1 is applied to TC0, producing TC1. The process is now repeated with TC1, i.e. it finds the rule in PR which gets the best score when applied to TC1. This will be rule R2 which then is applied to TC1 producing TC2. The process is done iteratively producing rules R3, R4, etc and corresponding temporary corpora TC3, TC4, etc until the score of the best rule fails to reach above some predetermined threshold value or until no rule can further improve the tagging of the corpus. The sequence of temporary corpora can be thought of as successive improvements closer and closer to the goal corpus. The output of the lexical rule learner is the ordered list of rules R1, R2 ... which are used for tagging new unannotated texts.

The score for a rule R in PR is computed as follows: for each tagged word in TCi the rule R gets a score for that word by comparing the change from the current tag to the resulting tag with the corresponding tag of the word in the goal corpus. Depending on the effect of the rule on the text to be tagged, the score of the rule R may be positive, negative or zero. A positive score means that the rule improves the tagging of this word, and a negative score means that the rule worsens the tagging. If the condition of the rule is not satisfied then the score is zero. The total score for

R, score(R), is then obtained by adding the scores for each word in TCi for that rule. When the total score for each R is obtained the rule which has the highest score is added to the set of rules which have already been learned. Rules are ordered, i.e. the last rule is dependent on the outcome of the earlier rules.

The lexical rule learner deals with the morphology of the language in order to drive the set of all permissible rules. Some of the transformation templates that are used in lexical rule learner are given below.

1. Change the most likely tag to Y if the current word has suffix/prefix X
2. Change the most likely tag to Y if deleting /adding the suffix x, |x|<4, results in word, |x| is length of x.
3. Change the most likely tag from X to Y if deleting/adding the prefix x, |x|<4, results in word, |x| is length of x.
4. Change the most likely tag from X to Y if word W ever appears immediately to the left/right of the word.
5. Change the most likely tag to Y if the character Z appears anywhere in the word.

Template numbers 2 and 3 are for the original Brill tagger which imply adding/deleting of prefix/suffix of only up to 4 characters was considered. In adapting Brill tagger for Afaan Oromo in the work of [2], after assessing the nature of Afaan Oromo words, the same trend as that of Brill templates 2 and 3 is found. It is possible to consider the following example.

- *Taphataniiru* "They have played"= tapha- **taniiru** has seven suffix

But since the transliterated version of the texts is used, templates 2 and 3 are changed to be up to 8 suffixes.The rules are changed to the following format.

- Change the most likely tag to Y if deleting /adding the suffix x, |x|<8, results in word, |x| is length of x.
- Change the most likely tag from X to Y if deleting/adding the prefix x, |x|<8, results in word, |x| is length of x.

#### 4.2.1.2. The Contextual Rule Learner

Once the tagger has learned the most likely tag for each word found in the tagged training corpus and the rules for predicting the most likely tag for unknown words, contextual rules are learned for disambiguation and better accuracy. The contextual rule learner finds rules on the basis of the particular environments i.e. the context of words.

In order the contextual rule learner to generate rules, it needs the goal corpus and the initial temporary corpus TC0 as an input. The initial state tagger takes untagged text and lexicons and the lexical rules obtained during lexical rule training process.

First, the learner generates the set of all permissible rules PR from all possible instantiations of all the predefined contextual templates. After it generates the contextual rules, it computes the score of each rule for a particular word. Then the learner can pick the rule R1 with the highest score and put on the rules component as an output. Afterwards the learner can take R1 and apply on TC0 to get TC1, on which the learning continues. The process is done iteratively putting one rule, the rule with the highest score in each iteration, on the rule component as an output in each iteration until no rule achieves a score higher than some predetermined threshold value or until no rule can further improve the tagging of the corpus.

Besides, the score of every Rule R is computed. Let R be a rule in PR, the score for R in TCi can be calculated as follows: for each word W in TCi, the contextual rule learner computes the score for R on this word W. Then, the scores for all words in TCi where the rule is applicable are added and the result is the total score for R. This can be done by comparing the tags of words in the tags of words in the TCi with the correct tags of words in the goal corpus. If R is applied to the word W, thereby correcting an error, the score for W is +1. If applying R to the word W introduces an error then the score for W is -1. In all other cases, the score for the particular word W is 0. Therefore the total score for R is computed as follows.

**Score(R) = numberoferrors corrected –number of errors introduced.**

Generally speaking, the goal of the contextual rule learner, in a similar way to that of lexical rule leaner, is to generate set of all permissible rules. These sets of rules in the contextual rule learner are totally different from the lexical rule learner for it uses different transformation templates. The trigger in this case, unlike the lexical rule learner which depends on the morphology of the

word, depends on the context (environment) of that word. Some of the triggers of the templates are listed below:

1. The preceding/following word is tagged with X.
2. One of the two preceding/following words is tagged with X.
3. One of the three preceding/following words is tagged with X.
4. The preceding word is tagged with X and the following word is tagged with Y.
5. The preceding/following two words are tagged with X and Y.

### 4.2.2. Brill Tagger Architecture

During the training, the Brill tagger learns rules, both lexical and contextual rules, which are used for tagging new untagged text. To tag a new text, the Brill tagger takes the rules that it has learned in the learning phase of the training as well as the text to be tagged as input. The rules are applied on the new untagged text and the tagger gives the new tagged text as an output.

