ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

DEVELOPMENT OF PART OF SPEECH TAGGER USING HYBRID APPROUCH

BY:

GETACHEW EMIRU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

OCTOBER, 2016
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

DEVELOPMENT OF PART OF SPEECH TAGGER USING HYBRID APPROUCH

BY:

GETACHEW EMIRU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

ADVISOR: SOLOMON TEFERRA (PhD)

Signature of the Board of Examiners for Approval

Name	**signature**
1. _____	_____
2. _____	_____
3. _____	_____

Declaration:

This thesis is my original work which hasn't been submitted as a partial requirement for a degree in any university.

_____

Getachew Emiru

October, 2016

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

_____

Solomon Teferra (PhD)

Acknowledgments

First and foremost I thank my almighty God, for his help, mercy and who realized this work and added a new status on my life.

Second, I am grateful to my advisor Dr. Solomon Teferra for his constant encouragement, guidance and necessary information which was contributing for the completion of this thesis. Without his insightful and constructive comments, the completion of this thesis would have not been possible. Also, I have special thanks for Mr. Birhanu Takele and Bali Girma for their valuable guidance especially on word class, sentence structures of Afaan Oromoo which was very important for the success of this work.

In addition, I want to thank Mr. Abraham Gizaw who provided me Afaan Oromoo tagged data which helped to investigate this study.

I am also greatly indebted to my friends who directly or indirectly involved in this research; particularly my classmates.

It is my pleasure to express my gratitude wholeheartedly to my family; specially my father, Emiru Amenu and my mother, Giditu Wakweya.

At Last but not least, I want to thank administrative and academic staffs of the school of Information science, Addis Ababa University.

Table of Contents

Content **Pages**

List of Tables

List of Figures

Acronyms and Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AO	Afaan Oromoo
HMM	Hidden Markov Model
MLE	Maximum-Likelihood Estimation
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part of Speech
TEL	Transformational Error Driven Learning

Abstract

Part of speech tagger is one of the subtasks in NLP application which is essential for other Natural Language Processing (NLP) applications. It is a process of assigning a corresponding POS tag for a word that describes how the word is used in a sentence. Even though the accuracy is less, different researchers developed part of speech taggers for Afaan Oromoo using approaches like rule based and HMM separately. In this thesis, the development of part of speech tagger using hybrid approach that combines rule based and HMM approaches was conducted for Afaan Oromoo. The transformation based learner, which is a rule based tagger, tag the words based on rules, or transformations induced directly from the training corpus without human intervention or expert knowledge. The HMM tagger, tags the words based on the most probable path for a given sequence of words. The hybrid approach of Afaan Oromoo part of speech taggers developed in this thesis uses HMM tagger as initial annotators and Brill's' tagger as a corrector based on fixed threshold value. NLTK 3.0.2 and python 3.4.3 were used for the implementation and experiment. To minimize data requirement and the cost of data preparation we used bootstrapping method. To train and test the model 1517 sentences were used, that is collected from Afaan Oromoo news agencies and Medias. For experimental analysis we used 85% for training and the remaining 15% was used for testing. The performance analysis of the three taggers, namely: HMM, rule based and hybrid tagger were tested with the same training and testing set they achieved accuracy of 91.9%, 96.4% and 98.3%, respectively. In conclusion, the accuracy of the hybrid tagger clearly shows that a clear improvement performance rather than separated taggers. To increase the performance of the tagger wide coverage/domain area of training data and morphologically segmented words were recommended for future works.

Keywords: *Hybrid Tagger, Afaan Oromoo, Artificial Intelligence (AI), part of speech tagging*

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Natural language processing (NLP) is a multidisciplinary field that composes of computer science, artificial intelligence, and computational linguistics. It is concerned with the natural interactions between computers and human beings in human languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation. Computational linguistics is the study of Natural Languages by means of computational perspectives. It is meant to be able to identify 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) (Manning & Schutze, 2000).

The study and application of general Machine Learning (ML) algorithms to the area of Natural Language Processing (NLP) is a currently very active area of research. Some of this most research tasks are: Automatic summarization, Co-reference resolution, Discourse analysis, Machine translation, Morphological segmentation, Named entity recognition (NER), Natural language generation, Natural language understanding , Optical character recognition (OCR), Part-of-speech tagging, Parsing, Question answering, Sentiment analysis, Speech recognition, Word sense disambiguation and Information retrieval (IR)(Jurafisky & Martin, 2006).

Some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aids in solving larger tasks. What distinguishes these tasks from other potential and actual NLP tasks is not only the volume of research devoted to them but the fact that for each one there is typically a well-defined problem setting, a standard metric for evaluating the task, standard corpora on which the task can be evaluated, and competitions devoted to the specific task.

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.

Part of Speech (POS) Tagging is the essential basis of Natural Language Processing (NLP). It is the process in which each word is assigned to a corresponding POS tag that describes how this word is used in a sentence. The development of this an automatic POS tagger requires either a comprehensive set of linguistically motivated rules or a large annotated corpus (Sandipan, 2009). But such rules and corpora have been developed for a few languages like English and some other languages. POS taggers for Afaan Oromo languages are not readily available due to lack of such rules and large annotated corpora and moderate accuracy can only be achieved in rule based techniques.

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 (Tarveer, 2008). 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 and hybrid approaches. But in this study the hybrid model (rule based and statistical hidden Markov model) approaches was used in order to develop Afaan Oromo part of speech tagger.

In 1992, Eric Brill introduced a POS tagger which is called Transformational based learning tagger that was based on rules or transformations, where the grammar is induced directly from the training corpus without human intervention or expert knowledge. According to Brill (Brill, 1995), there is a very small amount of general linguistic knowledge built into the system with no language-specific knowledge. The only additional component necessary is a sufficiently large and manually annotated training corpus which serves as input to the tagger. The system is then able to derive lexical/morphological and contextual information from the training corpus and 'learns' how to deduce the most likely part-of-speech tag for a word. Once the training is completed, the tagger can be used to annotate new unannotated corpus (Megyesi, 1998).

In this study, Afaan Oromoo part of speech tagger using hybrid approaches was conducted and the model was tested using separated training and testing set. The Brill's and HMM used in this thesis was which is built in nltk and trained based on the suffix of the language. The developed hybrid model is used HMM tagger as initial state annotators and Brill's tagger as a correctors.

1.2 Statement of the Problem

Lexical attributes, like syntactic and semantic attributes are in most cases ambiguous in every language, particularly in Afaan Oromoo. This ambiguity of the languages can be solved using automated NLP applications like POS tagger and word sense disambiguation. Parts-of-speech (also known as POS, word classes, morphological classes, or lexical tags) for language processing it give the large amount of information about a word and its neighbors in a given sentence(Jurafsky & Martin, 2006). 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 (Manning & Schutze, 2000).

The rule-based approaches use contextual information to constrain the possible part of speech tags or to assign a part of speech tag to a word (Brill, 1992) (Mikheev, 1996). These rules are often known as context frame rules. For instance, a context frame rule for English might be as follows: "if the word is preceded by a determiner and followed by a noun, tag it as an adjective". In addition to context information, many taggers use morphological information to aid in the disambiguation process. Some systems go beyond using contextual and morphological information by including rules pertaining to such factors as capitalization and punctuation. The usefulness of this type of information highly depends on the language being tagged.

Brill's or transformation-based tagging is one of rule based algorithm which extracts the rules from manually annotated text. 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 words most frequent tag, and then applying the transformations.

The statistical (stochastic) approaches select the most likely interpretation based on the estimation of statistics from unambiguously tagged text. Either word frequencies or probabilities can be used as the criterion to be maximized. Hidden Markov-model based taggers is one of statistical/stochastic approaches that assign to a sentence the tag sequence that maximizes *Prob(word* | tag),Prob(tag | previous n tags). These probabilities can be estimated directly from a manually tagged corpus. These stochastic taggers have a number of advantages over the manually built taggers, including obviating the need for laborious manual rule construction, and possibly capturing useful information that may not have been noticed by the human engineer. Almost all recent work in developing automatically trained part-of-speech taggers has been on further exploring Markov model based tagging(Cutting et al. 1992)(Schutze & Singer, 1994). The drawback of this approach is we could condition tags on preceding tags not on preceding words or it uses trigram taggers by going to fourgram or even higher order taggers than context. The approach is also not feasible because of the large number of parameters we would need. Even with trigram taggers, we had to smooth and interpolate because maximum likelihood estimates were not robust enough.

One of the strengths transformation-based tagging methods 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. 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 (Manning & Schutze, 2000).

Brill's or Transformation based tagging does not make the battery of standard methods available that probability theory provides. For example, no extra work is necessary in a probabilistic model for a .k-best tagging - a tagging module that passes a number of tagging hypotheses with probabilities on to the next module downstream (such as the parser). It is possible to extend transformation-based tagging to .k-best tagging by allowing rules of the form add tag A to B if . . ." so that some words will be tagged with multiple tags. However, the problem remains that we don't have an assessment of how likely each of the tags is. The first tag could be 100 times more likely

than the next best one in one situation and all tags could be equally likely in another situation. This type of knowledge could be critical for constructing a parse.

An important characteristic of learning methods is the way prior knowledge can be encoded. Transformation-based tagging and probabilistic approaches have different strengths here. The specification of templates for the most appropriate triggering environments offers a powerful way of biasing the learner towards good generalizations in transformation based learning. The templates seem obvious, because of what we know about syntactic regularities. A large number of other templates that are obviously inappropriate are conceivable (e.g., .the previous even position in the sentence is a noun.).

In contrast, the probabilistic Markov models make it easier to encode precisely what the prior likelihood for the different tags of a word are (for example, the most likely tag is ten times as likely or just one and a half times more likely). The only piece of knowledge we can give the learner in transformation-based tagging is which tag is most likely.

Since the transformation-based tagging tags the words based on the rules extracted from training corpus, Brill's tagger have not addressed the problem of unknown words. In the transformation-based unknown-word tagger, the initial-state annotator naively assumes the most likely tag for an unknown word is "proper noun" if the word is capitalized and "common noun" otherwise (Brill, 1995). The hmm give tags for unknown words using the Maximum Likelihood Estimation (MLE) of parameters. Maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations, by finding the parameter values that maximize the likelihood of making the observations given the parameters(Christophe, 2013).

In this paper, to gain the strength of these separated approaches the researcher has proposed the hybrid/combination of rule-based with a statistical hidden Markov model approaches.

Not only has the choice of the methods/approaches, the coverage of domain of training data had a great impact on performance of the taggers. This specific domain coverage caused of occurrence of many unknown words which greatly degraded the performance of the taggers. The percentage of words not in the dictionary can be very high when trying to tag material from some specific domain. If the training set is small, the tag set large, the test corpus significantly different from the training corpus, or we are confronted with a larger than expected number of unknown words, then performance can be decreased. It is important to stress that these types of external conditions often have a stronger influence on performance than the choice of tagging

method - especially when differences between methods reported are on the order of half a percent (Manning & Schutze, 2000). By consideration of Manning and Schutze idea, beside of being combining the approaches the training corpus used in this thesis are wide coverage/balanced corpus which are collected from different domain areas: Afaan Oromoo news, holy bible and dram scripting.

1.3 Research Questions

The following research questions will be answered in the research:

- ✓ How to improve the performance of part of speech tagger using combination approaches, particularly for Afaan Oromoo?
- ✓ How to improve the performance of a rule based tagger by the probabilistic power of an HMM model?
- ✓ How to develop a hybrid POS tagger that performs better than both rules based and machine learning individual tagging?

1.4 Objectives of the Study

1.4.1 General Objective

The general objective of the research is to investigate the use of hybrid (rule based and statistical hidden Markov models) approaches to the development of POS tagging for Afaan Oromoo.

1.4.2 Specific Objectives

So as to achieve the above general objective, the research accomplished the following specific objectives:

- ✓ To review, analyze and understand the theoretical basis of POS tagging and identify the relations that exist between the previous research attempts in the area.
- ✓ To study the basic word category and morphological property of Afaan Oromoo language.
- ✓ To collect and design corpus for training and testing of the model

- ✓ To develop HMM POS tagger
- ✓ To develop rule based POS tagger
- ✓ To see to what extent of POS tags are misplaced in HMM and rule based taggers
- ✓ To design and model hybrid POS tagger for Afaan Oromoo language.
- ✓ To see the performance of the hybrid tagger
- ✓ To forward conclusion and recommendation based on the findings

1.5 Scope of the Study

This paper is prepared on the development of Afaan Oromoo part of speech tagger using hybrid model (rule based and HMM approaches). It focused on labeling of words with their correct tags of language particularly for Afaan Oromoo. Because of the python didn't recognized the symbol /'/ and /-/, the study didn't included any glottal words /hudhaa/ or /'/, compound words and tagset that can identify gender, tenses with different feature set. The corpus used in this thesis work is domain specific corpus, a text corpus that is collected only from Afaan Oromoo news, drama scripting and holy bible.

1.6 Challenges and Limitations of the Study

Afaan Oromoo has a very rich morphology as other African languages. In agglutinative languages most of the grammatical information is conveyed through affixes and other structures. Therefore, the grammatical information of the language is described in relation to its morphology. As Afaan Oromoo is an agglutinative and morphologically rich, each root word can be combined with multiple morphemes to generate huge number of word formation. For the purpose of supporting such inflectionally rich languages, the structure of each word has to be identified. Afaan Oromoo has compound, derived and simple nouns, verbs, and adjectives. It also has first person, second person, and derived pronouns. Nouns get inflected for number. Gender, number, tense, voice, aspect and mood cause inflections to verbs. Many times it is context which decides whether a word is a noun or adjective or adverb or post position.

Afaan Oromoo glottal words /hudha/ or /'/ and compound words are also the other challenges for accomplishment of this work. Hudhaa or /'/ in Afaan Oromoo represent one consonant letter but during the implementation the symbol is represented by other characters which is unknown in

Afaan Oromoo letters. The compound words are words which is composed from two separated words and used as one word by separating the words using hyphen /-/ symbols. The tagger tags the compound words as two words. Collecting of the data from different domain categories and preparing/tagging the collected data which is needed for building a model and testing was also another challenge's this work.

1.7 Methodology

To achieve the objective of this research experimental quantitative research methodology was applied. This experimental quantitative research is preferred due to it could be used to determine the relationship between quantitative variables or compare groups (Kothari, 2004). In case of this study the percentage of tags exits separately in the corpus was analyzed and the percentage of misplaced tags tagged by each tagger and the accuracy of the three taggers measured in figure independently.

1.7.1 Modeling Techniques

In this study the development of part of speech tagger for Afaan Oromoo using hybrid approach was conducted. This hybrid approaches is a combination of rule based and statistical hidden Markov model. In this hybrid approach adapted for Afaan Oromo POS tagger in which HMM used as initial state annotator and Brill's tagger used as a corrector. HMM tagger tags the words by finding the most probable path of a word in a given sequence using emission and transition probability. The HMM based tagger relies on the statistical property of words along with their part of speech categories. Such statistical property can be distributional probability of words with tags which can be obtained during the training phase of the system.

The corrector Brill's (transformation error driven learning) rules are automatically learned from a manually annotated corpus. The rules are the important elements for annotating words in the TEL approach. Both models, the rule based and HMM based taggers, have their own pros and cons. To gain the advantage of both approaches the hybrid model that uses HMM tagger as initial state tagger and Brill's as a corrector was used.

1.7.2 Data/Corpus Preparation

Corpus, plural corpora, is a collection of text, documents, speech, etc. It can be a text, speech or documents whereby each word in the text is attached with linguistic information (Jurafky & Martin, 2006)(Teubert, 2001)(Thomas, 1996).

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 that specifies the word class category. The annotated corpus can be used in many NLP applications like part of speech tagger training and testing, parsing and etc. In this thesis work, the annotated corpus used is considered to be a text tagged with the corresponding part of speech tags.

A balanced corpus is the tagged text i.e. that is collected from different domain areas of the language. The domains can be text of news category, fiction category, editorial category, scientific category, etc. but, getting this all different domains areas needs time, effort/skills of language experts and money. By consideration of these limitations, for this study we used a corpus collected from domains of news category 247 sentences, from drama scripting 238 sentences and from holy bible 215 sentences. In this thesis 700 sentences were collected from different domains and added to 817 sentences prepared on previous work(Abraham, 2013). Totally 1517 sentences were used for training and testing purpose.

To tag those a raw corpus collected from specified domain area the bootstrapping methods was used and the correction was made by language experts. Bootstrapping is preparing text from already tagged text: this technique generally consists of using a small tagged corpus to train a system and having the system tag another subset of the corpus that gets disambiguated later. We have used these techniques but the necessary human effort is still considerable.

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.

1.7.3 Tools

To conduct this research on POS tagger for Afaan Oromo text, an open source Natural Language ToolKit (NLTK) and Python programming language were used. The rationale behind the choice

of these two tools is that they are suitable for processing different NLP tasks. NLTK is an open source tool that contains open source python modules, linguistic data and documentation for research and development in natural language processing (Steven and et al, 2009). Python is an easy to learn but powerful programming language especially for text processing in NLP applications. Using this python the module was created and import to NLTK. Python has efficient high level data structures and a simple but effective approach to object-oriented programming (Bird, 2006). Chapter 5 contains the detail tools and techniques and how the tools are implemented.

1.7.4 Testing Procedure

To test the model incremental and accuracy measurement approaches was used. In Incremental approaches the model evaluated using 10% of training set and increased by 10% of the remaining training set. These different portions of training set was evaluated using the same test set and the over whole performance of the taggers were evaluated using confusion matrix evaluation.

1.8 Significance of the study

The main goal of the study is to develop hybrid model of part of speech tagger for Afaan Oromoo using hybrid (combines of rule based and HMM) approaches for Afaan Oromoo Language. Previously, Afaan Oromoo part of speech tagger developed using single approaches such as Rule based and HMM, but the satisfactory accuracy didn't achieved. The main reason of scoring less accuracy is why both approaches has own pros and cons. The hybrid model developed contains advantage of both approaches. It contains the rules and contextual probability of words of training data that is used to tag the raw text. The result of this study will be useful for Afaan Oromoo Information retrieval (IR) in stemming, since knowing a word's part-of-speech can help tell us which morphological affixes it can take. It can also enhance an IR application by selecting out nouns or other important words from a document. Automatic assignment of part-of- speech plays a role in parsing, in word-sense disambiguation algorithms, and in shall parsing of texts to quickly find names, times, dates, or other named entities for the information extraction applications. The corpora that have been marked parts-of-speech are very useful for linguistic research. For instance, they can be used to help find instances or frequencies of particular constructions.