The Brill tagger is given Afaan Oromo untagged text as an input then it tags this text using the rules that it has learned during the learning phase. It first tags the text using the lexical rules. Since the lexical rules do not deal with contexts (environments of the word), the contextual rules are applied to look into the contexts of words in the tagged text and improve the performance of the tagger. Finally a tagged Afaan Oromo text is given as an output of the tagger.

# CHAPTER FIVE

# IMPLEMENTATION OF AFAAN OROMO PART OF SPEECH TAGGER

## 5.1. Introduction

This chapter deals with the detail explanation of implementation of the designed Brill's tagger architecture and corpus preparation for Afaan Oromo language. The Brill's rule based tagger is customized and adopted for Afaan Oromo language. The NLTK with Python was also used to test Afaan Oromo corpus on HMM tagger and also to check the choice of an initial state tagger for the Brill's tagger. The rationale behind the choice of these two tools is that NLTK supports many tasks for part of speech tagging. It is also simple and open source used for different tasks of natural language processing. The Brill's tagger tool is also used for training and testing of the corpus. An incremental approach is used to prepare the corpus for Afaan Oromo language.

## 5.2. Corpus Preparation

The corpus is a fundamental tool for any type of research on natural language processing. The availability of computers in the 1950's immediately led to the creation of corpora in electronic form that could be searched automatically for a variety of language features and compute frequency, distributional characteristics, and other descriptive statistics [11]. Corpus, plural corpora, is a collection of text. It can be a flat text i.e. a text with no additional linguistic information or a text whereby each word in the text is attached with linguistic information .

The corpus with additional linguistic information can be called as annotated/tagged corpus. Such linguistic information in the annotated corpus can be part of speech information, sentiment information that specify the word's word class category and sentiment category respectively. The annotated corpus can be used in many NLP applications like part of speech tagger training and testing, parsing, sentiment analysis etc. In this thesis work, the annotated corpus used is considered to be a text tagged with the corresponding part of speech tags.

The first phase of corpus creation is data capture, which involves collecting a text data from different sources of information. The corpus includes different types of text from multiple source information to make it balanced corpus. However, a category specific corpus contains

words that are mostly used in that category and if a text from other category to be tagged is given to the tagger trained on this corpus, the performance of the tagger may be degraded.

The essence of developing a balanced corpus is, in fact, to increase the performance of the tagger when it tags any text taken from any category which implies directly that balanced corpus contains as many words as possible from different categories in their appropriate sense. Larger size of the corpus provides greater learner tendency for the system. The numbers of unknown words are decreased, which results in increasing the accuracy of the system.

In Afaan Oromo, there is no such large corpus prepared yet. Therefore, an incremental approach is used for developing a balanced corpus. The Afaan Oromo texts are collected from Afaan Oromo news, newspapers, journals, books, fictions and publications. About 258 sentences including a total of 1708 words tagged corpus was also taken from the previous work of [8] and [9]. The incremental approach is started with training the Brill's tagger with this already existing tagged corpus. Then, the collected raw text is tagged with the tagger. Afterwards, the tagged text is given to language experts for correction and approval. The tagged corpus used for training is updated to contain the new correctly tagged and approved text which is in turn used for training the tagger. This process is repeated until the desired size of the corpus for this thesis work is achieved. In this case, a corpus of 1100 sentences containing about 17473 words are prepared and used. A sample corpus is shown in appendix A.

## 5.3. Implementation of the Brill's Tagger

There are three main phases in implementing a Brill tagger for any language. These are (i) training phase – in which it first extracts rules from the training corpus using statistical techniques. (ii) Verification phase – in which these rules are verified by taking an annotated text with its tags removed as the input and generates the tagged text; this tagged text is compared with its original tagged text and learns where it has gone wrong; (iii) Testing phase – in which new unseen texts can be tagged and cross check with the reference text (i.e. manually tagged version of the test corpus) for performance evaluation.

A Brill's tagger has two main phases, which are initial state tagger and the learning phase. The initial stage annotator takes untagged Afaan Oromo corpus as an input and gives the corresponding tagged corpus. The output of the initial state tagger is an input for the learning

phase for rule generation. Finally, Brill's tagger gives a lexical and contextual tagging rules of the language.

### 5.3.1. Implementation of the Initial State Tagger (HMM Tagger)

The implementation of Brill's tagger starts at the choice of the initial state tagger. Brill's tagger takes unannotated corpus as input, which passes through initial state annotator. The Original Brill's tagger uses Default tagger, which assigns a specific open class tag (noun in most cases) for all words in the corpus. The choice of initial state annotator affects the overall accuracy and performance of the Brill's tagger.

In this study, an HMM tagger is preferred to be used as the initial state tagger for the adopted Brill's tagger. The HMM tagger is a statistical approach of POS tagger, which assigns the most probable part of speech tag for each word by calculating the word emission probability and tag transition probability. The HMM tagger uses HMM model for tagging the raw Afaan Oromo text. The Viterbi algorithm is implemented for finding the optimal path in the HMM tagger. Then, the output of the HMM tagger is given to the Brill's tagger learning phase.

### 5.3.2. Implementation of the Brill's Tagger Learning Phase

The learning phase takes a tagged corpus by the initial state tagger to generate rules. The learning algorithm of transformation-based tagging selects the best transformations and determines their order of application. It goes through a loop of iteration to generate rule. In each iteration of the loop, we choose the transformation that reduces the error rate most, where the error is measured as the number of words that are miss-tagged in tagged corpus. It stops when there is no transformation left that reduces the error rate by more than a pre specified threshold (until applying new rules leaves the text in the same state, which is then supposed to be the final state of the tagging). This procedure is a greedy search for the optimal sequence of transformations.

Initially, we tag each word with the HMM initial state tagger. Then the learning phase is used to generate the lexical and transformational rules. These rules are later used for tagging. This can be summarized by the following steps.

- Initialization:
  - ✓ HMM tagger using Viterbi algorithm, assign the most optimal part of speech tag sequence for a given word sequence.
  - ✓ Learning or guessing rules with lexical rules on the same basis as contextual rules
- Learning Phase
  - ✓ Iteratively compute the error score of each candidate rule (difference between the number of errors before and after applying the rule)
  - ✓ Select the best (higher score) rule.
  - ✓ Add it to the rule set and apply it to the text.
  - ✓ Repeat until no rule has a score above a given threshold (that is, if the chosen threshold is zero (which can lead to over-fitting), until applying new rules leaves the text in the same state, which is then supposed to be the final state of the tagging).

# CHAPTER SIX
# EXPERIMENTION AND PERFORMANCE ANALYSIS

## 5.1. Introduction

Different experiments have been conducted on Afaan Oromo part of speech tagger. The corpus is divided into two sets: the training set and the testing set. The former one comprises 80% of the corpus while the remaining 20% is used for testing purpose. In this Chapter, the detail experiments conducted for this thesis work are discussed briefly.

## 5.2. Experiments

The Brill tagger with modifications is used for conducting experiments in the rule based tagger. Ten different experiments are conducted on the Brill's tagger using different sizes of the training set and different initial state annotators. The experiment starts from the first 10% of the training corpus, repeatedly adding 10% of the corpus until the entire corpus is used. Table 6.1 and Figure 6.1 show the different experiments conducted using different portions of the training set with the corresponding performance of the rule based tagger for the different initial state annotators.

***Table 6.1 Brill's Tagger performance using different initial state taggers***

| **Initial State Tagger** | | **Size of the Training set** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
| *Performance per different initial state taggers (%)* | Default Tagger | 54.2 | 59.4 | 60.3 | 64.2 | 67 | 72.1 | 76.5 | 83.2 | 84 | 89.6 |
| | HMM Tagger | 71.4 | 72.9 | 74.6 | 75.8 | 79.1 | 79.9 | 81.7 | 87.0 | 92.4 | 93.35 |

The default tagger assigns a specific part of speech tag for each word. In this work, when it takes the default tagger as an initial state annotator, the Noun (NN) and Proper noun (NNP) if capitalized part of speech tagger is selected to be default tag. The HMM tagger assigns the most optimum tag sequence given the word sequence. A significantly higher performance is achieved when the HMM tagger is used as the initial stated tagger, which implies that HMM tagger simplifies the learning work of the Brills training as well as the accuracy of the rule generated. The following diagram shows the learning curve during the training of the Brill's tagger using the HMM tagger as the initial state tagger.

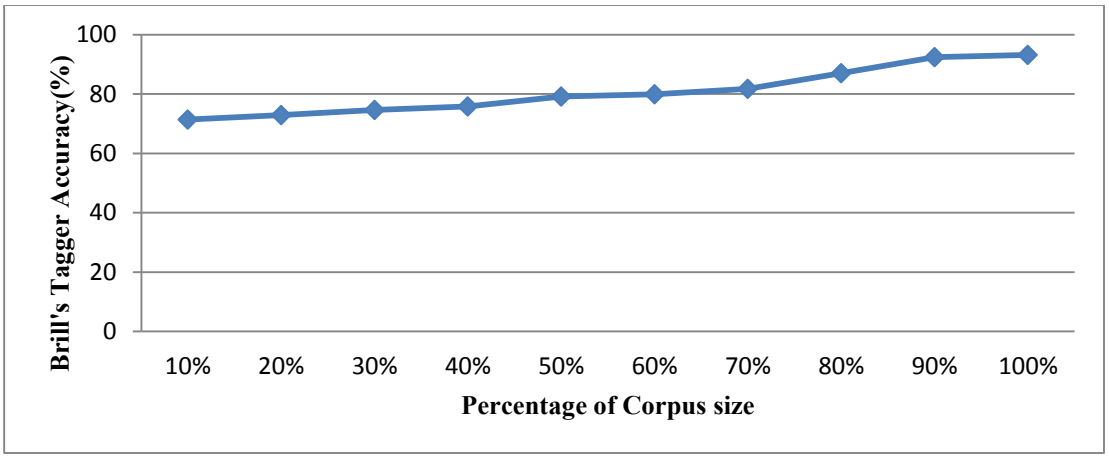


*Figure 6.1 Learning curve of the Tagger*

### 5.2.1. Brill's Tagger Versus Corpus Size

The tagger is also checked on the size of training corpus used. The size depends on the number of words in the corpus. Accordingly, it is shown that the size of the corpus used for Brill's tagger has a significant effect on the accuracy of the tagger. Figure 6.2 shows the increasing Brill's tagger accuracy with the increase in the size of Afaan Oromo corpus.