1.9 Organization of the Paper

This thesis is organized into six chapters. The first chapter describes the introductory part of the thesis. The second chapter is all about literature review and related works. 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 tagset and corpus preparation of Afaan Oromoo language. Afterwards, the fourth chapter deals with design and implementations of Afaan Oromo HMM, Brill's and hybrid tagger and chapter five mainly focuses on the experimental analysis of the three part of speech taggers namely HMM, Brill's and hybrid tagger. Finally, the last chapter presents conclusion and recommendation of future work.

CHAPTER TWO

LITERATURE REVIEW

2 Literature Review

This chapter tried to discuss the literature review and related works done on part of speech tagging specifically on Afaan Oromo and some other local Ethiopian language.

Natural Language processing is one of the current hot research areas for scientists and academic researchers. The ultimate goal of research on Natural Language Processing is to parse and understand language. Some of the most commonly researched tasks in NLP are:- Automatic summarization, Coreference resolution, Discourse analysis, Machine translation Morphological segmentation, Named entity recognition (NER), Natural language generation, Natural language understanding , Optical character recognition (OCR), Part-of-speech tagging, Parsing, Question answering, Sentiment analysis, Speech recognition, Word sense disambiguation and Information retrieval (IR). Note that some of the natural language processing tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks. What distinguishes these tasks from other potential and actual NLP tasks is not only the volume of research devoted to them but the fact that for each one there is typically a well-defined problem setting, a standard metric for evaluating the task, standard corpora on which the task can be evaluated, and competitions devoted to the specific task((Jurafisky & Martin, 2006)(Abraham ,2013).

As we have seen in the preceding works, we are still far from achieving this goal. 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. Tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part of speech (Manning & Schutze, 2000).

Automatic assignment of descriptors to the given tokens is called Tagging. The descriptor is called tag. The tag may indicate one of the parts-of-speech, semantic information, and so on. So tagging a kind of classification (Robin, 2009).

Part-of-speech tagging is the process of assigning a part-of-speech or other syntactic class marker to each word in a corpus. It means assigning appropriate grammatical classes (i.e. appropriate Part of-Speech tags) to each word in a natural language sentence. Because tags are generally also applied to punctuation, tagging requires that the punctuation marks (period, comma, etc) be separated off of the words. Thus tokenization of the sort is usually performed before, or as part of, the tagging process, separating commas, quotation marks, etc., from words, and disambiguating end-of-sentence(Robin, 2009(Jurafisky & Martin, 2006).

A word's part-of-speech can tell us something about how the word is pronounced. The word content, for example, can be a noun or an adjective. They are pronounced differently (the noun is pronounced CONtent and the adjective conTENT). Thus knowing the part-of-speech can produce more natural pronunciations in a speech synthesis system and more accuracy in a speech recognition system. (Other pairs like this include OBject (noun) and obJECT (verb), DIScount (noun) and disCOUNT (verb)(Cutler ,1986).

Parts-of-speech can also be used in stemming for information retrieval (IR), since knowing a word's part-of-speech can help tell us which morphological affixes it can take. They can also enhance an IR application by selecting out nouns or other important words from a document. Automatic assignment of part-of- speech plays a role in parsing, in word-sense disambiguation algorithms, and in shall parsing of texts to quickly find names, times, dates, or other named entities for the information extraction applications.. Finally, corpora that have been marked for parts-of-speech are very useful for linguistic research. For example, they can be used to help find instances or frequencies of particular constructions.

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 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 (Fahim et.al, 2006) (Jurafisky and Martin, 2006).

2.1 Tagset

Tagset is the set of tags from which the tagger is supposed to choose to attach to the relevant word. The collection of tags used for a particular task is known as a tagset.

Every tagger will be given a standard tagset based the nature of the language. The tagset may be coarse such as N (Noun), V(Verb), ADJ(Adjective), ADV(Adverb), PREP(Preposition), CONJ(Conjunction) or fine-grained such as NNOM(Noun-Nominative), NSOC(Noun-Sociative), VFIN(Verb Finite),VNFIN(Verb Nonfinite) and so on. Most of the taggers use only fine grained tagset (Robin, 2009).

2.2 Part of Speech Tagging Approaches

There are different approaches for POS Tagging. The following figure classifies different Part of speech tagging models.

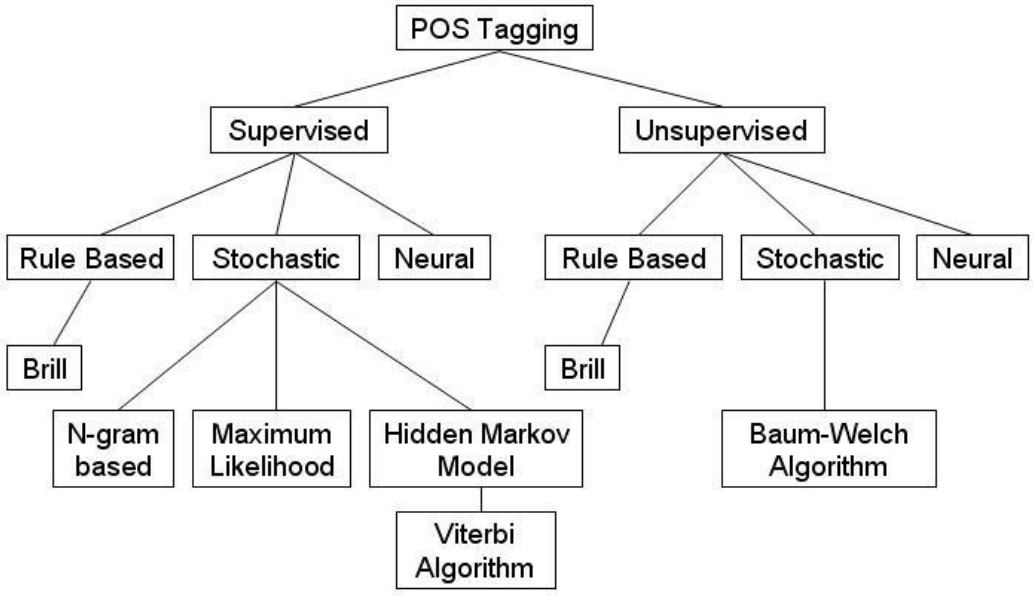

Figure 2.1 Classification of POS tagging approaches

The supervised POS tagging models require a pre annotated corpus which is used for training to learn information about the tagset, word-tag frequencies, rule sets, etc (Karthik et.al, 2006). The performance of the models generally increases with increase in the size of the corpus.

The unsupervised POS tagging models do not require a pre-annotated corpus. Instead, they use advanced computational techniques like the Baum Welch algorithm to automatically induce tagsets, transformation rules, etc. Based on this information, they either calculate the probabilistic information needed by the stochastic taggers or induce the contextual rules needed by rule based systems or transformation based systems (Linda, 1995).

Supervised taggers typically rely on pre-tagged corpora to serve as the basis for creating any tools to be used throughout the tagging process, for example: the tagger dictionary, the word/tag frequencies, the tag sequence probabilities and/or the rule set. Unsupervised models, on the other hand, are those which do not require a pre tagged corpus but instead use sophisticated computational methods to automatically induce word groupings (i.e. tag sets) and based on those automatic groupings, to either calculate the probabilistic information needed by stochastic taggers or to induce the context rules needed by rule-based systems. Each of these approaches has pros and cons.

The primary argument for using a fully automated approach to POS tagging is that it is extremely portable. It is known that automatic POS taggers tend to perform best when both trained and tested on the same genre of text. 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 hand tagging of training data is a costly and time-consuming process. There are, however, drawbacks to fully automating the POS tagging process. The word clustering which tend to result from these methods is very coarse, i.e., one loses the fine distinctions found in the carefully designed tag sets used in the supervised methods.

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
Improving Brill's Tagger Lexical and Transformational Rule for Afaan Oromoo Language and open classes is relevant when handling unknown words, since these are more likely to belong to open classes.

Throughout years a lot of different methods or approaches have been used to solve the lexical ambiguity of a word in a text. Most tagging algorithms fall into one of two classes: rule-based taggers and stochastic taggers. There are also other approaches such as Artificial Neural Networks and memory based approaches.

2.2.1 Statistical Approach

The stochastic models include frequency, probability or statistics. The term 'stochastic tagger' can refer to any number of different approaches to the problem of POS tagging. Any model which somehow incorporates frequency or probability, i.e. statistics, may be properly labelled stochastic. The simplest stochastic taggers disambiguate words based solely on the probability that a word occurs with a particular tag. In other words, the tag encountered most frequently in the training set is the one assigned to an ambiguous instance of that word. The problem with this approach is that while it may yield a valid tag for a given word, at can also yield inadmissible sequences of tags.

They can be based on different methods such as n-grams, maximum-likelihood estimation (MLE) or Hidden Markov Models (HMM). HMM-based methods require evaluation of the argmax formula, which is very expensive as all possible tag sequences must be checked, in order to find the sequence that maximizes the probability. So a dynamic programming approach known as the Viterbi Algorithm is used to find the optimal tag sequence (Manoj, 2005).

There have also been several studies utilizing unsupervised learning for training a HMM for POS Tagging. The most widely known is the Baum-Welch algorithm (Baum, 2000), which can be used to train a HMM from un-annotated data. And lastly, both supervised and unsupervised POS tagging models can be based on neural networks.

An alternative to the word frequency approach is to calculate the probability of a given sequence of tags occurring. This is sometimes referred to as the *n-gram* approach, referring to the fact that

the best tag for a given word is determined by the probability that it occurs with the n previous tags. The most common algorithm for implementing an n-gram approach is known as the *Viterbi Algorithm*, a search algorithm which avoids the polynomial expansion of a breadth first search by "trimming" the search tree at each level using the best N *Maximum Likelihood Estimates* (where n represents the number of tags of the following word).

2.2.1.1 Hidden Markov Models (HMM)

Markov models extract linguistic knowledge automatically from the large corpora and do POS tagging. Markov models are alternatives for laborious and time-consuming manual tagging(Robin, 2009).

The name Markov model is derived from the term Markov property. Markov property is an assumption that allows the system to be analyzed. According to Markov property, given the current state of the system, the future evolution of the system is independent of its past. The Markov property is assured if the transition probabilities are given by exponential distributions with constant failure or repair rates (Robin, 2009).

Hidden Markov Model is a standout amongst the effectively utilized language model (1-gra, .n-gram) for deriving labels which utilizes rare measure of data about the language, separated from simple context related data. It is a stochastic based construct which could be utilized to tackle the classification issues that have a state sequence form. The model has a number of interconnected states connected by their transition probability. A transition probability is the probability that a system moves from one state to another. A process begins in one of the states, and moves to another state, which is governed by the transition probability. An output symbol is emitted as the process moves from one state to the next. These are also known as the Observations. HMM basically outputs a sequence of symbols. The emitted symbol depends on the probability distribution of the particular state. But the exact sequence of states with respect to a typical observation sequence is not known (hidden).

The model described here follows the concepts given as Bayes (1763) cited in (Jurafisky & Martin, 2006), 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 BAYESIAN known since the work of INFERENCE.

Bayesian inference or Bayesian classification was applied successfully to language problems as early as the late 1950s, including the OCR work of Bledsoe in 1959, and the seminal work of Mosteller and Wallace (1964) on applying Bayesian inference to determine the authorship of the Federalist papers.

In a classification task, we are given some observation(s) and our job is to determine which of a set of classes it belongs to. Part-of-speech tagging is generally treated as a sequence classification task. So here the observation is a sequence of words (let's say a sentence), and it is our job to assign them a sequence of part-of-speech tags.

In order to calculate best sequence of tags, the Bayesian interpretation starts by considering all possible sequences of classes of a given sentence, all possible sequences of tags. Out of this universe of tag sequences, we want to choose the tag sequence which is most probable given the observation sequence of n words w^n_1. In other words, we want, out of all sequences of n tags t^n_1 the single tag sequence such that $P(t^n_1 | w^n_1)$ is highest. We use the hat notation ˆ to mean "our estimate of the correct tag sequence".

(2.1) $$\hat{t}^n_1 = \underset{t^n_1}{\operatorname{argmax}} \frac{P(t^n_1 | w^n_1)}{t^n_1}$$

The function $\operatorname{argmax}_x f(x)$ means "the x such that f (x) is maximized". Equation (2.1) thus means, out of all tag sequences of length n, we want the particular tag sequence t^n_1 which maximizes the right-hand side. While (2.1) is guaranteed to give us the optimal tag sequence, it is not clear how to make the equation operational; that is, for a given tag sequence t^n_1 and word sequence w^n_1, we don't know how to directly compute $P(t^n_1 | w^n_1)$.

The intuition of Bayesian classification is to use Bayes' rule to transform (2.1) into a set of other probabilities which turn out to be easier to compute. Bayes' rule is presented in (2.2); it gives us a way to break down any conditional probability P(x|y) into three other probabilities:

(2.2) $$P(x|y) = P(y|x)P(x)P(y)$$

We can then substitute (2.2) into (2.1) to get (2.3):

(2.3) $$\hat{t}^n_1 = \underset{t^n_1}{\operatorname{argmax}} \frac{P(w^n_1|t^n_1)P(t^n_1)}{P(w^n_1)}$$

We can conveniently simplify equation (2.3) by dropping the denominator P (w^n_1). Why is that? Since we are choosing a tag sequence out of all tag sequences, we will be computing $P(w^n_1|t^n_1)P(t^n_1)P(w^n_1)$ for each tag sequence.

But P (w^n_1) doesn't change for each tag sequence; we are always asking about the most likely tag sequence for the same observation w^n_1, which must have the same probability $P(w^n_1)$. Thus we can choose the tag sequence which maximizes this simpler formula:

(2.4) $$\hat{t}^n_1 = \underset{t^n_1}{\operatorname{argmax}} \; P(w^n_1|t^n_1)P(t^n_1)$$

To summarize, the most probable tag sequence $\hat{t}^n_1$ given some word string w^n_1 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 $P(t^n_1)$), and the likelihood of the word string$P(w^n_1|t^n_1)$:

(2.5) $$\hat{t}^n_1 = \underset{t^n_1}{\operatorname{argmax}} \; | P(w^n_1|t^n_1) \; | P(t^n_1)$$

Unfortunately, (2.5) 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, and of the other tags around it:

(2.6) $$P(w^n_1|t^n_1) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

The second assumption is that the probability of a tag appearing is dependent only on the previous tag, the bigram assumption.

(2.7) $$P(tn1) \approx \prod_{i=1}^{n} P(ti|ti-1)$$

Plugging the simplifying assumptions (2.6) and (2.7) into (2.5) results in the following equation by which a bigram tagger estimates the most probable tag sequence: The HMM does this with the Viterbi algorithm, which efficiently computes the optimal path through the graph given the sequence of words forms.

(2.8) $$\hat{t}^n_1 = \underset{t^n_1}{\operatorname{argmax}} P(t^n_1|w^n_1) \approx \underset{t^n_1}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})$$

Equation (2.8) contains two kinds of probabilities, tag transition probabilities and word likelihoods. Let's take a moment to see what these probabilities represent. The tag transition probabilities, P(ti|ti−1), represent the probability of a tag given the previous tag.

Simply we can calculate tag transition probabilities as follows:

(2.9) $$P(t_i|t_i-1) = \frac{C(t_i-1, t_j)}{C(t_i-1)}$$

The word likelihood probabilities, P(wi|ti), represent the probability, given that we see a given tag, that it will be associated with a given word. During likelihood probability we can compute the MLE estimate of a word. That is as follows:

(2.10) $$P(w_i|t_i) = \frac{C(t_j,w_i)}{C(t_i)}$$

If we assume the probability of a tag depends only on one previous tag then the model developed is called bigram model. Each state in the bigram model corresponds to a POS tag. The probability of moving from one POS state to another can be represented as $P(t_i|t_j)$. The probability of word being emitted from a particular tag state can be represented as $P(w_i|t_j)$. Assume that the sentence,

"Caalaan kaleessa deeme " / Chala went yesterday / is to be tagged. Obviously, the word, ***Caalaa*** is noun so can be annotated with tag, say NN, ***Kaleessa*** is Adverb so the tag can be AD, and ***deeme*** is a verb so the tag can be VV. So, the sentence tagged as follows:

Caalaan|NN kaleessa|AD deeme|VV

Given this model, P(NN AD VV | **Caalaan kaleessa deeme**) is estimated as

P(NN | START) * P(AD | NN) * P(VV | AD) * P(**Caalaan** | NN) * P(**Kaleessa** | AD) * P(**deeme** | VV)

This is how to derive probabilities required for the Morkov model.

2.2.1.1.1 Unknown words in HMM

Supervised training is the process of maximizing the joint probability of the symbol and state sequences. This is done via collecting frequencies of transitions between states, symbol observa-

tions while within each state and which states start a sentence. These frequency distributions are then normalised into probability estimates, which can be smoothed if desired.

They are around 4 types of smoothing techniques namely lambda fd, bins: nltk.LaplaceProbDist(fd, bins), lambda fd, bins: nltk.LidstoneProbDist(fd, 0.1, bins), lambda fd, bins: nltk.WittenBellProbDist(fd, bins), lambda fd, bins: nltk.SimpleGoodTuringProbDist(fd, bins. In this paper to assign probability for unknown words among the four smoothing techniques we used lambda fd, bins nltk.LidstoneProbDist(fd, 0.1, bins) which is built in nltk hmm tagger.

Lidstone's law is which generalises Laplace's and allows to add an arbitrary value to unseen events(Manning & Schiutze, 2000). So, for a relatively large number of unseen events, we can choose to add values lower than 1. For a relatively small number of unseen events, we may choose 1, or even larger values, if we have a large number of observations (N).

$$P_{Ld}(x) = \frac{count(x) + \lambda}{N + B\lambda} \qquad \lambda > 0$$

Where B is the number of bins training instances are divided into.

To use Laplace's or Lidstone's laws in a HMM-based tagger we have to smooth all probabilities involved in the model:

$$\pi_i = \frac{count(s_i(t=0)) + \lambda_\pi}{count(sentences) + B_{tag}\lambda_\pi}$$

$$a_{ij} = \frac{count(s_i \rightarrow s_j) + \lambda_A}{count(s_i) + B_{tag}\lambda_A}$$

$$P(s_i) = \frac{count(s_i) + \lambda_s}{count(words) + B_{tag}\lambda_s}$$

$$P(w_k) = \frac{count(w_k) + \lambda_w}{count(words) + B_w\lambda_w}$$

Where $\mathbf{B_{tag}}$ is the number of possible tags, $\boldsymbol{\pi_i}$ is the initial state probabilities, $\mathbf{a_{ij}}$ is the state transition probabilities and $\mathbf{b_{ijk}}$ is the symbol emission probabilities and $\mathbf{B_w}$ is the number of words in the vocabulary.

Since there are different counts involved in each probability, we have to consider different $\boldsymbol{\lambda}$ values for each formula. In the case of Laplace's law, all $\boldsymbol{\lambda}$ are set to 1, but when using Lidstone's,

we want to determine which the best set of λ is. This advantage makes it is more preferable than Laplace's and others to assign best accuracy for unseen words.

2.2.2 Transformation-Based Approach

Transformation-based learning (TBL) is a rule-based algorithm for automatic tagging of parts-of-speech to the given text. TBL transforms one state to another using transformation rules in order to find the suitable tag for each word. TBL allows us to have linguistic knowledge in a readable form. It extracts linguistic information automatically from corpora. The outcome of TBL is an ordered sequence of transformations of the form as shown below.

Tag_i -> Tag_j in context C

A typical transformation-based learner has an initial state annotator, a set of transformations and an objective function.

Initial Annotator

It is a program to assign tags to each and every word in the given text. It may be one that assigns tags randomly or a Markov model tagger. Usually it assigns every word its most likely tag as indicated in the training corpus. For example, walk would be initially labelled as a verb.

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 tag two before is ti and the following word is W. Consider the following sentence,

The rabbit runs.

A typic TBL tagger (or Brill Tagger) can easily identify that rabbit is noun if it is given the rule,

If the previous tag is an article and the following tag is a verb.

How Transformation Based Learning Works.

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 transforamtions 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.

Advantages of Transformation Based Learning

- Small set of simple rules that are sufficient for tagging is learned.
- As the learned rules are easy to understand development and debugging are made easier.
- Interlacing of machine-learned and human-generated rules reduce the complexity in tagging.
- 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.
- 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

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

2.3 Related Works

This section describes the review of related works on part of speech tagging. Different languages are considered including Afaan Oromoo, which is the focus of this study.

At first, part of speech tagging of a hybrid approach, by applying combination of the rule based and statistical approaches for Turkish language has been reviewed (Levent et al, 2007). The researchers in this work have made use of some characteristics of the language in terms of heuristics in addition to the combination of the stochastic and rule based approaches. They have used both word frequencies and the n-gram (unigram, bigram, trigram) probabilities. They have combined the Turkish morphological analyzer with stochastic methods in order to improve the accuracy of the system as morphological analyzer helps in guessing the probable tag of words that do not exist in the corpus. The corpus that is used for this study contains 7,200 Turkish sentences and the tag set consists of 13 parts of speech. The researchers divided the corpus into two main parts; the training set that contains about 6,000 sentences (roughly 83% of the corpus) and the test set contains the rest (about 17% of the corpus). Due to the rich derivational morphology of Turkish, they have used the surface forms of words instead of the base (root) words. Hence what they have suggested is, the tagger, given a surface form of a word, is able to determine the part of speech of the surface form of a word.

This study generally finds the tag of a word in two steps: first the statistical analyzer component extracts n-gram probabilities and heuristics data from the training corpus and the n-gram probabilities are calculated based on the sequence of the words using unigram, bigram and trigram equations. Then it considers the arrangement of words in a sentence that means it checks the grammatical structure of the sentence that is to be tagged. The general sentence formation in Turkish is Subject-Object-Verb (SOV) and hence, it is not common to get that the first word of a sentence is a conjunction. In order to test the accuracy of the system, the researchers have performed three experiments by using different parts of the corpus for training and testing set in each. As a result they have got an average accuracy of 84.7%.

Part of speech tagger for Amharic was developed by Mesfin (2001) using a Stochastic HMM approach. In his work, he has developed a prototype simple automatic part of speech Tagger for Amharic language. He has used the Viterbi Algorithm to find the sequence of tags for sequence of words in a given sentence. And a module for sentence splitter was developed in order to facil-

itate the preparation of texts in a file to be tagged with appropriate parts of speech. Moreover, he has also used the bi-gram model in order to find the contextual probability of part of speech tags. POS tags were assigned on the basis of the review made regarding the linguistic properties of the Amharic word classes. The researcher has used 23 tagsets and one page long text as a corpus which he has divided it into training set and test set. Then the experiment was done on the test set as well as on the training set and accordingly the results achieved were as accurate as 97% on the training set and 90% on the test set. Taking into consideration the size of the corpus he has used, it is impossible to conclude that he has got a result that can be mentioned satisfactory as he used a very small size corpus.

The first Amharic part of speech tagger using hybrid (neural network and rule based) approach was developed by Solomon (2008). The researcher has proposed POS system comprising two steps: The first step is that the Amharic text is tagged using the Neural Network approach. Afterwards, the second step is, the tagged text is checked for detection and correction of any anomaly using the rule based approach. He has adapted Back Propagation Algorithm and Transformation based learning method for the development of the Amharic Part of Speech Tagger.

The researcher has collected data from different source such as the Ethiopian Language Research center (ELRC), Addis Ababa. He has collected around 210,000 words from one ELRC project called "The Annotation of Amharic News Document" which is meant to tag each Amharic word in its context with the most appropriate part of speech manually. The project in turn has collected the sentences from Walta Information Center, a private News Agency located in Addis Ababa, Ethiopia, that makes daily news in Amharic and English through its website. Mesfin (2001) has used 23 tag sets and one page long text while Solomon has used 30 taggsets and 210,000 words of text corpus. Moreover, he has used relatively large number of Amharic tagsets and size corpus than the previous works for the Amharic Language.

Solomon(2008) has used both lexical probability and contextual probability to find the most probable tag of a word. The lexical probability is simply the probability of a word occurrence with a specific tag (P(Ti|Wi)) that can be calculated by dividing the occurrence of the number of appearances of the Wi and Ti by the number of occurrences of Wi in the Text corpus. The lexical probability is stored in a lexical probability table for each word. Contextual probability is the transition probability that can be determined by calculating the probability that the tag occurs with n-previous tags. Solomon has stated the tagging of POS to be tackled in two levels: at sen-

tence level and at word level.

To evaluate the proposed method, the researcher has conducted a lot of experiments. Relatively a large number of data is used to train and test the proposed tagger. As a result, the experimental performance of this work indicates that 91% and 94% accuracy for rule-based and neural network tagger, respectively. But the result reaches to 98% when the experiment has been conducted on the hybrid tagger. Though the text corpus taken for this thesis work is not as large size as that of brown corpus etc, it has achieved a higher performance on the hybrid approach.

Another work, part-of-speech tagging for under-resourced and morphologically rich languages the case of Amharic was investigated by Martha (Martha et al, 2011). The paper was conducted to identify the best method of part-of-speech (POS) tagging for under-resourced and morphologically rich languages. For experiment analysis the POS tag-set developed within "The Annotation of Amharic News Documents" project at the Ethiopian Language Research Center(ELRC) has been used. These tag-set has 11 basic classes: nouns (N), pronouns (PRON), adjectives (ADJ), adverbs (ADV), verbs (V), prepositions (PREP), conjunction (CONJ), interjection (INT), punctuation (PUNC), numeral (NUM) and UNC which stands for unclassified and is used for words which are difficult to place in any of the classes. The corpus used to train and test the taggers is also the one developed in the above mentioned project and it consists of 210,000 manually annotated tokens of Amharic news documents. The researchers after doing the pre-processing tasks on manually annotated tokens, for training and testing purposed they used a corpus that consists in 8,075 tagged sentences or 205,354 tagged tokens.

The total corpus has been divided into training, development test and evaluation test sets in the proportion of 90:5:5. The development test set has been used for Parameter tuning and the taggers are finally evaluated on the evaluation test set.

The experiments have been conducted using different tagging strategies and different training data sizes for Amharic. Those tagging strategies investigated in their paper were Disambig, Moses, Conditional Random Field (CRF++), Support Vector Machine Tool(SVMTool), Memory-Based Tagger (MBT) and Trigram 'n' Tags (TnT). In this paper, experiments on word segmentation and tag hypotheses combination have also been conducted to improve tagging accuracy.

The result of the paper showed that MBT is a good tagging strategy for under-resourced languages as the accuracy of the tagger is less affected as the amount of training data increases compared with other methods, particularly TnT. The researchers were also showed that seg-

menting words composed of morphemes that have different POS tags is a promising direction to get better tagging accuracy for morphologically rich languages. The researchers experiment on hypothesis combination showed that HPR and hybrid combination methods are practical to bring improvement in tagging under resourced languages.

Part of speech tagging of Tigrigna using hybrid model was developed by Teklay(2010). A hybrid approach, which is the combination of rule based and HMM tagger, is designed for Tigrigna language part of speech tagger. Teklay was preferred behind the hybrid tagger is, it performs better than the individual ones. He used a corpus with a total of 26, 000 words is collected from different Tigrigna news broadcasting agencies. This means the corpus prepared for this work is only domain news. 36 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 HMM and rule based taggers as a supervised learning approach is used. The tagsets identified does not indicate number, gender, tenses etc.

The researcher divided the corpus into two (training set and testing set) for testing and training purpose. The training set consists 75% of the corpus (around 20,000 words) and the testing set consists 25% of the corpus (around 6000 words).

Hence, different experiments are conducted for the three types of taggers namely the HMM tagger, the rule based tagger and Hybrid tagger. Accordingly, 89.13%, 91.8% and 95.88% performances are obtained for HMM, rule based and hybrid taggers respectively. At the end the researcher concluded that the hybrid tagger for Tigrigna language performs better than HMM tagger and rule based tagger used separately.

Another work is done by Abraham(2013) to improve Brill's tagger lexical and transformation rule for Afaan Oromo POS tagging. In his study, he used several tools and techniques in order to achieve the desired goals. To customize Brill's tagging system for Afaan Oromo, Brill's rule based Tagger and NLTK were used. Python and Perl Programming languages are also used for coding as required. For the study 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. 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. Transformation-based Error driven learning are adapted for Afaan Oromoo part of speech tagging. A 10-fold

cross validation method is used for training and testing of the system as it provides a better and reasonably acceptable result. 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 made. The previously adapted Brill's tagger shows an accuracy of 80.08% where as the improved Brill's Tagger result shows an accuracy of 95.6% which has an improvement of 15.52%.

Development part-of-speech tagger using HMM for Afaan Oromo language was also conducted by Getachew(2009). The researcher reviewed literatures on Afaan Oromo grammars and identifying 17 tagset and word categories, the study adopted Hidden Markov Model (HMM) approach and has implemented unigram and bigram models of Viterbi algorithm.

For training and testing purpose 159 sentences (with a total of 1621 words) that are manually annotated sample corpus are used. The corpus is collected from different public Afaan Oromo newspapers and bulletins to make the sample corpus balanced. A database of lexical probabilities and transitional probabilities are developed from the annotated corpus. The performance of the prototype, Afaan Oromo tagger is tested using tenfold cross validation mechanism. The result shows that in both unigram and bigram models 87.58% and 91.97% accuracy is obtained, respectively.

Abraham(2014) in his research, he experimented on the use of one of the state-of-the-art probabilistic model for sequence classification, Maximum Entropy Markov Model(MEMM), to tag Oromo text according to the lexical category. This model assigns the correct tag or part of speech to each word based on the context of the sentence, considering many features. During his work in order to implement the tagging experiment, the collected corpus for the study was manually tagged and the correctness of the tagging was commented by experts of linguistics in Oromo language studies. The tagging process of his model is based on the 33 identified tagset and the corpus that is manually tagged, considering contextual position of words in a sentence. The researcher used tagged corpus is for training the model and testing its performance.

The researcher was split his data into contiguous sections for training and testing. The total corpus consists of 452 sentences (total of 6094 tokens). Nine tenth of the corpus was used to train the model and one tenth of the corpus was used for testing the model in tenfold cross validation. Abraham Tesso used python programming language with a python-based Natural Language Processing Toolkit (NLTK) for implementation. The average performance of the model was found

to be good with 93.01% accuracy on test data, which is very much promising, result to be recognized even with such limited resources and corpus.

CHAPTER THREE

OVERVIEW OF AFAAN OROMOO

3.1 Introduction

Afaan Oromoo is one of Afro-Asiatic languages most widely spoken of the Cushitic family and it is the working language of Oromia region. Oromia is one of the Regional States in the current Federal Government of Ethiopia that is mainly occupied by Oromo People. Oromo People speak their own native language known as Afaan Oromoo. It is one of the major indigenous African languages that is widely spoken and used in most parts of Ethiopia and some parts of the neighboring countries. The language is spoken by 36.7% of Ethiopian population according to 2007 Ethiopian census data and also this language is spoken in neighboring countries Such as Kenya, Somalia, Eritrea, and Djibouti, by close to 40 million people, making it Africa's the fourth most widely spoken language after Hausa, Arabic, and Swahili(Mekuria, 1994)(Census, 2007)(Bichaka, 2004).

Afaan Oromoo besides being the working language of Oromia state, currently it is the instructional medium for primary and junior secondary schools throughout the region and its administrative zones. Moreover, a number of literature works, news papers, magazines, education resources, official documents and religious writings are written and published in Afaan Oromoo. It is also a language spoken in common by several members of many of the nationalities like Hararі, Anuak, Barta, Sidama, Gurage, etc., who are neighbors to Oromoo people.

Like most of other African and Ethiopian languages, Afaan Oromoo has a very rich morphology. It has the basic features of agglutinative languages where all bound forms (morphemes) are affixes. In agglutinative languages like Afaan Oromoo most of the grammatical information is conveyed through affixes (prefixes, in fixes and suffixes) attached to the stems. Both Afaan Oromoo nouns and adjectives are highly inflected for number and gender(Baye, 1981). For instance, in comparison to the English plural marker s (-es), there are more than 12 major and very common plural markers in Afaan Oromoo nouns (e.g.oota, ooli, -wwan, -lee, -an, een, -oo, etc.)(Gumii, 1995)(Adugna, 2004).

Afaan Oromoo verbs are also highly inflected for gender, person, number and tenses. Moreover, possessions, cases and article markers are often indicated through affixes in Afaan Oromoo (Baye, 1981).

3.2 Morphology of Afaan Oromoo

In linguistics morphology refers to the mental system involved in word formation or to the branch of linguistics that deals with words, their internal structure, and how they are formed(Mark & Fudeman , 2011).What we have been describing as 'elements' in the form of a linguistic message are technically known as 'morphemes'. Morpheme is a unit of grammatical function includes forms used to indicate past tense or plural. English word forms such as talks, talker, talked and talking must consist of one element talk, and a number of other elements such as -s, -er, -ed and -ing. All these elements are described as morphemes. In Afaan Oromoo words are also formed from two or more morphemes. There two types of morphemes: free and bound morphemes. Free morphemes are that can stand by themselves as single words and bound morphemes are words that cannot normally stand alone and are typically attached to another form. In Afaan Oromoo words: nama, namummaa, namni and namoota consists of bound morphemes such as –a, , -oota, -n, and i. Those bound morphemes also divided to inflectional and derivational morphemes. An inflectional morpheme never changes the grammatical category of a word. However, a derivational morpheme can change the grammatical category of a word (George, 2006).

Some examples of frequent gender markers in Afaan Oromo includes: -eessa/-eettii, -a/-ttiior-tu. For instance, singular noun obboleessa, i.e. obbol + eessa(M, brother) vs. singular noun obboleettii, i.e. obbol + eettii(F, sister) can be formed from the same stem.

Likewise, Afaan Oromo adjectives have cases, person, number, gender and possession. Afaan Oromo verbs are also highly inflected for gender, person, number, tenses, voice and transitivity. Obviously, these extensive inflectional features of the language are presenting various challenges for part of speech tag tasks in Afaan Oromo. In inflectional language even though words are highly inflected for gender, person, number and tenses word meanings/word class are not changed. But inflected languages are very challenge to give part of speech to their correspondence words unless morphological analyzer used.

Words are the fundamental building block of a language. Every human language spoken, signed or written is composed of words. Every area of speech and language processing, from speech recognition to machine translation, text to speech, spelling and grammar checking to language-based information retrieval on the Web, requires extensive knowledge about words that are heavily based on the lexical knowledge. In contrast to other data processing systems, language processing applications use knowledge of the language.

The basic processing step in tagging consists of assigning POS tags to every token in the text with a corresponding POS tag like noun, verb, preposition, etc., based both on its definition, as well as its context. The number of part of speech tags in a tagger may vary depending on the information one wants to capture.

3.3 Afaan Oromoo Sentence Structure

Afaan Oromoo 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. At the end the punctuation mark are added to show the termination of the sentence. For 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 as to know the relationship between words, which in turn lets us to categorize them correctly. For instance:

- Caalaan/NP mana/NN ijaare/VV ./PN | Chala/NP built/VV the/DT house/NN ./PN |

Typically, indirect objects follow direct objects. Afan Oromo has both prepositions and postpositions, though postpositions are more common.

3.4 Afaan Oromoo word classes

Word classification is a classification of word based on word semantic or semantic coherence rather than synthetic meaning of the word in a sentence. This classification depends on the word's contribution and meaning in a sentence. Semantic word classification or coherence for

noun can be name of place, people, things or places. This means, word indicating name of place, things, people or places are categorized as noun whatever the role it plays in the sentence. In this work we discussed some of the basic Afaan Oromoo word classes, which are standard for most Linguists. These are noun (Maqaa), pronoun (Bamaqaa), adjective (Ibsamaqaa), verb (Xumura), adverb (IbsaXumura), conjunction (Walqabsiistota), preposition (Durduuba) and Interjection (Rajjeeffannoo). This classes of words represents Afaan Oromoo part of speech that used to classify words in a sentence (Abraham, 2013) (Mohammed, 2010)(Jurafisky & Martin, 2006).

3.4.1 Afaan Oromoo Nouns(Maqaa)

Afaan Oromo nouns are words used to represent 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) (Adugna, 2004)(Baye, 1981).

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

Proper nouns (maqaa dhunfaa) are nouns that represent a unique entity like a specific person, place, river, building and country. In Afaan Oromoo language Proper nouns are used in capitalized. For examples;

- Jimma, Ambo and Nekemt
- Didessa, Dabana and Awash

The first example is when noun used to represent a place and on the second nouns represent river.

Common nouns are nouns which describe an entire group of entities. Common nouns are those nouns that have similar characters and grouped under the same entity. The group is a single unit, but it has more than one member. For instance: the word nama "people" may it can be represent male, female, child and young. Also there are word such as "mana", "barsiisaa", "saree" and "muka" are used as a common noun. Common nouns in Afaan Oromoo sentence can be used as follows:

- Boonaan **barataa** cimaadha | Bona is the clever **student** |

The underlined word is representing the common nouns. The common nouns in Afaan Oromoo can use affixes to show the plurality of nouns and this behavior of adding affix make it different from personal and collective nouns.