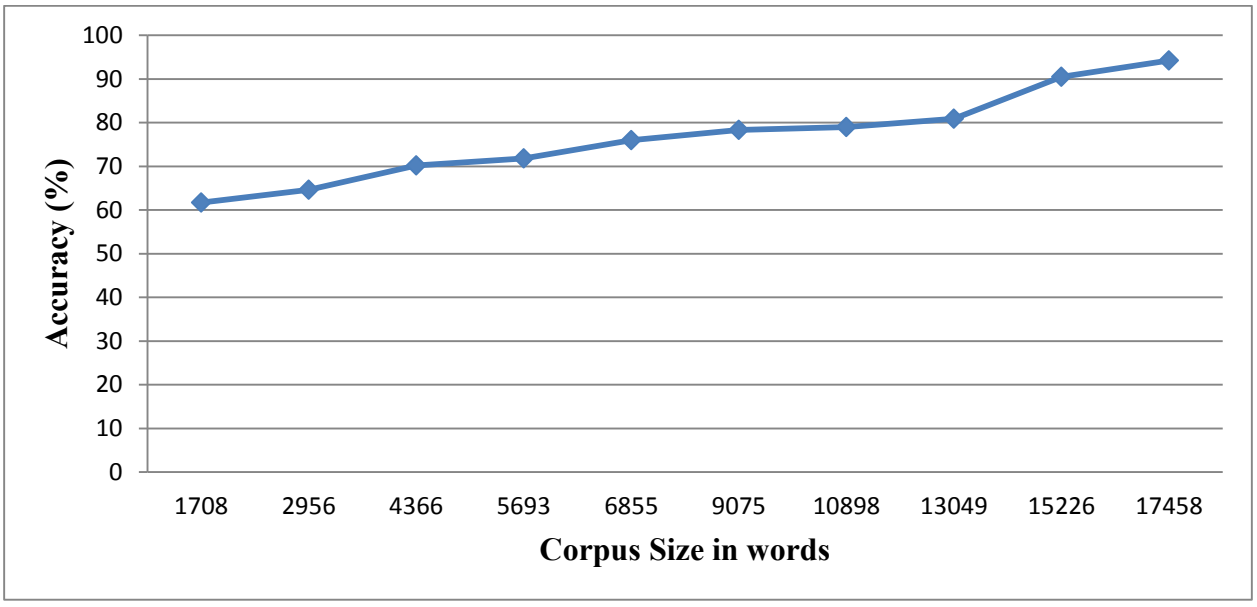


*Figure 6.2 Brill's tagger versus Corpus size*

## 5.3. Performance Analysis

In order to analyze the performance of the Brill tagger for the different part of speech tags, the frequency of the taggers in the entire corpus, training set and testing set is considered. Moreover, confusion matrix Table 6.3 is developed for the Afaan Oromo POS tagger. A total of 26 tags are identified in this research work and based on their frequency, they are divided into two groups namely the 10 most frequent tags and the rest as others. The frequency of the tags is given in Table 6.2.

***Table 6.2 Part of Speech Tags Frequency***

| **Tags** | **Entire Corpus Frequency** | **Training Corpus Frequency** | **Testing Corpus** | |
|---|---|---|---|---|
| | | | **Frequency** | **%** |
| NN | 5424 | 4375 | 1049 | 19.34 |
| VV | 3460 | 2557 | 903 | 26.1 |
| JJ | 1366 | 1184 | 182 | 13.32 |
| PUNC | 1229 | 1116 | 113 | 9.19 |
| PR | 1034 | 811 | 223 | 21.57 |
| PP | 908 | 712 | 196 | 21.59 |
| AX | 819 | 597 | 222 | 27.11 |
| CC | 448 | 271 | 177 | 39.51 |
| AD | 420 | 347 | 73 | 17.38 |
| NNP | 229 | 195 | 34 | 14.85 |
| Others | 2136 | 1595 | 541 | 25.33 |
| **Total** | **17473** | **13760** | **3713** | **21.25** |

***Table 6.3 Brill's Tagger Confusion Matrix using HMM as initial state tagger***

| | | Test tags (Predicted Tags) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference tags (Desired tags) | | NN | VV | JJ | PUNC | PR | PP | AX | CC | AD | NNP | Others | Total | Performance (%) |
| | NN | **975** | 22 | 1 | 10 | 16 | 16 | 8 | 3 | | | | 1051 | **92.8** |
| | VV | 34 | **872** | | 6 | 1 | 2 | 2 | | | | | 917 | **95.1** |
| | JJ | 14 | 1 | **180** | | | | | | | | | 195 | **92.3** |
| | PUNC | | | | **93** | 6 | | 1 | | 4 | | | 103 | **90.3** |
| | PR | 10 | 5 | | | **207** | | | 1 | | | | 223 | **92.8** |
| | PP | 5 | 6 | 7 | | | **175** | | | | | | 193 | **90.7** |
| | AX | | 1 | | | 6 | | **207** | | | 8 | | 222 | **93.2** |
| | CC | 1 | | 1 | | | | | **171** | | 4 | | 177 | **96.6** |
| | AD | 5 | 1 | | 1 | | | 1 | | **65** | | | 73 | **89.0** |
| | NNP | 2 | | | | | | | | | **32** | | 34 | **94.1** |
| | Others | 19 | | | | | | | | | | **522** | 541 | **96.5** |
| | Total | 1065 | 908 | 189 | 110 | 235 | 193 | 219 | 175 | 69 | 44 | 522 | **3729** | **93.0** |

The Brill's tagger confusion matrix using HMM tagger as the initial state tagger shows that it assigns 3729 tags correctly and 136 tags wrongly to the tokens in the testing set. The performance of the rule based tagger varies for the different part of speech tags with a higher performance for CC part of speech tag followed by others, VV, NNP, AX, NN, PR, JJ, PP, PUNC, AD of speech tags for the given testing set trained on the training set.