Collective nouns are nouns when a group of different entities described by one entity. For example the word forest "bosona" and food "nyaata" are used as collective noun in Afaan Oromoo.

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.

Nouns in Oromo are treated as either male or female, though there are typically no gender markers in the words themselves. Gender can be shown through a demonstrative pronoun, a definite article, a gender-specific adjective, or the verb form (if the noun is a subject). The notable exceptions are those nouns derived from verbs, where the masculine noun adds an -aa suffix and the feminine noun adds a -tuu suffix to the verb root. For example the English word teacher for masculine " barsiisaa" and for feminine "barsiistuu". Also for English word student to masculine we use "barataa" and for feminine we use "barsiistuu".

In written Oromo, plural forms tend to be more common, and may occur with numbers, adjectives, and other indicators. When a plural noun in modified by an adjective, only the adjective shows plurality.

When the plural form of noun is used, there are several forms it may take. Typically, the final vowel is dropped and the correct suffix attached: -oota, -toota, -lee, -een, -yyii, -wwan, -ootii, or -olii. Unfortunately, the correct suffix cannot be predicted from the noun, meaning plural forms must be learned individually. Plural forms also vary across dialects, and multiple forms may be correct for some words. The most common suffix is -oota. The following table shows that some Afaan Oromoo words with their suffixes.

English	Singular	Plural
Tooth	Ilka	ilkaan
Thing	waanta	Waantoota
Day	guyyaa	Guyyawwan, guyyoota

Mountain	Gaara	Gaarreen
River	Laga	Laggeen
Tree	muka	Mukkeen
Year	Waggaa	Waggottii
Book	kitaaba	Kitaaboli, kitaaboota

Table 3.1 Nouns with their suffix

Where English uses "the" to indicate definiteness (a specific something of shared knowledge), Oromo drops the final vowel and uses the suffix - (t)icha for masculine nouns and -(t)ittii for feminine nouns. Making a noun definite is less common in Oromo than in English, and is used only for objects known to both the speaker and the listener. A noun can be either definite or pluralized, but not both. A definite noun is therefore ambiguous in number, and context determines if it is singular or plural. Definite nouns are not modified by demonstrative pronouns or possessive pronouns. If modified by an adjective, the definite marker is attached to the adjective. For example the "man" the base noun (dictionary form) in Afaan Oromoo is "nama" the definite noun is namicha means "the man (men)".

Indefiniteness is marked in English by "a (n)" or "some", while Oromo tends to use the noun alone without modification. The word tokko ("one") is used to indicate "a certain" something, and tokko-tokko can be used to mean "some". For Examples:

- Kitaaban barbaada | I want a book (any book) |
- Kitaaba tokkon barbaada | I want a (certain) book |
- Kitaaba tokko-tokkon barbaada | I want some books |. If we want to use the definite noun we can say | kitaabichan barbaade | that means the final vowel " a" dropped and "ichan" added to stem word.

Afaan Oromoo nouns also change from accusative to the Nominative noun**.** That is, the form of a noun (declension) changes depending on its role (case) in the sentence. The main cases are nominative (for subjects), accusative (direct objects), genitive ("of" indirect objects), dative ("for", "to", "in order to" indirect objects), instrumental ("with", "by" indirect objects), locative ("at"

indirect objects), and ablative ("from" indirect objects). Nouns in Oromo are listed as direct objects (accusative case) in dictionaries.

To change a noun from the accusative (acc.) to the nominative (nom.), certain patterns are used. Nouns in the acc. that end in a single consonant and short vowel will drop the final vowel and add -ni as a suffix. So that the dictionary form of "person (acc.)" is nama, while "person (nom.)" is namni. If the acc. form ends in a double consonant and short vowel, the vowel is replaced by -i. For example, "honey (acc.)" is damma, while "honey (nom.)" is dammi. This applies to all masculine definate nouns, where the -icha suffix in the acc. becomes -ichi in the nom.

If the acc. form ends in a long vowel, -n in is suffixed to form the nom. For example, "name (acc.)" is maqaa and "name (nom.)" is maqaan. More examples:

English meaning	Accusative (dictionary) form	Nominative (subject) form
Actress	Taatuu	Taatuun
Air	Qilleensa	Qillensi
things	Waantoota	Waantooti

Table 3.2 Nouns words in Accusative and Nominative form

3.4.2 Afaan Oromoo pronouns(Bamaqaa)

In grammar, a pronoun is defined as a word or phrase that may be substituted for a **noun** or noun phrase, which once replaced, is known as the pronoun's antecedent. Its can do everything that nouns can do. This is used in place of noun in most cases when a noun is pre stated. A pronoun can act as a subject, direct object, indirect object, object of the **preposition**, and more.

Without pronouns, we'd have to keep on repeating nouns, and that would make our speech and writing repetitive, not to mention cumbersome. Most pronouns are very short words.

In the following sentence we can see a noun and their corresponding pronoun in Afaan Oromoo language

- Calaan mana barumsa deeme. Inni mana barumsa deeme.

- Toltuun naato-qabettiidha. Isheen naato-qabettiidha.

The underline word show that when the pronoun represent or used instead of nouns or as a noun. In Afaan Oromo, there are different categories of pronoun, which includes Personal pronoun (bamaqaa Ramaddii), demonstrative pronoun(bamaqaa akeeektuu), double pronoun(bamaqaa mirree), Possessive pronoun(bamaqoota qabeenyaa), Reflexive pronoun(bamaqaa ufinaa), reciprocal pronouns (bamaqaa Waliyyoo), relative pronoun(bamaqaa firomsee), Interrogative pronoun (Bamaqoota iyyafannoo) and indefinite pronoun(bamaqaa waalleyyuu)(Adugna, 2004)(Gumii, 1995)(Rippaabiliika, 2005).

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 is rarely used (and considered rude) and only the polite form of "you", isin, is used.

The personal pronouns as subjects and direct objects are listed below along with possessive markers.

Subject Pronouns		Direct Object Pronouns		Possessive Pronouns	
I	Ani	Me	Na	My. mine	Koo
We	Nuti, nu'i	Us	Nu	Our, ours	Keenya
You	Ati	You	Si	Your, yours	Kee
You(pl.)	Isin	You(pl.)	Isini	Your, yours(pl.)	Keessan(i)
He, it	Inni	Him,it	Isa	His, its(i)	Isaa
She	Isheen	Her	Ishee	Her, hers	Ishee
They	Isaan	Them	Isaanii	Their, theirs	Isaani

Table 3.3 Afaan Oromoo common pronouns

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 is rarely used (and considered rude) and only the polite form of "you", isin, is used.

Demonstrative pronouns are used to express "this", "that", "these", and "those", in Afaan Oromoo which means "kun" ,"sun", "kunnin", and "sunniin"

Demonstrative pronouns can be combined with pronouns and adjectives to express ideas such as "this one" or "that big one".

Words	Accusative (direct object)	Nominative (subject)
the red one	isa diimaa	inni diimaan
the red ones	isaani diimaa	isaan diimaan
the (many) red ones	isaani didiimaa	isaan didiimaan
this one	isa kana	inni kun(i)
this red one	isa diimaa kana	inni diimaan kun(i)
these ones	isaani kanneen	isaan kunniin
these red ones	isaani diimaa kanneen	isaan diimaan kunniin
that one	isa sana	inni sun(i)
that red one	isa diimaa sana	inni diimaan sun(i)
those ones	isaani sanneen	isaan sunniin
those red ones	isaani diimaa sanneen	isaan diimaan sunniin

Table 3.4 Demonstrative pronouns that can be combined with pronouns and adjectives

Interrogative pronouns are used in questions, and come before the verb and either before or after the subject. Often, if the verb is "is/are", this verb is dropped when using an interrogative pronoun. The main interrogative pronouns are: What "Maal", Why "Maaliif", How "Akkam", When "Yoom", Where "Eessa", From where, "Eessaa", Who "Eenyu", Whose "Kan eenyu", How much, many " Meeqa, hammam" and Which "Kam" are used in Afaan Oromoo.

3.4.3 Afaan Oromoo Adjectives (Ibsa Maqaa)

In English Adjectives are words that describe or modify the nouns words. They can identify or quantify another person or thing in the sentence. Adjectives are usually positioned before the noun or the pronoun that they modify. But in Afaan Oromoo Adjectives describe the nouns and come after the nouns that they modify. In the following examples, the highlighted words are adjectives:

- Isaan mana bareedaa kessa jiraatu | They live in a beautiful house. |
- lisan harra uffata gababa uffachaa jirti | Lisa is wearing a sleeveless shirt today. |

In Afaan Oromoo language adjectives are categorized into basic adjectives (maqibsa bu'uuraa), demonstratives adjectives (maqibsa akeektuu), possessive adjectives(maqibsa qabeenyaa), interrogative adjectives(maqibsa iyyafannoo) and Manner adjectives(maqibsa malummaa)(Adugna, 2004).

Oromoo adjectives can be male, female, or neutral. Masculine adjectives are used with masculine nouns, feminine adjectives modify feminine nouns, and neutral adjectives can be used with any noun. All non-neutral adjectives can be made masculine or feminine by attaching the appropriate suffix. Masculine suffixes for adjectives are: *-aa*, *-aawaa*, *-acha*, and *-eessa*. Feminine suffixes are: *-oo*, *-tuu*, *-ooftuu*, and *-eettii*. Standard morphology rules apply when attaching suffixes.

Examples:		
English meaning	Masculine	Feminine
Adorable	Jaallatamaa	Jaallatamtuu
Beautiful	Bareedaa	Bareedduu

Fast	si'aawaa	si'ooftuu
Sweet	mi'aawaa	mi'ooftuu
Poor	Hiyyeessa	Hiyyeettii

Table 3.5 Afaan Oromoo adjectives in Masculine and Feminine form

Neutral adjectives (e.g., "adii" – "white") use the same form for both masculine and feminine nouns. Plural of Adjectives also formed in Afaan Oromoo When adjectives are used to modify a noun, typically the noun remains in the singular and number is shown by the adjective only. Plural adjectives are formed by repeating the first syllable. For English word "white" Afaan Oromoo singular form is "adii" and a plural form is "adaadii" . And also a english word "beautiful " in Afaan singular form is "bareedduu" and a plural form of Afaan Oromoo "babbaredduu" have been formed

Some masculine adjectives will change their ending to *-oo* when pluralized. Some of these do not repeat the first syllable as a plural marker. Such as the English word "knowledgeable" in Afaan Oromoo singular form is "beekaa" and a plural form is "beekoo". Also a english word " strong " Afaan Oromoo singular form is "cimaa" and plural form is "ciccimoo".

To express an adjective with a pronoun, as in "the black one", one can simply use the correct pronoun in front of the adjective, as in "isa gurraacha". The pronoun used will depend on the its role in the sentence, so that "The black one looks nice" would be "Inni gurraachi gaarii fakkaata", while "I want the black one" would be "Isa gurraacha nan barbaada".

Adjectives show the same case as the noun they modify. Adjectives modifying a subject noun will undergo the same suffix patterns. For Examples:

- Inni diimaan sun meeqa? | How much is that redone |
- Fardichi lamaffaan deeme or Fardi lamaffichi deeme | The second horse went |

For definite nouns, either the noun or the adjective may take the definite suffix, but not both (as in the example above). This suffix will also show case. For instances: "Namni sooressichi dhufe" or "Namachi sooressi dhufe"----"The rich man came" "Obboleessi kan nama sooressicha dhufe" or "Obboleessi kan namicha sooressa dhufe".--- "The rich man's brother came"

On the first and second sentence the adjectives comes with definite "ichi" and "icha respectively; that shows or differentiate the man who riches from the other peoples. A noun modified by more than one adjective will have the only the first adjective show case and definiteness.

3.4.4 Afaan Oromoo verb (Xumura)

Verbs are the type of word class that stands to show action. In Afaan Oromoo verbs always comes at the end of each sentence. Most Oromo dictionaries will list verbs in their infinitive (e.g., beekuu - "to know"), and all infinitives end in -uu. The verb stem is this infinitive form with the final -uu dropped. The stem of beekuu is therefore beek-, and the verb is conjugated by adding suffixes to this stem (e.g., beekti - "She knows")(Gumii, 1995)(Adugna, 2004).
To express actions completed in the past, verbs are conjugated in the simple past tense. That is, if one knows the present-future conjugation pattern for a verb, one can accurately conjugate that verb in the simple past (or any other tense, for that matter). Compared to the present-future affirmative conjugation, only the final vowel changes to form the simple past. The table below shows how to change a verb conjugated in the present-future to the present negative and simple past affirmative.

final vowel in the present affirmative	Ending in present negative	Ending in past affirmative
A	U	E
I	U	E
U	An	an(i)

Table 3.6 Vowels changes a verb conjugated in the present-future to the present negative and simple past affirmative.

So that "you learn" is baratta, "you don't learn" is hin barattu, and "you learned" is baratte. The exceptions to this rule are the "to be" verbs: dha and jiru. These verbs are only used in the present-future tense, and in the past tense are replaced by the verb turuu, which in the present tense means "to stay/wait". To say "it is present" is jira, but "it was present" is ture(Adugna, 2004).

Participles, as known as verbal adjectives, modify nouns based on actions. In English, examples include "the sleeping lion" in Afaan Oromoo "lenca rafaa jiru" (*sleeping* "rafaa jiru" being a present participle) and "the fallen leaves" "balli harca'ee" (*fallen* "harca'ee" being a past participle).

Verbs in the Affirmative can be expressed as follows , for the first person singular (ani) form, the suffix -n (or -an to a consonant) must be added to the word preceding the verb, or the preverb nan must be used to express the verb in the affirmative. In speaking, the first method is the most common.

For Example; in the following sentence Jimma added the "n" suffix to show the affirmativeness:

- Jimman jiraadha or Jimma nan jiraadha | I live in Jimma |
- Nyaachuun barbaada or Nyaachuu nan barbaada | I want to eat |

For other forms, an optional preverb "ni" may be used. Typically, if there is no object in the sentence, the ni is mandatory. For instance; when "ni" used in a sentence can written as follows:

- Baajajii ni barbaadda? or Baajajii barbaadda? | Do you want a bijaj? |

Also we can express negativeness in the verb in Afaan Oromoo language. To express "not/don't/doesn't" in Oromo the word "hin" is added before the verb (either as an attached prefix or as a separate word), and the last vowel in the verb conjugated in the affirmative changes as follows: a → u, i → u, u → an. For example: the word "to go" which means "Deemuu" is written as "do not go" to express the negativeness but during Oromoo we can write as "hin deemuu". When we express with pronoum can be written as follows.

- Ani nan beeka – "I know" Ani hin beeku – "I don't know"
- Isaan ni deemu – "They go" Isaan hin deeman – "They don't go"

- Isheen ni dandeessi – “She can” Isheen hin dandeessu – “She can't”

The exception to this is the negative form of/ dha, which is miti meaing “am not/are not/is not”. Like dha, miti does not conjugate for person or number. For Examples:

- Rakkoo miti | It's not a problem |
- Sun kitaaba koo miti | That is not my book |

3.4.5 Afaan Oromoo Adverb (Ibsa xumura)

An adverb is a word that is used to change or qualify the meaning of an adjective, a verb, a clause, another adverb, or any other type of word or phrase with the exception of determiners and adjectives that directly modify nouns. In Afaan Oromoo Adverbs comes before a verb and it used to modify verbs. Traditionally considered to be a single part of speech, adverbs perform a wide variety of functions, which makes it difficult to treat them as a single, unified category. Adverbs normally carry out these functions by answering questions such as When, how, Where, in what way/how and to what extent/how much. For example;

- Abdisaan Nakemtee kaleessa dhufe. | Abdisa came from nekemt yesterday |
- Sanyiin magartuu bayye jaallata. | sanyi loves magartu very much |

In Afaan Oromoo there are different categories of adverb such as adverb of manner, place, time and degree.

Adverbs of manner can be formed from a verb, by putting the modifying verb in the simple past, or from an adjective or noun, by using the locative or instrumental declension.

- Laliseen ariitidhan deemte. | lalise went quickly |
- Gammachis suuta deemu jaallata. | Gamachis likes to go slowly |

The underlined words answered that how lalisee went and it show that adverbs of manner. Adverbs of times can be used in Afaan Oromoo to answer for the question “when?”. This shows that at what time and when actions are done.

- Bashatun kaleessa deemte | Bashatu went yesterday |
- Jiraatan hamma dhufe | Jirata came now |

The underlined words show that adverb of time it answer for the question when bashatu went.

During Adverb of place the word that show place comes before the verb and suffixes that show place are added to the accusative (dictionary) form. For instance; the place name "Jimma" is added the "-rraa" which is equivalent of English word "from" and written as "jimmarraa" in the sentence. We can write in the sentence as follows:

- Ji'a kamitti Jimmarraa deebiita? | In what month do you return from Jima? |
- Mana kootti si affeeruun barbaade. | I'd like to invite you to my house. |

Adverbs of degree in Afaan Oromoo the question of how much or how often actions are done. For instance:

- Amalli Boonaa baay'ee gaarii dha | Bona has very good conduct | The word "baay'ee" or "very" shows adverbs of degree.

3.4.6 Afaan Oromoo Preposition (Durduube)

A preposition links a noun to an action or to another noun. It link nouns to other parts of the sentence. Oromo prepositions divided into two categories: true prepositions and postpositions, with true prepositions coming before the noun and postpositions coming after the noun they relate to (Adugna, 2004) (Gumii, 1995).

- Keeniyaan Itoophiyaarraa (gara) kibbatti argamti | Kenya is located (to the) south of Ethiopia |
- Rooberan Boqonnaarra jira | Robera is on vacation |

From the above examples, we can notice that the postpositions **itti**, **irra**, and **irraa** most often occur as suffixes, *-tti*, *-rra*, and *-rraa*, on the nouns they relate to. In Afaan Oromoo the use of postpositions is preferred and occurs with a higher frequency than the use of Prepositions.

Some Common Prepositions and Postpositions of Afaan Oromoo identified as on the work of (Adugna, 2004)(Gumii, 1995)are presented using the table as follows:

Jidduu/gidduu — middle, between	haga, hanga — until
keessa — in, inside	hamma — up to, as much as
malee — without, except	akka — like, as
wajjin — with, together	waa'ee — about, in regard to
gubbaa — on, above	Irratti __ on it
fuuldura — in front of	gara — towards
gad(i) — down, below	eega, erga — since, from, after
ol(i) — up, above	
ala — out, outside	
bira — beside, with, around	
booda — after	
cinaa — beside, near, next to	
dur, dura — before	
duuba — behind, back of	
irra — on	
irraa — from	
itti — to, at, in	
jala — under, beneath	

Table 3.7 Afaan Oromoo prepositions

3.4.7 Afaan Oromoo Conjunction (Wal qabsiistu)

Whereas prepositions link nouns to other parts of the sentence, conjunctions usually link more complete thoughts together. A word that can be used to join or connect two phrases, clauses and sentences is known as a conjunction. In Afaan Oromoo Conjunction can be divided into coordinating and subordinating conjunctions.