## 6.4. Discussion

Different experiments are conducted for the Afaan Oromo Brill's tagger. Comparison with the Brill's tagger developed for Afaan Oromo in the work of [9] is done. Accordingly, different performance is obtained: the improved Brill's tagger performed better than previously adapted Brill's tagger. The performance of the original Brill's tagger and Improved Brill's tagger is 89.8% and 95.6% respectively, which results with the difference of 5.8%.

This performance improvement is made because of the improvement on the size of the training and testing corpus, the choice of HMM tagger as initial state tagger and the rule generating system in the lexical rule learner. The performance increment with the size of corpus is shown in Table 6.1 and Figure 6.2 above. Comparison from Table 6.1 also showed that adding more corpus without modifications of the previous tagger (i.e. using Default Tagger as initial state tagger) has significant improvement on the whole performance of the tagger. But the adapted Brill's tagger has higher performance compared to the original Brill's tagger.

In general, a 10 fold validation system is used to evaluate the accuracy of the tagger. This is done by dividing the entire corpus randomly into ten parts. The nine fold is used for training and the remaining tenth fold is used for testing the tagger that was trained on the previous nine folds. The process was repeated ten times by taking the other nine as training and the tenth one as testing corpus.

A performance comparison for each part of speech tagger for the previously adapted Brill's Tagger and Improved models is given in Table 6.4 to see the performance improvement through making improving the Brill's Tagger for Afaan Oromo Language. The Comparison is made with the 10 fold validation system.

*Table 6.4 Comparison of Original Brill's Tagger [9] and Improved Brill's tagger*

| S/No | No of words | Original Brill's Tagger | Improved Brill's Tagger Accuracy (%) |
|---|---|---|---|
| 1 | 2450 | 92.4 | 97.4 |
| 2 | 2406 | 91.6 | 97.6 |
| 3 | 2381 | 94.9 | 98.9 |
| 4 | 2112 | 91.2 | 96.7 |
| 5 | 1850 | 92.8 | 93.4 |
| 6 | 1297 | 86 | 92.6 |
| 7 | 1587 | 88.5 | 94.9 |
| 8 | 1310 | 84.5 | 93.6 |
| 9 | 1091 | 91.5 | 94.8 |
| 10 | 997 | 84.8 | 96.1 |
| **Average Accuracy** | | **89.8** | **95.6** |

Previously, the accuracy of HMM Afaan Oromo Tagger is 87.58% for Unigram and 91.97% for Bigram for the work of [8]. The Afaan Oromo Brill's tagger has got 80.08% accuracy from the work of [9]. In this work, the original Brill tagger is with average of 89.8% accuracy while the improved Brill's tagger showed 95.6% accuracy with 5.8% higher. Both are tested on the improved size of the dataset.

# CHAPTER SEVEN

# CONCLUSION AND RECOMMENDATION

## 7.1. Conclusion

The ultimate goal of research on Natural Language Processing is to parse and understand language, which is not fully achieved yet. For this reason, research in NLP has focused on intermediate tasks that make sense of some of the structure inherent in a language without requiring complete understanding. One such task is part-of-speech tagging, or simply tagging. Moreover, part of speech tagging can be conceived as the problem of assigning part of speech tags to a word in a sentence.

In this work, Brill's tagger is designed and adapted with identified possible improvements for Afaan Oromo language. A balanced corpus with a total of 1100 sentences is collected from different source of information, from which 258 are taken from previous work. 26 part of speech tags are identified as tag sets that are used in annotating these total words to create an annotated corpus for training the Brill tagger as a supervised learning approach is used. An incremental approach is used to create a tagged corpus. The tag sets identified does not indicate number, gender, tenses etc.

With the increment on the size of training corpus, the accuracy of the tagger increases. This is shown with the choice of the initial state tagger, which has a significant effect on the accuracy of the tagger. Accordingly, HMM tagger is chosen to be the one with best performance. This implies that using HMM tagger as an initial state tagger increases the accuracy of the rules generated during the learning phase of the Brill's tagger.

The comparison of the improved Bill's tagger is made with the Original Brill's tagger with 10 fold validation system. Accordingly, the overall accuracy for Original Brill's tagger is 89.8% while the improved Brill's tagger is 95.6%.

## 7.2. Recommendation

There are lots of research areas in natural language processing that can be done for different languages in Ethiopia. The same thing holds true for Afaan Oromo language. Therefore, to assist researchers, it will be of great paramount if a standard corpus for Afaan Oromo language is developed that will be available for NLP researchers in Afaan Oromo language.