Coordinating conjunctions are used to connect two independent clauses. Mostly, the-

se conjunctions are used when the speaker needs to lay emphasis on the two sentences equally.

- Innis gara biyya isaa dhufe, sabni isaa garuu isa hin simanne. | And he came to his country, but his people did not accept him [John 1:11]. |
- Nyaatan barbaada sababiinsa nan beela'e. | I want food because I am hungry. |

Also there are different types of conjunctions in Afaan Oromo some of them are fi " and" ,illee "also", waan " because", garuu 'but', moo/yookiin 'or', kanaafuu 'therefore', 'haa ta'u malee' however/so' , ta'u illee 'even though', ta'us "though" and etc.

3.4.8 Afaan Oromoo Introjections (Raajii)

Introjections are words we can use to express our feeling, emotions for suddenly happening of situations. Afaan Oromoo also has own introjections words as other language Amahric and English.

Those words are ishoo for showing happiness *wayyoo* for sadness *ah* for silent event or situation happened(Adugna, 2004)(Gumii, 1995)

- Ishoo! Baga gammadde. This means 'wow! congradulation!'

The word "**ishoo"** is used to express pleasure of something or for showing happiness and is used as introjections word.

3.4.9 Afaan Oromoo Numeral (Lakkoobsa)

Numbers come after the noun they modify, so that "four" is "Afur" "three Orange" is "Burtukana sadi", just as "two birr" is "qarshii lama" and "three hundred" is dhibba sadi. Also in Afaan Oromoo ordinal numbers are used to express the rank of something. This Ordinal numbers are formed by adding the suffix -ffaa or -affaa to the number. For examples : "tokkoffaa" is " first", "lammaffaa" is " second", "sadaffaa" is " third" and continued in such likes.

3.5 Afaan Oromoo Tagset

Tagset is the set of tags from which the tagger is supposed to choose to attach to the relevant word. These tagset is different from language to language its prepared based the nature of language(Jurafisk & Martin, 2006). Tags are the labels used for adding more information concerning the lexical category of each word in a sentence. Different researcher tried to identify tagset for Afaan Oromoo language with language experts. The first researcher who tried to develop POS tagger for the language is Getachew(2009). He identified 17 tags and tagged the corpus based on the words meaning in the contextual of sentence.

The second work of Afaan Oromoo part of speech tagging was done by Mohammed (Mohammed, 2010), during his work he identified negative indicator "hin" then add to Getachew's work. Totally Mohammed tagged the corpus by 18 tagset which means including basic tagset and their derived tagset. Also on the work of Abraham(2013) 8 tagset were identified and add to the work of (Mohammed, 2010). Around 26 tagset were prepared and tagged the corpus on the work of (Abraham, 2013). The identification of the tags is made by taking 11 basic word classes namely: noun, pronoun, verb, adjective, adverb, preposition, conjunction, numerals, punctuation, introjections, and negation as basic tags and others are derived from these basic classes or combination of basic classes.

In the same fashion nine (9) tags were identified, which is not used by any previous researcher in Afaan Oromoo POS tagger. These nine tags identified during this work are: Possessive pronoun(PP), Interrogative pronoun(PI), Interrogative pronoun + preposition(PIP), Interrogative pronoun + conjunction (PIC), Indefinite pronoun(PID), Reciprocal pronoun(PRE) and Independent preposition(PRI). Totally thirty five (35) tags were used in this thesis to tag the words to their tags. The total tagset used in this thesis work are presented in Table 3.8.

S/NO	***Basic category/tag***	***Derived category/tag***	***Description***	***Example***
1.	Noun	NN	Noun	Nyaata
2.		NNDS	Noun + Definiteness	Namtichi/namticha

3.		NNP	Noun + plural	Garreen Mukeen
4.		NPROP	*Proper noun*	Hundee ,Caalaa Lalisa
5.		NC	Noun + conjunction	Jaallannee-ffii
6.		NP	Noun + Preposition	Arsii-tti
7.	Pronoun	PP	pronoun	Isii/Ishee ,Isa Isaan
8.		PS	Preposition + pronoun	Isii-tti
9.		PC	Pronoun + conjunction	Isii-fi
10.		PREF	Reflexive pronoun	Ofii, walii
11.		PD	Demonstrative pronoun	Kuni, suni
12.		PP	Possessive pronoun	Koo,Keenya, Kee, Kessan
13.		PI	Interrogative pronoun	Akkam , Eenyu, Eessaa, Kam, Maal
14.		PIP	Interrogative pronoun + preposition	Akkam-itti
15.		PIC	Interrogative pronoun + con-junction	Maal-iif
16.		PID	Indefinite pronoun	Eenyuyyuu Eessayyuu

				Yoomiyyuu Meeqayyuu
17.		PRE	Reciprocal pronoun	Wal, Walumaan
18.		PDPR	Preposition + demonstrative pronoun	*Sunis-s*
19.	Verb	VV	verb	Barresite Dubise
20.		AX	Auxiliary	Ta'e , Rafe
21.		VC	Verb + conjunction	Beekti , Beekna Hinbeeknu
22.	Adjective	JJ	Adjective	Guddaa Cimaa
23.		JC	Adjective + conjunction	*Qaloo-fi*
24.		JP	Preposition + adjective	Gudda -tti
25.	Adverb	AD	adverb	Dilbata, wixata
26.		ADVPREP	Preposition + adverb	Dilbata -rraa
27.		ADVC	Adverb + conjunction	Dilbataa-fii
28.	Prepositions	PR	Preposition	-tti, oolee
29.		PRI	Independent preposition	Waa'ee, Wajjin keessaa
30.	Conjunction	CC	conjunction	Fii, yookan
31.	Numerals	JN	Cardinal Number	Tokko, lama

32.		ON	Ordinal number	Tokkoffaa, sa-daffaa
33.	Punctuation	PN	Punctuation	.,!?
34.	Interjection	II	Interjection	Ishoo, wayyoo
35.	Negation	NG	Negation	Hin

Table 3.8 Afaan Oromoo identified tagset

Standard Data corpus is an important component in every tasks of Natural language processing, even though their format and behavior is different in every tasks of NLP. Our data corpuses taken from the work of Abraham (Abraham, 2013) are totally 817 tagged sentences. But we made further correction with identified tagset and added 700 tagged data sentences, which is prepared for this work. In this thesis, totally 1517 tagged data sentences are used for experimental analysis.

The total corpus is divided into training set and testing set to train and test the tagger. Since our data corpus is very small compared to brown corpus in the NLTK, we used 85% of our data totally 1289 sentences for training and the remains 15% totally 228 sentences are used for testing purpose.

CHAPTER FOUR

DESIGN OF AFAAN OROMOO POS TAGGER

4.1 Introduction

Part-of-speech tagging is labeling the words in a text with a corresponding part of speech. POS tagging, which is also called word-category disambiguation or grammatical tagging, is based on both the meaning of the word and its context. Context is the relationship of the word with adjacent and related words in the phrase, sentence or paragraph that it exists in (Berna & Özlem, 2009). POS tagging is a well-studied problem in the field of NLP and one of the fundamental processing step for any language in NLP and language automation.

In this chapter, the design and implementation HMM, rule based and the hybrid approach was discussed in details. The flow chart diagram of transformation based approach, hidden Markov models and the hybrid of Afaan Oromoo POS tagger was clearly presented.

4.2 Approaches and Techniques

Many algorithms have been applied for part of speech tagging including hand-written rules (rule-based tagging), probabilistic methods (HMM tagging and maximum entropy tagging) and Artificial neural network, as well as other methods such as transformation based tagging, memory-based tagging and combination(hybrid) approaches (Jurafisky & Martin, 2006).

In this work the hybrid approach which combines rule based and HMM was used. In this thesis rule based algorithm the transformation-based learning approach which extracts rules from a corpus was used. Transformation-based learning (TBL) is a rule-based algorithm for automatic tagging of parts-of-speech to the given text. TBL transforms one state to another using transformation rules in order to find the suitable tag for each word. From the statistical approach the HMM tagger was selected to tag the words based on the most probable path of the word on a given sentence. The detail application and flow chart of the taggers presented in the sections.

4.3 Hidden Markov Models

The Hidden Markov Model is one of the most important machine learning models in speech and language processing. In order to define it properly, we need to first introduce the Markov chain, sometimes called the observed Markov model. Markov chains and Hidden Markov Models are both extensions of the finite automata.

A Markov chain is useful when we need to compute a probability for a sequence of events that we can observe in the world. In many cases, however, the events we are interested in may not be directly observable in the world. For example, in part-of speech tagging we didn't observe part of speech tags in the world; we saw words, and had to infer the correct tags from the word sequence. We call the part-of speech tags hidden because they are not observed. A Hidden Markov Model (HMM) allows us to talk HIDDEN MARKOV about both observed MODEL events (like words that we see in the input) and hidden events (like part-of-speech tags) that we think of as causal factors in our probabilistic model.

The overall approach and flow chart diagram of the HMM tagger have been used in this thesis was presented in the below section of the study.

4.3.1 Training of the HMM Tagger

The next important step is training the HMM tagger. The training method we describe here is based on supervised learning approach. It runs on the corpus, makes use of tagged data and estimates the probabilities of transition, P(tag | previous tag) and observation likelihood P(word | tag) for the HMM.

Then the transition probability $P(t_i|t_i\text{-}1)$ is calculated simply by using the equation of (2.9) of Chapter 2, where $c(t_i\text{-}1,t_i)$ is the count of tag sequence ti-1,ti in the corpus.

For calculating observation likelihood probability $P(w_i|t_i)$, we calculate the unigram (unigram model uses only one piece of information, which is the one that is considering) of a word along with its tag assigned in the tagged data. The likelihood probability is calculated simply by the equation of (2.10) of chapter 2, where $c(t_i,w_i)$ is the count of word i (w_i) is assigned tag i (t_i) in the corpus.

The calculated transition probability and likelihood probability is given to vaterbi matrix calculator to calculate the probability of the most-probable path of tags sequence in a given corpus.

4.3.2 The Viterbi Algorithm

The Viterbi algorithm is widely used in the NLP applications that allows considering all the words in the given sentence simultaneously and computes the most likely tag sequence.

The applicability of the viterbi algorithm stated by Jurafisky and Martin as follows:

> For any model, such as an HMM, that contains hidden variables, the task of determining which sequence of variables is the underlying source of some sequence of observations is called the decoding task. The Viterbi algorithm is DECODING perhaps the most common VITERBI decoding algorithm used for HMMs, whether for part-of-speech tagging or for speech recognition(Jurafisky & Martin, 2006).

The term Viterbi is common in speech and language processing, but this is really a standard application of the classic dynamic programming algorithm, and looks a lot like the minimum edit distance algorithm.

4.3.3 Flow Chart of Afaan Oromoo HMM Tagger

Figure 4.1 shows the overall flow chart diagram of the HMM tagger, which is a two-step process that first runs through the tagged corpus and extract the linguistic knowledge. Then it runs through the raw text inputs and generating the best tag sequence for the sequence of input words based on the knowledge that is gathered from the corpus. The tokenization modules Run through the tagged corpus, separate out the words and tags, prepare for probability calculation. And the probability calculations calculate the transition probability and the observation likelihood probability for each pairs of Words, Tag sequences in the corpus as explained in equation of 2.9 and 2.10 in chapter 2 respectively. The result of transition probability and observation likelihood probability is given to Viterbi Matrix Analyzer. The viterbi matrix analyzer Prepares a state graph that has all possible state transitions for the given text input, calculate and assign state transition probability for each transition in the matrix. The Tag Sequence Analyzer back trace the viterbi matrix, analyse the maximum probability path and assign tags to each word in the sentence based on highest probability. The above descriptions were represented using flow chart diagram as follows:

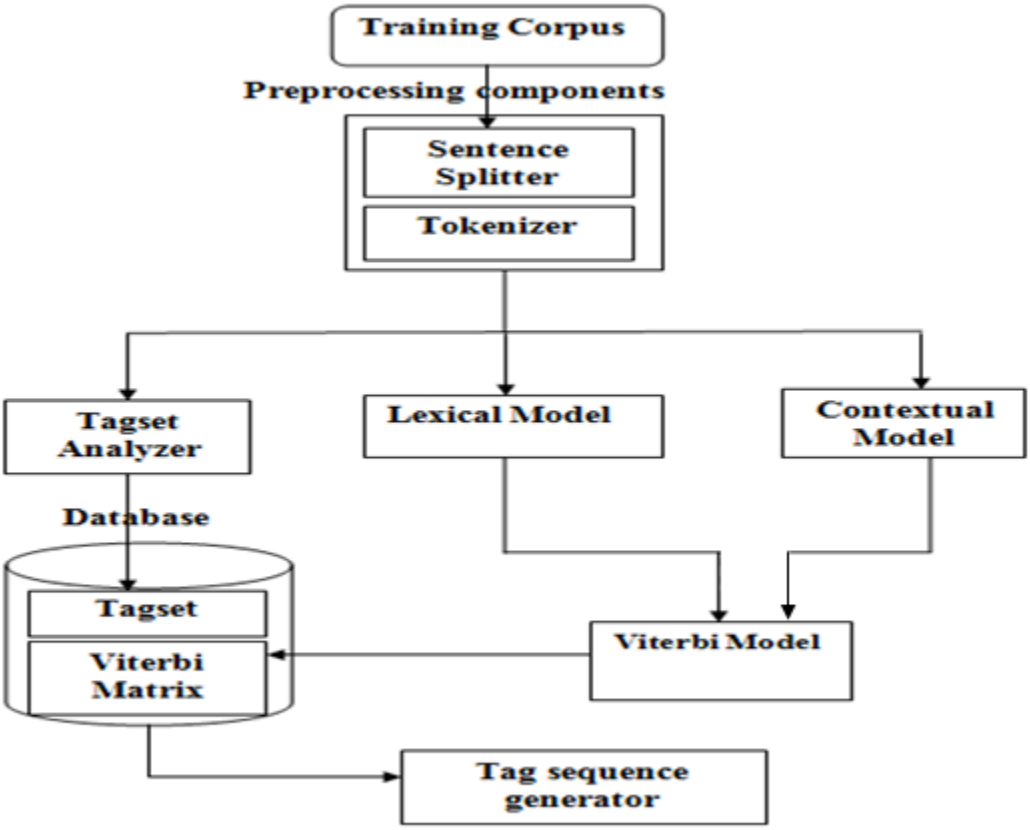

Figure 4.1 Flow chart diagram of Afaan Oromoo HMM tagger

In the flow chart diagram the lexical model contains the likelihood probability of the word in training corpus and contextual model contains the transition probability the word in the training corpus. These probabilities are given to viterbi model to select the most probability path of the word from the two models. At the end the viterbi matrix analyzes the most probable path and assign the word with its tags and tag sequence generator generate the word with assigned tags.

4.4 Transformational Based-Error Driven Approach

It has recently become clear that automatically extracting linguistic information from a sample text corpus can be an extremely powerful method of overcoming the linguistic knowledge acquisition bottleneck inhibiting the creation of robust and accurate natural language processing systems. Eric Brill introduced a POS tagger in 1992 that was based on rules, or transformations,

where the grammar is induced directly from the training corpus without human intervention or expert knowledge. The only additional component necessary is a small, manually and correctly annotated corpus - the training corpus - which serves as input to the tagger. The system is then able to derive lexical/morphological and contextual information from the training corpus and 'learns' how to deduce the most likely part of speech tag for a word. Once the training is completed, the tagger can be used to annotate new, unannotated corpora based on the tagset of the training corpus (Brill, 1992) (Megyesi, 1998).

The tagger does not use hand-crafted rules or prespecified language information, nor does the tagger use external lexicons or lists of different types. According to Brill (1992) 'there is a very small amount of general linguistic knowledge built into the system, but no language-specific knowledge'.

This inducing grammar directly from the training corpus without human intervention or expert knowledge make the TBL is more interesting and reduce the error rate that occur during giving a rule by hand. In addition to that, it saves a time and man power to give this rules.

To use the aforementioned advantage the work of (Abraham, 2013)(Muhammed, 2010),(Teklay, 2010) were used the TBL approaches for part of speech tagging for their local language.

The general framework of Brill's corpus-based learning is so-called Transformation-based Error-driven Learning (TEL). The name reflects the fact that the tagger is based on transformations or rules, and learns by detecting errors. The general description of how TEL works in principle and a more detailed explanation for the specific modules of the tagger is presented as follows as in the reference of (Brill, 1995)(Megyesi, 1998).

The TEL, shown in Figure 4.2, begins with an unannotated text as input which passes through the initial state annotator. It assigns tags to the input using most likely tag of (W_i, T_i). The output of the initial state annotator is a temporary corpus which is then compared to a goal corpus which has been manually tagged. 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.

The tagger uses TEL twice: once in the lexical module deriving rules for tagging unknown words, and once in the contextual module for deriving rules that improve the accuracy. Both modules use two types of corpora: the goal corpus, derived from a manually annotated corpus,

and a temporary corpus whose tags are improved step by step to resemble the goal corpus more and more.

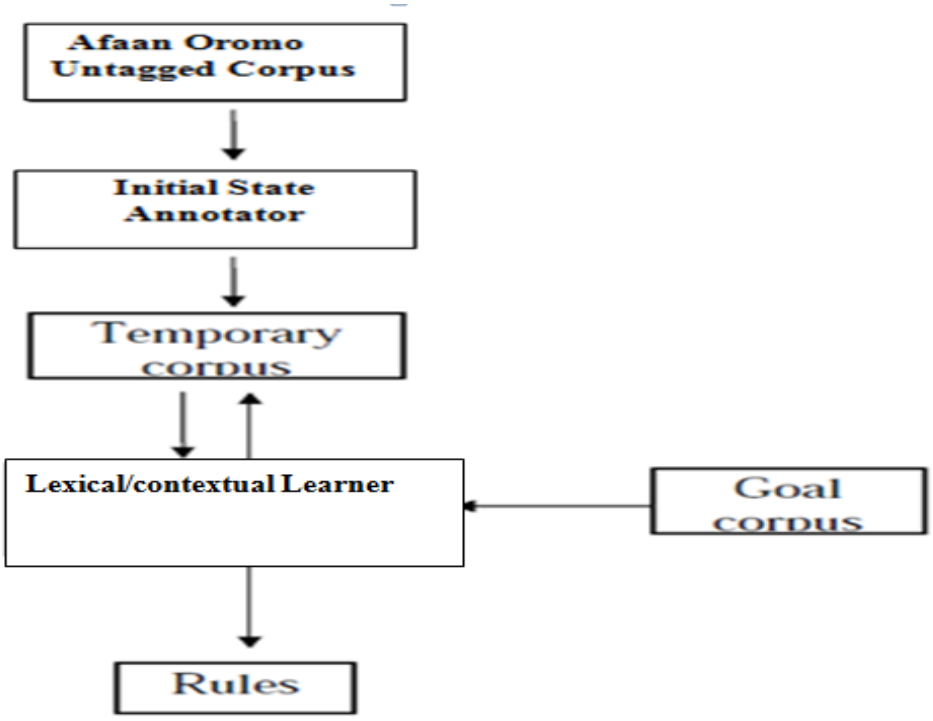

Figure 4.2 Transformation-based error driven approaches

In the lexical module of the tagger the goal corpus is a list of words containing information about the frequencies of tags in a manually annotated corpus. A temporary corpus on the other hand is a list consisting of the same words as in the goal corpus, tagged using most likely tag of (W_i, T_i). In the contextual learning module the goal corpus is a manually annotated running text while a temporary corpus consists of the same running text as in the goal corpus but with different tags.

4.4.1 Rules

A rule 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. 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.

To learn these transformations, the learner, in essence, tries out every possible transformation, and counts the number of tagging errors after each one is applied. After all possible transformations have been tried, the transformation that resulted in the greatest error reduction is chosen. Learning stops when no transformations can be found whose application reduces errors beyond some pre-specified threshold.

The general algorithm of learning transformation is as the following:

1. apply initial-state annotator to corpus
2. while transformations can still be found do
3. for from_tag = *tag1* to *tagn*
4. for to_tag = *tag1* to *tagn*
5. for corpus_position = 1 to corpus_size
6. if (correct_tag(corpus_position) --= to_tag
&& current_tag(corpus_position) == from_tag)
7. num_good_transformations(tag(corpus_position -1))++
8. else if (correct_tag(corpus_position) == from_tag
&& current_tag(corpus_position) == from_tag)
9. num_bad_transformations(tag(corpus_position-1))++
10. find *maxT* (num_good_transformations(T) - num_bad_transformations(T))
11. if this is the best-scoring rule found yet then store as best rule:
Change tag from from_tag to to_tag if previous tag is T
12. apply best rule to training corpus
13. append best rule to ordered list of transformations

In the algorithm described above, processing was done left to right. For each transformation application, all triggering environments are first found in the corpus, and then the transformation triggered by each triggering environment is carried out. The search is data-driven, so only a very small percentage of possible transformations really need be examined. In the above algorithm of

learning transformation, we give pseudocode for the learning algorithm in the case where there is only one transformation template:

Change the tag from X to Y if the previous tag is Z.

In each learning iteration, the entire training corpus is examined once for every pair of tags X and Y, finding the best transformation whose rewrite changes tag X to tag Y.

For every word in the corpus whose environment matches the triggering environment, if the word has tag X and X is the correct tag, then making this transformation will result in an additional tagging error, so we increment the number of errors caused when making the transformation given the part-of-speech tag of the previous word.

If X is the current tag and Y is the correct tag, then the transformation will result in one less error, so we increment the number of improvements caused when making the transformation given the part-of-speech tag of the previous word.

4.4.2 Learning Phase

Transformation-based learning has two major modules that are used for learning rules from the prepared tagged corpus. Those modules are: the lexical module and the contextual module. The overview of the two modules are discussed as follows

4.4.2.1 Learning lexical rules

The ideal goal of the lexical module is to find rules that can produce the most likely tag for any word in the given 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 learner module is weakly statistical. It uses the first half of the manually tagged corpus as well as any additional large unannotated corpus containing the manually tagged corpus with the tags removed. The learner module uses three different lexicons or lists constructed (by the user) from the manually tagged corpus and the large unannotated corpus. The Small word tag list, built from the manually annotated corpus, serves as the goal corpus in TEL and contains lines of the form.

Word Tag Frequency.

It simply contains the frequencies of each Word/Tag pair in the manually annotated corpus. Freq(W_i,T_i) denotes the Frequency of a word W_i with a specific tag T_i, and Freq(W) denotes the frequency of the total W_i in the manually annotated corpus. These numbers are used by the lexical learner in two ways: Freq(W,T) is used to compute the most likely tag T for the word W (i.e. the one with the highest frequency), and P(T|W) : = Freq(W_i,T_i) / Freq(W_i) is the estimated probability that the word W is tagged with the tag T.

For instance,

- Caaltun/NN uffata/NN kuta/JJ uffatti/VV ./PN | Chaltu wears miniskirt |
- Caalaan/NN muka/NN kuta/VV ./PN | Chala cuts the tree |

If the frequency of the word freq (kuta/VV) = 30 and freq (kuta/JJ) = 20, the total frequency of the word is freq (kuta) = 50. So, the probability of the two tags calculated as follows: P(VV|kuta) = 30/50 = 0.6 and P(JJ|kuta) = 20/50 = 0.4 . Based on the result the lexical module tag the word | kuta | with tag VV.

The other two lists are called bigwordlist and bigbigramlist are constructed from the large unannotated corpus. bigwordlist is a list of all words occurring in the unannotated corpus, sorted by decreasing frequency. The bigbigramlist is a list consisting of all word pairs (hence the name bigram) occurring in the unannotated corpus. Bigbigramlist does not contain the frequencies of word pairs; it only records if a given word pair occurs in the unannotated corpus or not.

Note that once the user has constructed these lists, the lexical learner does not need either the manually annotated or the unannotated corpus because the lists contain all the information the learner needs. The bigwordlist and the bigbigramlist are only used to check the trigger condition, which is completely determined by the data in these two lists.

First, the learner constructs a word list from smallwordtaglist, i.e. the words with tagging information removed. Then the initial state annotator assigns to every word the default most likely tag. The default tags is assigned nouns to all words in the corpus.

The word list thus obtained is the initial temporary corpus WL0; it was called TC0 in the general description of TEL above.

Once WL0 has been produced by the initial state annotator the learner generates the set of all permissable rules PR from all possible instantiations of all the predefined lexical templates and computes a score for every rule R in PR (see below). The rule which achieves the best score be-

comes rule number one in the output. Then the learner transforms WL0 to WL1 by applying this rule. This process is repeated until no rule can be found with a score greater than some threshold value, i.e. compute the new scores for all the rules in PR, pick the one with the best score, output this rule as rule number two and apply it on WL1 to get WL2, etc.

The scoring function is defined as follows: If the rule R has the template:

If Trigger then change tag X to tag Y and w is a word in WLi with current tag X satisfying the trigger condition, then R gets the score P(Y|w) - P(X|w) for the word w. The total score for R is then obtained by adding all the 'word scores'.

score(R) := S P(Y|w) - P(X|w)

w where the sum runs through all w in WLi with current tag X, satisfying the trigger condition.

If the rule R has the template:

if Trigger then change current tag to tag Y and w is a word in WLi satisfying the trigger condition, then R gets the score P(Y|w) - P(Current tag of w |w) for the word w.

The total score for R is then obtained by adding all the 'word scores'.

score(R) := S P(Y|w) - P(Current tag of w |w)

w where the sum runs through all w in WLi, satisfying the trigger condition. Note that the score the rule R gets for w always is of the form P(new tag|w)-P(old tag|w). A positive score means that the new tag is more likely than the old tag while a negative score means that the new tag is less likely than the old tag. The trigger condition is tested using bigwordlist and bigbigramlist, and the estimated probabilities are computed from the frequencies in smallwordtaglist.

The set of rules extracted during lexical modules are presented as follows.

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

The above rules are prepared for English original brill tagger. During Afaan Oromoo, some of the rules or the numbers of suffix are extend than English suffixes. For instance, rule number 3

and 4 are for the original Brill tagger which implies adding/deleting of prefix/suffix of only up to 4 characters was considered in English language. But in the case of Afaan Oromoo the number of suffix can be extend to 12 characters. We can see the above suffixes in the following word.

- **Qulqulleessitootaaf = Qulqull-eessitootaaf**

As we have seen the example Afaan Oromoo suffixes can be extended to 12 characters. By consideration of the number of suffixes the rule number 3 and 4 are changed to be up to 12 suffixes. The rules are changed to the following format.

- Change the most likely tag to Y if deleting /adding the suffix x, |x|<12, results in word, |x| is length of x.
- Change the most likely tag from X to Y if deleting/adding the prefix x, |x|<12, results in word, |x| is length of x.

In rule number 4 and 5 the rule changes a tag X to another tag Y while in the l-3 rules the new tag will be Y regardless of the current tag.

The lexical rules produced by the lexical learning module are then used to initially tag the unknown words in the contextual training corpus. Some of the lexical rules extracted by lexical modules from training corpus are presented in appendix A.

4.4.2.2 Learning contextual rules

Once the tagger has learned the most likely tag for each word found in the manually annotated training corpus and the method for predicting the most likely tag for unknown words, contextual rules are learned for disambiguation. The learner discovers rules on the basis of the particular environments (or the context) of word tokens.

The contextual learning process needs an initially annotated text. The input to the initial state annotator is an untagged corpus, a running text which is the second half of the manually annotated corpus where the tagging information of the words is removed. The initial state annotator needs a list, a so called training lexicon, which consists of list of words with a number of tags attached to each word. These tags are the tags that are found in the first half of the manually annotated corpus (the ones used by the lexical module). The first tag is the most likely tag for the word in question and the rest of the tags are in no particular order.

Word Tag1 Tag2 … Tagn

With the help of the training lexicon, the bigbigramlist (the same as used in the lexical learning module, see above) and the lexical rules, the initial state annotator assigns to every word being in the untagged corpus the most likely tag. It tags the known words, i.e. the words occurring in training lexicon, with the most frequent tag for the word in question. The tags for the unknown words are computed using the lexical rules: each unknown word is first tagged with NN and then each of the lexical rules are applied in order. The reason why bigbigramlist is necessary as input is that some of the triggers in the lexical rules are defined in terms of this list. The annotated text thus obtained is the initial temporary running text RT07 which serves as input to the contextual learner.

Transformation-based error-driven learning is used to learn contextual rules in a similar way as in the lexical learner module. The input to the contextual learner is the second half of the manually annotated corpus (i.e. the goal corpus), the initial temporary corpus RT0 and the training lexicon (the same one as above). First, the learner generates the set of all permissible rules PR from all possible instantiations of all the predefined contextual templates. Note that PR in the contextual module and PR in the lexical module are different sets of rules, since the two modules have different transformation templates. Here, the triggers of the templates for the rules usually depend on the current context. The following are some of the triggers of the contextual transformation templates:

1. The preceding/following word is tagged with Z.
2. One of the two preceding/following words is tagged with Z.
3. One of the three preceding/following words is tagged with Z.
4. The preceding word is tagged with Z and the following word is tagged with V.
5. The preceding/following two words are tagged with Z and V.
6. The current word is V and the following word is Z.
7. The current word is V and the preceding/following word is tagged with Z.
8. The current word is V and the word two words before/after is tagged with Z.

Sample of contextual rules extracted are presented in Appendix B.

For all rules in PR for which the trigger condition is met the scores on the temporary corpus RT0 are computed. It picks the rule with the highest score, R1, which is then put on the output list.

Then the learner applies R1 to RT0 and produces RT1, on which the learning continues. The process is reiterated putting one rule (the one with the highest score in each iteration) on the

output list8 in each step until learning is completed, i.e. no rule achieves a score higher than some predetermined threshold value. A higher threshold speeds up the learning process but reduces the accuracy because it may eliminate effective low frequency rules.

If R is a rule in PR, the score for R on RTi is computed as follows. For each word in RTi the learner computes the score for R on this word. Then the scores for all words in RTi where the rule is applicable are added and the result is the total score for R. The score is easy to compute since the system can compare the tags of words in RTi with the correct tags in the goal corpus (the texts are the same). If R is applied to the word w, thereby correcting an error, then the score for w is +1. If instead an error is introduced the score for w is -1. In all other cases the score for w is 0.

Thus, the total score for R is score(R): = number of errors corrected - number of errors introduced.

There is one difference compared to the lexical learning module, namely the application of the rules is restricted in the following way: if the current word occurs in the training lexicon but the new tag given by the rule is not one of the tags associated to the word in the training lexicon, then the rule does not change the tag of this word.

The rules produced by the contextual learning module together with the lexical rules and several other files can then be used as input to the tagger to tag unannotated text, as will be described in the next section.

4.4.3 Flow Chart of Afaan Oromoo Brill's Tagger

During Transformation based approach the rules or grammar is induced directly from the training corpus. These rules are induced using two major modules of Transformation-based error-driven learning. Those modules are the lexical and contextual rules.

The lexical module uses transformation-based error driven learning to produce lexical tagging rules and the contextual module, also using transformation-based error-driven learning, produces contextual tagging rules. So, to induce the rules from Afaan Oromoo training corpus we used the above two modules that means lexical and contextual rules.

When using these rules to tag an unannotated text, at the first Afaan Oromoo untagged text is given to the preprocessing components. The pre-processing of the untagged text is split the text into sentences, tokenizing the sentences into words and removal of punctuation marks. After the

preprocessing the processed text is given to Initial State Annotator. The initial state annotator applies to the unknown words (i.e. words not being in the lexicon), then applies the ordered lexical rules to these words. The known words are then tagged with the most likely tag and finally the ordered contextual rules are applied to all words. The Afaan Oromoo Brill's taggers is prepared based on the nature of the language by some modification on the work of (Abraham, 2013)(Megyesi, 1998).

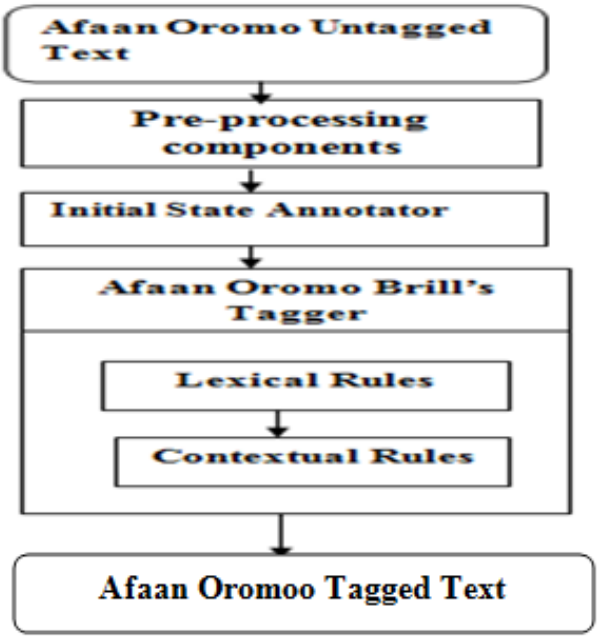

Figure 4.3 Flow chart diagram of Afaan Oromoo Brill's tagger (Abraham, 2013)

4.5 Flow Chart of Afaan Oromoo Hybrid Tagger

Hybrid models are basically combination of different approaches and form as one model to tag/annotate unannotated text(Juafisky & Martin, 2006)(Kanak & et.al, 2014) (Prajadhip et.al, 2015).

The proposed hybrid models for Afaan Oromoo are designed by combination of rules based and statistical hidden markov models. When the hidden markov models used as initial tagger to tag unannotated text, the rule based tagger are used for error detection and correction. To annotate the unannoted text the designed tagger follows the following procedure. At first, unannotated

text is given to the tokenizer module. The tokenizer segregates words, punctuation marks and symbols of an input text, and subsequently assigns them into tokens by creating whitespaces between them. Then, the processed text if given to the HMM tagger to tag the raw text as initial tagger. The HMM tagger tags the raw text based on the viterbi algorithm of best tag sequence trained from the matrix of transition probability and emission probability of the training corpus. The hybrid tagger of Afaan Oromoo was designed based on the work of (Teklay, 2010) (Prajadhip et.al, 2015) with some modification. The flow chart diagram of hybrid tagger is presented as follow with detail descriptions.

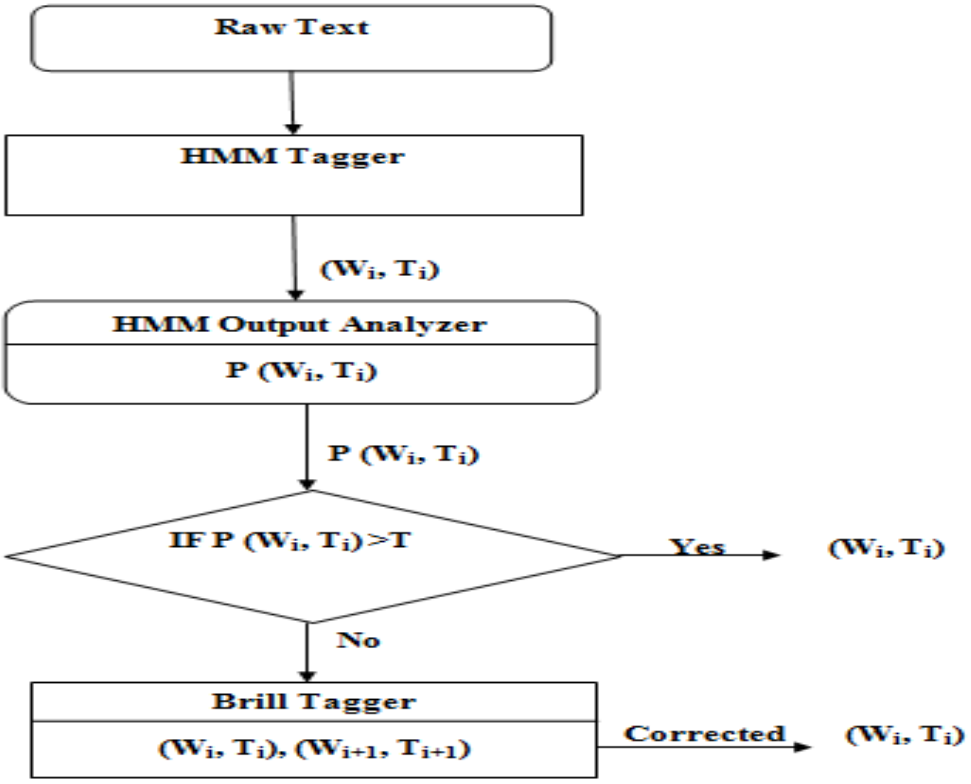

Figure 4.4 Flow chart diagram of Afaan Oromoo hybrid tagger (Teklay, 2010)

The raw/input texts are given to the tagger. From the raw text a word sequence Wi is given to the HMM tagger as an input, the HMM tagger assigns the tag sequences Ti using the Viterbi Algorithm which implies that the output of the HMM tagger is the word tag sequence (Wi,Ti). This word tag sequence is given to the output analyzer that checks whether the predetermined threshold value for a word Wi is attained or not. The threshold value is a value used for checking the

confidence level of tagging a given sequence of words. And hence, the output analyzer decides based on the threshold value. Accordingly if the threshold value of a word tag pair is below the confidence level, a fixed window size, in this case a window size of two which implies bigram of words is given to the rule based tagger for correction and the rule based tagger produce the corrected tagged words. Otherwise the output of the HMM tagger is the final output of the word to be tagged. This process is repeated until the HMM tagger tagged words are checked by threshold value and corrected by rule based tagger

4.6 Implementation of Hybrid Tagger

The proposed hybrid tagger is designed by combination of HMM and Brill's tagger. During the implementation of this hybrid tagger HMM used as initial state annotator and the tagged/output of the HMM taggers are given to Brill taggers for error detections and correction.