Finally, this research work suggests the following items as future works:

- Using morphologically analyzed corpus for training of Brill's tagger's to consider the inflectional properties of the language.
- Comparison of two hybrid approaches: the hybrid of rule based and HMM tagger and the hybrid of rule based and ANN for Afaan Oromo language
- Extending this work by training in using tagsets that can identify gender, number, tense etc with different feature set
- Conducting similar researches for other local languages by adapting this work.

## References


[1] Christopher D. Manning Hinrich Schutze. Foundations of Statistical Natural Language Processing, 2nd Ed. The MIT Press Cambridge, Massachusetts London, England, 2000.

[2] Tarveer S. Natural Language Processing and Information Retrieval. Published by Oxford University press in Indian Institute of Technology, Allahabad, India, 2008.

[3] Brill, E. A simple rule-based part of speech tagger. Department of Computer Science, University of Pennsylvania, Philadelphia, Pennsylvania, U.S.A, 1995.

[4] Megyesi B. Brill's POS Tagger with Extended Lexical Templates for Hungarian. Master's Thesis, Department of Linguistics, Computational Linguistics, Stockholm University, Stockholm, Sweden, 1999.

[5] Abdulsamad M. 'Seerlugaa Afaan Oromoo'. Bole Printing Enterprise, Addis Ababa, Ethiopia.1997.

[6] Mohammed S. & Pedersen T. Guaranteed Pre Tagging for the Brill Tagger. University of Minnesota, Duluth, USA.

[7] Gamta Tilahun. "Forms of Subject and Object in Afaan Oromo", Journal of Oromo Volume 8 Number 1&2, July 2001.

[8] Getachew Mamo. Part-of-Speech Tagging for Afaan Oromo Language. Master's Thesis, Addis Ababa University, 2009.

[9] Mohammed-Hussen. Part Of Speech Tagger for Afaan Oromo Language using Transformational error driven learning (TEL) approach. Master's thesis, Addis Ababa University, 2010.

[10] Robin. Natural Language Processing. Article on Natural Language Processing. Published on December 16th, 2009.

[11] Wolfgang Teubert. Corpus Linguistics and Lexicography, John Benjamin's publishing Co. International Journal of Corpus Linguistics Volume 6, 2001, 125-153

[12] Fahim Muhammad Hasan, Naushad UzZaman, Mumit Khan, Comparison of Different POS Tagging Techniques (n-grams, HMM and Brill's Tagger) for Bangla, International Conference on Systems, Computing Sciences and Software Engineering (SCS2 06) of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 06), pp: 4-14.

[13] Hall, Johan. A Probabilistic Part-of-Speech Tagger with Suffix Probabilities. Master's Thesis, School of Mathematics and Systems Engineering, Växjö University, 2003.

[14] Blunsom Ph. Hidden Markov Models: pcbl@cs.mu.oz.au, accessed on August 19, 2004

[15] KhineZin, (2009). Hidden Markov model with rule based approach for part of speech tagging of Myanmar language, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.

[16] Brants, T. TnT - a statistical part-of-speech tagger. In Proceedings of the 6th Conference on Applied Natural Language Processing, Seattle, Wash, 29 April–4 May 2000, pp.224–231.

[17] Gerold S and Martin Volk. Adding Manual Constraints and Lexical Look-up to a Brill Tagger for German, Computational Linguistics Group, Department of Computer Science, University of Zurich, 2000.

[18] Schmid, H. Part-of-speech tagging with neural networks. In Proceedings of COLING-94, Kyoto, 1994.

[19] Nuno C. & Gabriel Pereira, Neural Networks, Part of Speech Tagging and Lexicon. Technical Report DI-FCT/UN, University Nova de Lisboa– Faculty of Technology, Department of Informatics, Portugal, 1998.

[20] Teklay Gebregzabihe. Part of Speech Tagger for Tigrigna Language.Master's thesis, Addis [21] Solomon Asres, (2008). Automatic Amharic Part-of-Speech Tagging Using Hybrid Approach (Neural Network and Rule-Based). Master's thesis, Addis Ababa University.

[22] Megyesi B. 1998. Improving Brill's POS Tagger for an Agglutinative Language. Thesis in Computational Linguistics, Department of Linguistics. Stockholm University, Sweden.

[23] Petasis G, Paliouras G, Vangelis, Karkaletsis D and Androutsopoulos, Resolving Part-of-Speech Ambiguity in the Greek Language using Learning Techniques, Institute of Informatics and Telecommunications, N.C.S.R, "Demokritos", 2002.

[24] FDRE Population census Commission, Summary and Statistical report of the 2007 population and housing census.Printed by United Nations Population Fund (UNFPA) Addis Ababa, December 2008.

[25] Tilahun Gamta. Qube Afaan Oromo: Reasons for Choosing the Latin Script for Developing an Oromo Alphabet. Published on the Journal of Oromo studies Volume I Number I Summer 1993.

[26] Http://www.Wepeadia.com/ Wiki: Oromo language (1/3), visited on Aug 3 2010.

[27] Brill E and Marcus M. 1992. Tagging an Unfamiliar Text with Minimal Human Supervision. In Proceedings of the Fall Symposium on Probabilistic Approaches to Natural Language, 1992.

[28] Andrew Roberts. Machine Learning in Natural Language Processing, October 16, 2003

[29] Hassan S. Statistical Part of Speech Tagger for Urdu. Thesis, National University of Computer and Emerging Science, Department of Computer Science. Lahore, Pakistan, 2007.

[30] Qing Ma, Kiyotaka Uchimoto, Masaki Murata, and Hitoshi Isahara. Elastic Neural Networks for Part of Speech Tagging, Communications Research Laboratory, MPT, Japan

[31] Diriba Merga, Automatic Sentence Parser for Oromo Language, Thesis, School of Graduate studies, Addis Ababa University, 2001.

[32] Daniel Bekele. Afaan Oromo-English Cross-Language Information Retrieval. Master's thesis Addis Ababa University, 2011.

[33] Clark, S., J. R. Curran & M. Osborne. Bootstrapping POS taggers using unlabelled data. In Proceedings of the Seventh CoNLL conference held at HLT-NAACL, Edmonton, Alberta, Canada, 27 May –1 June, 2003, pp. 49–55.

[34] Jurafsky, D and Martin H. James. Speech and Language Processing, Prentice Hall, 2000.

[35] Church, K (1988) A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the second conference on Applied Natural Language Processig, ACL.

[36] Cutting, D, Kupiec, J, Pederson, J, and Sibun, P (1992) A practical part-of-speech tagger. In: Proceedings of the third conference on Applied Natural Language Processing, ACL.

[37]http://www.scribd.com/doc/78614218/MODERN-AFAAN-OROMO-GRAMMAR. Visited on January, 2012.

[38] Bryan Jurish. A Hybrid Approach to Part-of-Speech Tagging, Berlin, German, 2003

# Appendices

## Appendix A: sample corpus

Gahee/JJ dubartooti/NN baadiyyaa/JJ wabii/JJ soorataa/NN mirkaneesuuf/VV qabataafi/JC murteessaa/JJ ta'e/AX cimsuudhaaf/VV qaamoleen/NN dhimmi/JJ ilaalatu/VV xiyeeffatanii/AD hojjechuu/VV akka/PR qaban/AX ibsame/VV./PUNC
Kunuunsi/JJ qabeenya/NN uumamaafi/JC eegumsi/JJ naannawaa/NN wabii/JJ midhaan/NN nyaataa/JJ mirkaneessuuf/VV shoora/NN olaanaa/JJ akka/PR gumaachu/VV ittigaafatamaan/JJ abbaa/NN Taayitaa/NN eegumsa/JJ Naannawaa/NN ibsame/VV ./PUNC
Kun/PP kakuu/VV Oromoon/NN qabudha/AX ./PUNC
Guyyaan/NN kun/PP sadarkaa/JJ adduyaattis/NC ta'ee/AX sadarkaa/JJ biyyaa/NN keenyaatti/JJ yeroo/AD jalqabaatiif/AD kabajameera/VV ./PUNC
Guyyichi/NP guyyaa/NN muddamsaa/NN ta'uuf/VV ./PUNC
Michuun/VV koo/PP kompyuutara/VV isaa/PP irra/PR jira/AX ./PUNC
Galmootan/NP wallitti/AD fuunaanee/VV kaa'a/VV ./PUNC
Galmeen/NP barbaachisaan/NP badanii/VV jiru/AX ./PUNC
Buna/NN kan/PR addeessutu/VV galmechaa/NN ira/PR jira/AX./PUNC
Galama/NN gara/PR biraan/JJ kiisii/NN xarapheezzaa/NN keessaa/AD arge/AX./PUNC
Waraana/NN geedaramuuf/VV guddaa/JJ kennuufi/VV fudhatu/VV keessa/PR hojjate/VV ./PUNC
Maqaan/NNP waajirichi/NN ofirraa/NN rukutu/VV hojjeessuuf/VV beeksisaniiru/VV dhaabbilee/NN 23/ON ./PUNC
Dubartii/VV gara/PR maaraguutti/VV jalqabaa/JJ taatee/AX addeessu/VV barbaachisaa/VV ./PUNC
Waanni/PP gurraachota/NN bu'awwan/JJ saayinsii/JJ mul'atu/AX jabeessi/VV ittiin/NN sanyii/JJ waggaa/NN eegi/VV ./PUNC
Qonnaa/NN keetii/PP gidduutti/PR waanni/NN ciminasaatiinJ/J waraanuun/VV beekamaadha/NN ./PUNC
Yuunvarstii/VV kanaa/PP fufe/VV ce'uu/VV safuu/NN tolee/JJ irraan/PR gahu/VV motummaa/NN hidhataa/NN bula/JJ ./PUNC
Galmootan/NP toora/NN tooraan/NN kaa'a/VV ./PUNC
Deeskiin/NN isaa/PP burjaaja'aa/VV dha/AX ./PUNC
Tuulaa/NN waraqaatu/VV deeskii/NN koo/PP irra/PR jira/AX ./PUNC
Mee/PR dubbii/NN kana/PP xiqqo/JJ qabatamaa/VV goonee/VV haa/PR ilaalluu/VV ./PUNC
Kan/NNP waan/PR ofii/PP kabaju/VV ofifille/VV kabaja/VV argata/VV ./PUNC
Fayyaan/NN waan/PR hunda/JJ caala/VV ./PUNC
Qananiisaan/NNP dorgommii/VV eegale/PR wal/PP irraa/PR hin/PR kutu/VV ./PUNC
Egaa/NN sababoota/NN kan/PR keessaafi/PR alaa/PR kanaan/PP Gadaan/NNP ammamuu/AD socho'u/VV addunyaatti/NN makamuuf/JJ danqaraa/NN isa/PP dura/PR jiran/AX kan/PR cabsee/VV darbuu/VV hin/PR dandeenye/VV ./PUNC
Bifti/JJ duulee/NN dhiibbaa/JJ jirudha/AX ./PUNC
Bofti/NN baayyee/JJ soch'u/VV Tisiisa/NN jiran/AX keessaafi/PR alaa/PR farra/JJ tisiisaa/NN ti/AX ./PUNC
Fayyaan/NN aadde/VV Faantuu/VV ammamuu/AD matatii/NN mul'ate/VV meeshaa/NN dhiigaa/NN hoteela/JJ GiyoonittiNN/NNP gorsaa/NN bal'aa/JJ kenna/VV ./PUNC
Tisiisa/NN qammooJJii/JJ ilalluu/VV dura/PR darbuu/VV kanaan/PP barbaachisaadha/VV ./PUNC