During the implementation of the hybrid tagger the HMM tagger first annotates the given raw text and provides confidence level of the tag sequence. If the confidence level (threshold value) is not achieved by this tagger, it is corrected by the rule based tagger. Optimal threshold value is set depending on the outcome of the performance of the hybrid tagger under experiment. During the tagging process, the HMM tagger uses the viterbi algorithm to find the probabilities of the tag/word pair and the probability of the optimal path. Hence, it is possible to compare the probabilities of each tag for each word with the fixed threshold value. When the probability of the assigned tag for the given word is greater than the fixed threshold it suffice that the assigned tag does not need correction. Otherwise the word is given to the rule based tagger to correct its tag. When the rule based is to correct the tag of the word, it applies the set of transformation rules that it has learned during the learning phase.

The HMM tagger labels all the word sequences using the information obtained from the HMM model. Afterwards, a sliding window size of two is fixed and checked whether the probability of the assigned tag of the first word in the window is less than the threshold value or not. If the corresponding probability is greater than the fixed threshold value, the assigned tag does not need any correction. Otherwise, the tokens in the window are given to the rule based tagger for correction. The rule based tagger tags the tokens using rules and it gives the first word and its corresponding corrected tag as an output. Then the window is shifted to incorporate one additional

word by removing the first correctly tagged word. This step is repeated until the HMM tagged words are finished.

Generally, the complete algorithm of the hybrid tagger is given as follows:

1. Get HMM model tagger

2. Get the transformation rules from the rule based tagger learning phase.

3. Read raw text.

4. Tag the raw text using HMM tagger.

5. Get the probability, Prob, of each tag given the word while HMM tagger is assigning the tag for the word ($P(T_i|W_i)$)

6. Fix experimental threshold value, θ

7. While there are HMM tagged words compare the probability of each tag for each word against: θ

8. If $P(T_i|W_i) <$ then θ

    8.1. Apply rules to W_i and W_{i+1}

    8.2. Get Wi|Ti as an output assumed to be the corrected one.

9. Else the selected tag suffices and considered to be correct and taken as an output.

10. End of the HMM tagged words

The proposed hybrid tagger follows the above algorithm to tag the raw corpus/text. This algorithm applied on previously individual trained taggers of HMM and Brill's taggers.

CHAPTER FIVE

EXPERIMENTS AND PERFORMANCE ANALYSIS

5.1 Introduction

This chapter presents the details experimental analysis of the designed model. The model was developed and tested in NLTK using a python. There is an infrastructure about natural language processing, which is called Natural Language Toolkit (NLTK) implemented in Python programming language(Steven, 2015) (Steven et al, 2009). How the corpus was collected and the preprocessing of the collected corpus discussed in details. To see their performance experimental analyses have been conducted on each separated approaches: HMM tagger, rule based tagger and hybrid tagger.

5.2 Natural Language Toolkit and Python

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces and it has lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries (Steven et al, 2009). It is an open source Python library for Natural Language Processing. NLTK is implemented as a large collection of minimally interdependent modules, organized into a shallow hierarchy. A set of core modules defines basic data types that are used throughout the toolkit. The remaining modules are *task modules*, each devoted to an individual natural language processing task. For example, the nltk.parser module encompasses to the task of *parsing*, or deriving the syntactic structure of a sentence; and the nltk.tokenizer module is devoted to the task of *tokenizing*, or dividing a text into its constituent parts.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.” So, for our work we used NLTK 3.0.2 that compatible with our windows. Python is a simple yet powerful programming language with excellent functionality for processing linguistic data.

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. Now a day there are different version of free python is available on the internet. But for our work python 3.4.3 is selected due to Unicode character, not ASCII is tolerated.

5.3 Corpus Preparation

As Nadja Nesselhauf(2011) explains, a corpus can be defined as a systematic collection of naturally occurring texts, may be written or spoken data. During Annotated corpora, some kind of linguistic analysis has already been performed on the texts, such as sentence analysis, or, more commonly, word class classification.

A corpus is considered representative if what we find on the basis of the corpus also holds for the language it is supposed to represent. Representativeness is typically achieved by balancing, i.e. covering a wide range of frequent and important text categories that are proportionally sampled from the target population (Megyesi, 1998).

We used the aforementioned annotated corpora/Text as input to train our model, in case of our work we used tagged text as training data and testing data for Afaan Oromoo Text tagger(Jurafisky & Martin, 2006).

As Megyesi(1998) discussed on the above the annotated corpus is expected to represent/covering a wide range of frequent and important text/domain categories. Those domains categories can be text of news, books, fiction, editorial, scientific, periods and etc. A corpus that is prepared from the aforementioned categories is considered as a balanced corpus. To prepare these balanced corpus it needs time, money and language experts who manually tag the corpus to their word classes. Due to these constraints we used some of the domain categories such as Holly Books, news, and fiction even if the number of our corpus is very small compared to brown corpus in NLTK corpora.

A raw corpus/text was collected from Afaan Oromoo news agencies and Medias: Oromia Radio and Television, Nekemte FM radio agencies, VOA (America voice of Afaan Oromoo), holly Bible and Afaan Oromoo drama scripting languages which is called Dhebu. From collected corpus, 700 sentences was tagged and added to 817 tagged sentences taken from the work of (Abra-

ham, 2013). Since Abraham's data prepared using 26 tagset, a few corrections was made using identified tagset. Totally, 1517 tagged sentences were used for thesis to train and test the model. Even though our corpus is very small compared to brown corpus in NLTK, we used 85% tagged sentence for training and the remaining 15% was used for testing. Sample of training data used for training is presented in appendix C.

5.4 Preprocessing Components

A processed text/corpus is input for our model and for his high accuracy. Since this processed text/corpus is input for the model, we have to implement preprocessing components on our corpus.

Since Afaan Oromoo is one of the languages that uses Latin script, the text/corpus easily readable by the machine, no need to implement language transliteration. Our preprocessing components have three major components; sentence splitter, tokenizer and tagset analyzer. At first, the content of corpus is given to sentence splitter module to split the text into sentence by using the Afaan Oromoo sentence end markers such as '**!**' '**?**' ',' and '**.**'. For our language processing, we want to break up the string into words and punctuation. The splitted sentence is given to tokenizer module to break up the string into words and punctuation. During training phase tagged words or punctuations were tagged in the form words/tags. The tagset analyzer extracts the tags from the output of the tokenizer. This extracted tagset is used for HMM tagger to tag a new text (Steven et al, 2009).

5.5 Performance Analysis of HMM Tagger

To ground this discussion, An HMM is one of a common NLP application used for part-of-speech (POS) tagging. An HMM is desirable for this task as the highest probability tag sequence can be calculated for a given sequence of word forms. This differs from other tagging techniques which often tag each word individually, seeking to optimize each individual tagging greedily without regard to the optimal combination of tags for a larger unit, such as a sentence. The HMM does this with the Viterbi algorithm, which efficiently computes the optimal path through the graph given the sequence of words forms. For this performance analysis the NLTK HMM tagger tool is used with some modifications. To see the performance analysis of HMM tagger, experi-

mental analysis is conducted on different portions of training set. For this performance analysis the researcher have used incremental approaches started from 10% training set and its accuracy were measured on separated testing set. Having got a low performance of the tagger trained on the 10% of the training set, the researchers‟ kept on adding the training data by 10% until they got a desired performance of the tagger. Finally, the HMM tagger scored 91.9 on 100% training set.

The different experiments conducted using different portions of the training set with the corresponding performance of the taggers was presented in figure 5.1 as follows.

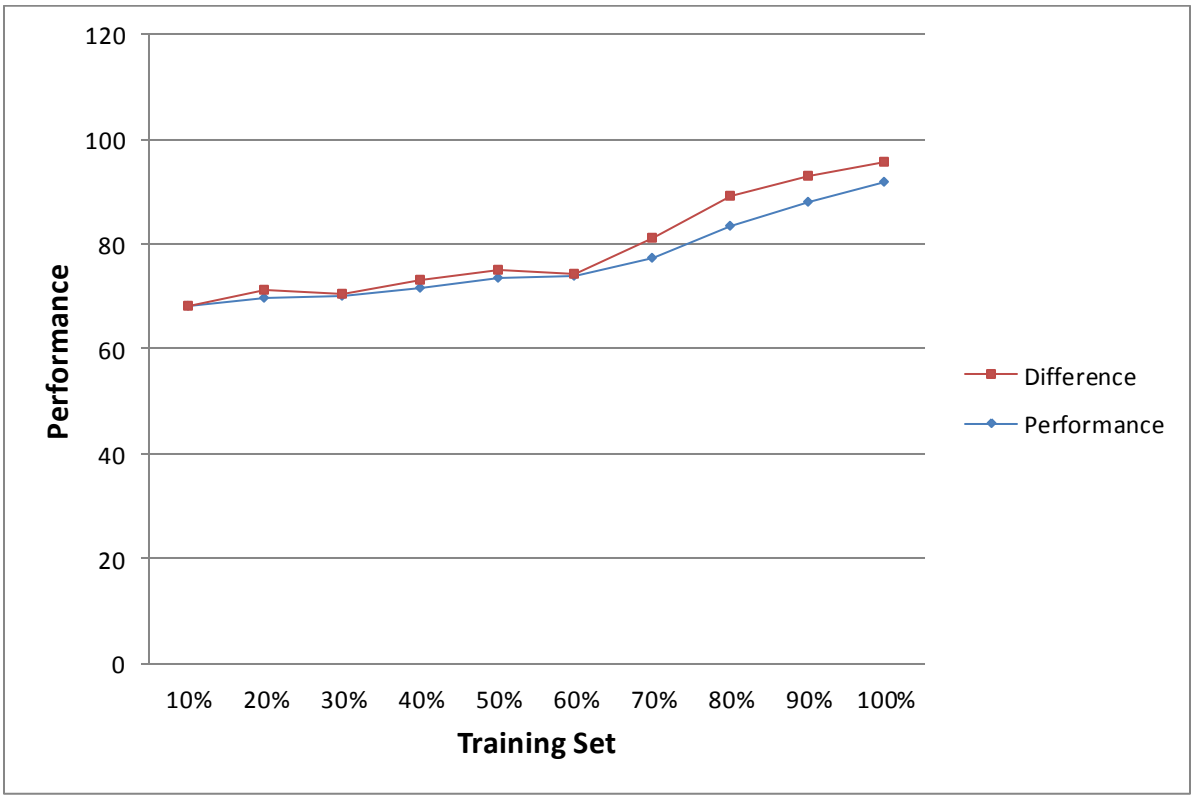

Figure 5.1 Performance curve analysis of HMM tagger

Generally, the curve shows that as training set increases the accuracy of the tagger also increases simultaneously.

5.6 Performance Analysis of Rule Based Tagger

For this work Brill's transformational rule-based tagger exists in NLTK modules are used with some modification based on the nature of language. Those modifications are mostly done on the suffix of the language. This modified Brill's also evaluated using different portion of training set to check its performance as the same fashion on the above HMM tagger. During experimental analysis, the different initial state annotators of Brill's are namely: default, unigram, bigram and trigram tagger evaluated independently to choose which scored high accuracy as initial state annotators for Brill's. The ten different experiments conducted on the rule based tagger using different portions of the training set and different initial state annotators was presented using figure 5.2 as follows.

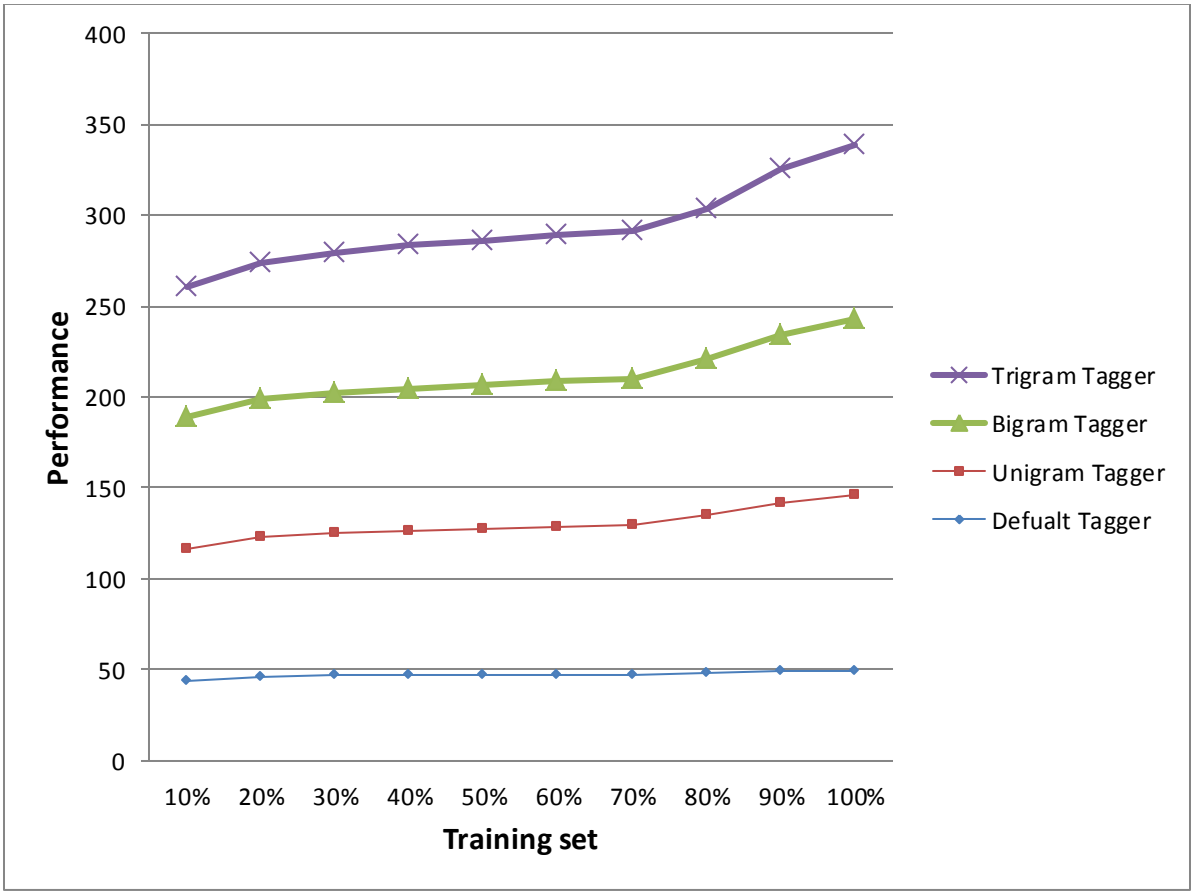

Figure 5.2 Performance curve analysis of rule based tagger

As we have seen on the above figure 5.2 the Brill's tagger used default tagger as initial state annotators. The initial state annotator assigns a specified default tag for unknown words, which

does not exist in the training corpus. The default tag is selected by the most frequent tag in the training corpus (Beni, 2008). In case of our training corpus the Noun (NN) part of speech tag selected as default tag for unknown words based on most frequently appearing in the training corpus.

Also they are different types of tagger used as the backoff tagger such as unigram tagger, bigram tagger and trigram tagger. The backoff taggers are implemented in the Brill's tagger to detect and tag the word which is not tagged by the preceding tagger (Steven and et al., 2009). When unigram tagger assigns the most likely tag for a given word, the bigram and trigram taggers assign tags to words based on the preceding one and two words with their corresponding tags respectively (Jurafsky & Martin, 2006).

In this study, among the above four taggers: default, unigram, bigram and trigram taggers; the higher performance of the rule based tagger is obtained when the initial state annotator is unigram tagger which shows that the training corpus used does not have many bigram and trigram instances that can be found in the training data. Since unigram can't detect words contextually, we used bigram tagger for brill's tagger and unigram tagger as a backoff tagger.

During the experiment the rule based tagger has also performed the worst when it uses the default tagger. The reason behind this worst performance is, the rule based tagger faces problems in learning rules as the corpus initially will be annotated uniformly using one type of tag, the Noun (NN) tag(Abraham, 2013)(Teklay, 2010).

5.7 Performance Analysis of Hybrid Tagger on Different Threshold Values

The designed hybrid approaches are combined of the HMM and rule based tagger. This design is implemented by undergoing several distinct steps. In this hybrid approaches, the HMM tagger was used as Initial state annotators. The HMM tagger first annotates the word sequences and if the desired threshold value is not attained, the words are given to the rule based tagger for correction. In this thesis work based on performance analysis, threshold valued fixed to 0.5 since taking threshold value less than this value does not bring significant difference on the performance of the tagger. Since there is no probability less than Zero and above one, the HMM tagger tags the words with the probability of between 0-1. Due to the shortage of training data the HMM tagger tags the words mostly by probability of less than 0.5. By considerations of this

accuracy we considered that if the probability of the tags greater than and equal to 0.5 more accurate and less than 0.5 is less accurate. During the implementation of hybrid approach, those less accurate which mean less than 0.5 are given to Brill's taggers for correction. when the threshold value is 1 the performance goes down as most of the time there is no probability greater or equal to 1, the rule based tagger tags all the words and it didn't improved the performance of the tagger. The highest accuracy is achieved by threshold value of 0.5.

The performances of the hybrid tagger on different threshold values are presented in figure 5.3.