## Appendix B: Brill's Tagger Lexical Learned Rules

```
u char VV 189.333333333333
aahassuf 2 NN 43.3333333333333
ehassuf 1 VV 37
NNP e fchar NN 24.5
NN fi fhassuf 2 NC 19
leehassuf 3 NN 18
NN ifhaspref 1 PP 15.0571428571429
lahassuf 2 JJ 12
fhassuf 1 VV 11
NN niifhassuf 3 VV 11
baahassuf 3 JJ 10
VV t fhaspref 1 AX 10
NN .fgoodleft VV 9
VV qafhaspref 2 AX 8
amhaspref 2 AD 8
NN bofhaspref 2 PR 7
inahassuf 3 JJ 6
maanhassuf 4 JJ 6
PP r fchar PR 6
keeshaspref 4 PR 6
. char PN 6
NNP l fchar NN 4
tanhassuf 3 VV 4
O char NN 3
rrahassuf 3 NN 3
VV K fchar PP 3
roohassuf 3 AD 3
J char J 3
VV G fchar NN 2
u char VV 91.0666666666667
. goodleft VV 45.125
S-T-A-R-T goodright NN 42.1666666666667
ehassuf 1 VV 21.3333333333333
aahassuf 2 NN 18.5
ishaspref 2 PP 11
NN fi fhassuf 2 NC 8.5
NNP ifchar NN 7
hingoodright VV 6
dhagoodleft JJ 6
VV qafhaspref 2 AX 6
NN aniifhassuf 4 VV 5
VV n faddsuf 1 AX 4.25
NNP r fchar NN 4
leehassuf 3 NN 4
lahassuf 2 JJ 4
NN irfhaspref 2 PR 4
amhaspref 2 AD 4
```

## Appendix C: Brill's Tagger Contextual Learned Rules

```
VV JJ NEXTBIGRAM NN NN
NN JJ CURWD sadarkaa
NN JJ PREVWD kan
NN VV PREV1OR2WD wal
NN VV CURWD sochiirra
NN PR CURWD bakka
PR CC CURWD malee
NP AD PREVTAG AD
NN JJ PREVWD bara
NN JJ NEXTBIGRAM NP NN
NN JJ CURWD xiqqo
VV NN WDPREVTAG VV dubbii
VV AX CURWD ta'u
NP NN NEXTTAG JN
JJ PR SURROUNDTAG VV NN
PP NN PREV1OR2TAG PR
PR JJ PREV1OR2WD biyyaa
NNP NN NEXT1OR2OR3TAG VV
NN JJ NEXTWD oromiyaa
JJ NN NEXT1OR2OR3TAG VV
NC CC CURWD fi
NN PN CURWD ?
NN NNP NEXT1OR2WD jila
NN PP CURWD ati
NN NNP LBIGRAM STAART Itti
NN AD CURWD bira
PP NN NEXT2TAG NN
VV NN NEXT1OR2TAG PS
VV NN NEXTWD akka
NN JJ WDNEXTTAG nagaya AX
NN NNP NEXTWD oogganame
NN PP NEXT1OR2WD ,
VV JJ CURWD dhugaadha
NN AD CURWD sirritti
NN PR PREVWD keessaafi
VV PR WDNEXTTAG dursa VV
NN PR WDNEXTTAG jalaa VV
VV JN WDPREVTAG NN afur
PS PP PREVTAG JJ
NP NN PREV1OR2OR3TAG NN
II VV NEXT1OR2TAG STAART
VV NN PREVWD irraas
JJ NN NEXT1OR2OR3TAG VV
PP NN NEXT1OR2OR3TAG NN
NN VV NEXTTAG PN
NP NN PREV1OR2OR3TAG NN
NC CC CURWD fi
NN PN CURWD ?
NN NNP NEXT1OR2WD jila
```

# Declaration

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

**Declared by:**

Name: Abraham Gizaw Ayana

Signature: _____

Date: _____

**Confirmed by Advisor:**

Name: Sebsibe Hailemariam (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa, June, 2012