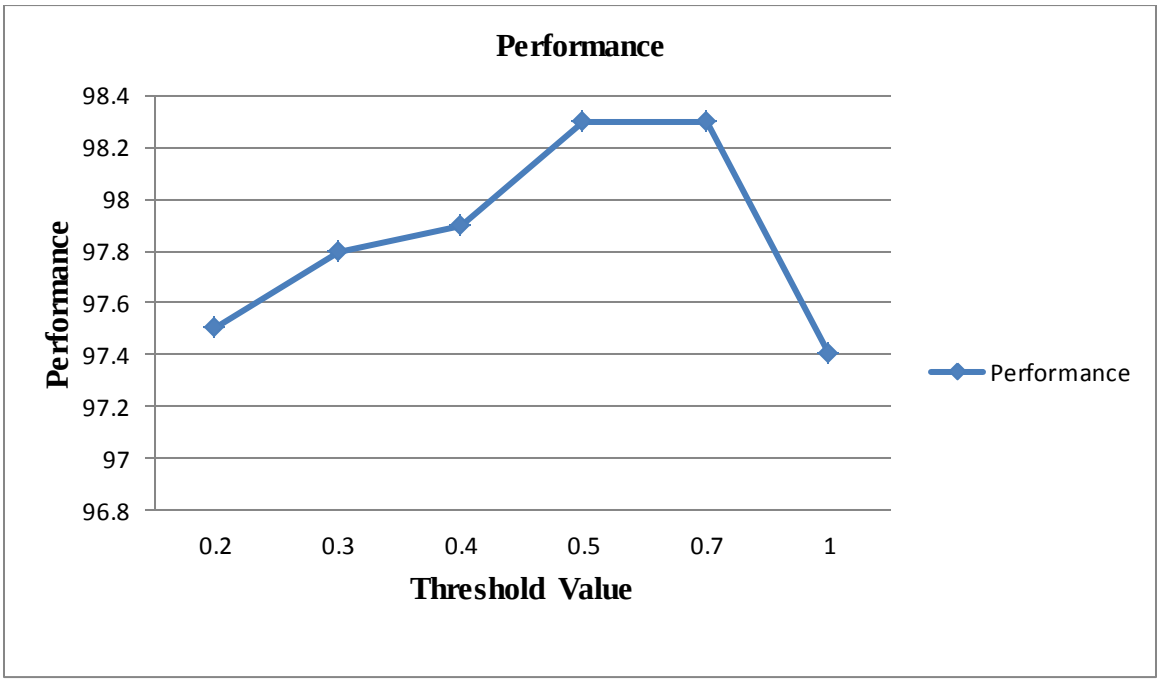

Figure 5.3 Performance curve analysis of hybrid tagger

On the above learning curve analysis of figure 5.3, the figure clearly shows improved performance 98.3 was achieved at 0.5 threshold value. Based on the achieved accuracy 0.5 used as the fixed threshold value to correct the words.

5.8 Experimental Analysis

Here the experimental analysis of the three taggers which is developed in this thesis work is analyzed using the frequency of the tagset. In order to analyze the performance of the three Afaan Oromoo taggers, the frequency of the tagset in the entire corpus, training set and testing set is considered. In this work, 35 identified tagset was used to train the model. To simplify the analysis we categorized the tagset into basic and derived tagset. The frequency of Afan Oromo basic tagset which is totally 11 tagset analysis independently, the derived tagset are categorized as others tagset (NPROP, NC, NP, PS, PC, PREF, PD, PDPR, AX, VC, JC, JP, ADVPREP, ADVC, ON) and the frequency of tagset in the entire corpus, training set and test set is presented using table 5.1 as follows:

Tag name	Entire Corpus Frequency	Training Frequency %	Testing Frequencies %
Noun	20.38%	20.07%	26.4%
Verb	19.75%	19.98%	22.2%
Adjective	13.35%	12.96%	10.5%
Preposition	9.80%	10.45%	9.8%
Punctuation Marks	9.52%	9.45%	10.3%
Pronoun	6.46%	6.61%	9.5%
Adverb	4.97%	5.43%	3.7%
Conjunction	1.70%	1.92%	1.4%
Numerals	0.64%	0.59%	0.5%
Interjection	0.28%	0.33%	0.2%
Negation	0.05%	0.01%	0.03%

Others	13.1%	12.2%	5.47%
Totals	100%	100%	100%

Table 5.1 Frequency of tagset in training, testing and entire corpus

As we have seen in table 5.1 of frequency of the tags, tag Noun is frequently occurring POS tag than other POS tags in the entire corpus. In this thesis, taking their frequencies into consideration Noun is used as the default tag for Afaan Oromoo Brill's taggers to tag unknown words.

5.8.1 Experimental Analysis of HMM Tagger

Experimental analysis of HMM tagger presented using a table 5.2 as follows:

		Test tags (Predicted tags)												
		NN	VV	PN	JJ	PR	PP	AD	CC	JN	II	NG	Others	Total
Reference tags (Desired tags)	NN	**23.3%**	0.8%	0.1%	0.4%	0.3%	0.3%	-	-	-	0.01%	-	-	25.21%
	VV	0.45%	**20.8%**	-	0.1%	0.2%	-	-	0.1%	-	0.02%	-	-	21.67%
	PN	-	-	**10.16%**	-	-	-	-	-	-	-	-	-	10.16%
	JJ	0.25%	0.2%	-	**9.6%**	-	-	-	-	-	-	-	-	10.05%
	PR	-	0.1%	-	-	**8.9%**	-	-	-	-	-	-	-	9%
	PP	0.2%	0.1%	0.04%	-	-	**7.6%**	-	-	-	-	-	-	7.94%
	AD	-	-		0.1%	-	-	**3.7%**	-	0.1%	-	-	-	3.9%
	CC	-	-	-	-	-	0.6%	-	**1.2%**	-	-	-	-	1.8%
	JN	-	-	-	-	-	-	-	-	**0.3%**	-	-	-	0.3%
	II	-	-	-	-	-	-	-	-	-	**0.17%**	-	-	0.17%
	NG	-	-	-	-	-		-	-	-	-	**0.03%**	-	0.03%
	Others	2.2%	0.2%	-	0.3%	0.1%	1.3%	-	0.1%	0.1%	-	-	**14.24%**	9.77%
	Total	26.4%	22.2%	10.3%	10.5%	9.5%	9.8%	3.7%	1.4%	0.5%	0.2%	0.03%	5.47%	**100%**

Table 5.2 Experimental analysis of HMM tagger

The above experimental analysis of HMM indicate that from 26.4% NN in test set 23.3% were correctly classified as NN, 0.45% incorrectly classified as a VV (verb), 0.25 incorrectly classified as JJ (Adjective), 0.2% incorrectly classified as PP (preposition) and the remains 2.2% incorrectly classified by Others tagset which includes (NPROP, NC, NP, PS, PC, PREF, PD, PDPR, AX, VC, JC, JP, ADVPREP, ADVC, ON). Those cells whose values 0 is indicated using hyphen (-) and the value of correctly classified tags were presented in the diagonal of the tables. So, totally were 23.3% nouns correctly classified, the remains of 3.1% incorrectly classified.

Since this tagger strives to find the most probable path for a given sequence of words, there is no defined pattern of confusion like that of the rule based tagger rather the confusion is distributed almost across all part of speech tags. This misplaced is due to the fact that there is no standard and large corpus for training the tagger. As a result, the HMM tagger scores less than the rule based tagger.

5.8.2 Experimental Analysis of Brill's Tagger

Experimental analysis of Brill's tagger presented as follows:

		Test tags (Predicted tags)												
Reference tags (Desired tags)		NN	VV	PN	JJ	PR	PP	AD	CC	JN	II	NG	Others	Total
	NN	**24.3%**	-	-	0.4%	0.1%	-	0.1%	-	-	-	-	-	24.9%
	VV	1%	**22%**	-	0.3%	0.1%	-	0.2%	0.1%	-	-	-	-	23.7%
	PN	-	-	**10.3%**	-	-	-	-	-	-	-	-	-	10.3%
	JJ	0.1%	-	-	**9.8%**	-	-	-	-	-	-	-	-	9.9%
	PR	-	-	-	-	**9.2%**	-	-	-	-	-	-	-	9.2%
	PP	-	-	-	-	-	**9.4%**	-	-	-	-	-	-	9.4%
	AD	-	-	-	-	-	-	**3.4%**	-	-	-	-	-	3.4%
	CC	-	-	-	-	-	0.3%	-	**1.1%**	-	-	-	-	1.4%
	JN	-	-	-	-	-	-	-	-	**0.4%**	-	-	-	0.4%
	II	-	-	-	-	-	-	-	-	-	**0.2%**	-	-	0.2%
	NG	-	-	-	-	-	-	-	-	-	-	**0.03%**	-	0.05%
	Others	1%	0.2%	-	-	0.1%	0.1%	-	0.2%	0.1%	-	-	**9.87%**	7.17%
	Total	26.4%	22.2%	10.3%	10.5%	9.5%	9.8%	3.7%	1.4%	0.5%	0.2%	0.03%	*5.47%*	100%

Table 5.3 Experimental analysis of Brill's tagger

The above Brill's experimental analysis show that out of 26.4% NN in test set 24.3% were correctly classified as NN, 1% incorrectly classified as VV (verb), 0.1 incorrectly classified as JJ (adverb) and the remains 1% incorrectly classified by Others tagset which includes (NPROP, NC, NP, PS, PC, PREF, PD, PDPR, AX, VC, JC, JP, ADVPREP, ADVC, ON). Totally, were 24.3% nouns correctly classified, the remains of 2.1% are incorrectly classified. This result shows that the Brill's tagger less confused compared to HMM tagger. The reason behind Brill's tagger achieved high accuracy is some of regular expression was used in the Brill's taggers and the number of training data could affect the tagger less than its effect on the HMM tagger.

5.8.3 Experimental Analysis of Hybrid Tagger

The experimental analysis of the hybrid Afaan Oromoo part of speech tagger is also done on the basic tagset and test set with Brill's and HMM tagger. During the hybrid tagger we have got good performance compared to Brill's and HMM tagger. This experimental analysis of the hybrid Afaan Oromoo part of speech tagger is presented in table 5.4 as follows:

		Test tags (Predicted tags)												
Reference tags (Desired tags)		NN	VV	PN	JJ	PR	PP	AD	CC	JN	II	NG	Other	Total
	NN	**25.2%**	0.1	-	0.2%	0.1%	0.2%	-	-	-	-	-	-	25.8%
	VV	0.3%	**22.1%**	-	0.1%	-	-	-	0.1%	-	-	-	-	22.6%
	PN	-	-	**10.3%**	-	-	-	-	-	-	-	-	-	10.3%
	JJ	0.1%	-	-	**10.2%**	-	-	-	-	-	-	-	-	10.3%
	PR	-	-	-	-	**9.4%**	-	-	-	-	-	-	-	9.4%
	PP	-	-	-	-		**9.6%**	-	-	-	-	-	-	9.6%
	AD	-	-	-	-	-	-	**3.7%**	-	-	-	-	-	3.7%
	CC	-	-	-	-	-	-	-	**1.3%**	-	-	-	-	1.3%
	JN	-	-	-	-	-	-	-	-	**0.5%**	-	-	-	0.5%
	II	-	-	-	-	-	-	-	-	-	**0.2%**	-	-	0.2%
	NG	-	-	-	-	-	-	-	-	-	-	**0.03%**	-	0.03%
	Others	0.8%	-	-	-	-	-	-	-	-	-	-	**7.47%**	6.27%
	Total	26.4%	22.2%	10.3%	10.5%	9.5%	9.8%	3.7%	1.4%	0.5%	0.2%	0.03%	*5.47%*	**100%**

Table 5.4 Experimental analysis of hybrid tagger

The hybrid experimental analysis indicate that out of 26.4% NN in test set were 25.2% correctly classified as NN, 0.3% incorrectly classified as VV(verb), 0.1% incorrectly classified as JJ(Adjective) and the remains 0.8% incorrectly classified by Others tagset which includes (NPROP, NC, NP, PS, PC, PREF, PD, PDPR, AX, VC, JC, JP, ADVPREP, ADVC, ON). Totally, were 25.2% nouns correctly classified, the remains of 1.2% are incorrectly classified. The result shows that misplaced of tags by the hybrid tagger is less than the misplaced made by the HMM tagger and rule based tagger, which clearly shows improved performance was achieved on combinations rather than separated taggers.

The errors of assigning tags to other undesired tags is attributed to different factors such as lack of standard corpus and incorrect labeling of the words in the prepared corpus.

Generally, the Accuracy obtained in the hybrid tagger shows that the hybrid tagger is more preferable than separated taggers.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Part of Speech (POS) tagging is the essential basis for other Natural Language Processing (NLP) applications. It is a process of assigning corresponding POS tag for a word that describes how the word is used in a sentence. It is the process of labeling words with their corresponding part of speech categories. In NLP there are different research tasks such as grammar checker, spelling correction and parsing, machine translation, and automatic summarization. .

Now a day, there are different approaches to develop part of speech tagger. Among the most common approaches stated by different authors are: rule based, HMM, artificial neural network, memory based and combination/hybrid of individual approaches.

In this thesis, the development of part of speech tagger using hybrid approach that combines rule based and HMM approaches was conducted for Afaan Oromoo. To develop this hybrid model the HMM and Brill's tagger were developed and evaluated independently. The transformational based learner/rule based taggers tag the words based on rules, or transformations induced directly from the training corpus without human intervention or expert knowledge. The HMM taggers tag the words based on the most probable path for a given sequence of words.

The hybrid approach used HMM tagger as initial state annotators and Brill's tagger as a corrector. The total corpus is divided into training set and testing set to train and test the tagger. Even though our data corpus is very small compared to brown corpus in the NLTK, we used 85% of our data which means 1289 sentences for training and the remains 15% that consists of 228 sentences were used for testing purpose.

NLTK 3.0.2 which is compatible for our windows and Python 3.4.3 compatible with the language are used in the implementation and experiment of Afaan Oromoo tagger. For performance analysis the three taggers namely: HMM, rule based and hybrid taggers are evaluated with the same training and testing set; they achieved accuracy of 91.9%, 96.4% and 98.3%, respectively. To see the performance of the hybrid tagger the investigations was done on different threshold values and 0.5 scored the better performance and it was used as a fixed threshold value of the hybrid tagger. When we compare the accuracy of the taggers, the hybrid tagger was increased by 1.9% than the rule based and 6.4%

than the HMM tagger which is developed in this thesis. Based on their performance and learning curve analysis, the study has concluded that the hybrid tagger has been benefited from the advantages of the two separated approaches and achieved an improved performance.

6.2 Recommendation

Natural language processing is now active research areas on grammar checker, spelling correction, parsing and part of speech taggers for all languages particularly for Afaan Oromoo. To develop Part of speech tagger it needs standard data corpus and language experts. Rule based uses rules extracted from the training corpus and also some rules are given to the machine using regular expression. The language expert easily identifies those rules and gives to the machine. In addition to standard data corpus and language experts, we forwarded the following points for future works:

- ✓ Improving this work using wide coverage of domain area which is not included in this study and increasing the size training corpus those collected from different domain categories: News, Scientific journals and periodicals.
- ✓ Since Afaan Oromoo is morphological rich: nouns, adjectives and verbs inflected for cases, number and a gender, morphological segmentations is needed to segment words to their morphemes.
- ✓ Extending this work by using tagset that can identify gender and tense with different feature sets has significance impact for applicability of the tagger.
- ✓ Comparison with other hybrid approaches like: the hybrid of rule based and ANN and combination of the taggers using voting on those has the ability of tagging unknown words like support vector machine, memory based tagging and conditional random fields.

References

Abara Nefa, (1988). Long Vowels In Afaan Oromoo: A Generative Approach. In M.A. Thesis. School of Graduate Studies, Addis Ababa University.

Abdulsamad M, (1997). 'Seerlugaa Afaan Oromoo'. Bole Printing Enterprise, Addis Ababa , Ethiopia.

Abraham Gizaw, (2013). Improving brill's Tagger Lexical and Transformation rule For Afaan Oromo Language. In M. Sc. Thesis. School of Graduate Studies, Addis Ababa Universi ty, Ethiopia.

Abraham Tesso, (2014). Automatic Part-of-speech Tagging for Oromo Language Using Maxi mum Entropy Markov Model (MEMM) , School of Computer Science and Tech nolo gy, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China.

Adugna Barkessa, (2014). Bu'uura Barnoota Afaaniifi Afoola Oromoo,Semmoo. Oromiya Finfinnee, Ethiopia. Design and Printed By Far East Trading PLC.

Baum, L,(2000). "An Inequality and Associated Maximization Technique in Statistical Esti mation on Probabilistic Functions of a Markov Process",*IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Volume: 22, Issue: 4*, pp. 371-377.

Baye Yimam, (1981). Oromoo Substantives Some Aspects Of Their Morphology and Syntax, School Of Graduate Studies, Addis Ababa University.

Beni Ruef, (2008). Transformation-Based Learning and Part-of Speech Tagging of Old English

Berna Arslan & Özlem Patan, (2009). Developing Methods For Part Of Speech Tagging In Turkish ,Department of Computer Engineering ,Boğaziçi University .

Bichaka Fayissa, (2004). The Journal of Oromo Studies, Middle Tennessee State University, USA Volume 11, Numbers 1 And 2.

Bird, S, (2006). NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on interactive Presentation Sessions. Association for Computational Linguistics, Mor ristown, NJ. Volume 1.

Brill, E. and Wu, J, (1998). Classifier combination for improved lexical disambiguation. In COLING/ACL-98, Montreal, Canada, pp. 191–195.

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

Brill, E, (1992). A simple rule-based part of speech tagger. Proceedings of the Third Annual Conference on Applied Natural Language Processing, ACL.

Census Data, (2007). Summary and Statistical Report of the Ethiopian population and Housing Census Results.

Christophe Hurlin,(2013). Maximum Likelihood Estimation Advanced Econometrics - HEC Lausanne, University of OrlÈans

Cutler, A.(1986). Forbear is a home: Lexical prosody does not constrain lexical access. Lan guage and Speech, 29,201-219

Cutting, Douglas R., Jan 0. Pedersen, David Karger, and John W. Tukey.(1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *SIGZR* .92**,** pp. 318-329.

Edward Loper NLTK Tutorial Tagging.

Fahim Muhammad Hasan, NaushadUzZaman, and Mumit Khan, (2006). Comparison of Differ ent POS Tagging Techniques (n-grams, HMM and Brill's Tagger) for Bangla, International Conference on Systems, Computing Sciences and Software Engineering (SCS206) of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 06), pp: 4-14.

Gamta Tilahun, (1989). Oromo-English Dictionary. Addis Ababa University Printing Press, Ad dis Ababa.

Gamta Tilahun, (2001). Forms of Subject and Object in Afaan Oromo.Journal of Oromo Vol- ume 8 Number 1&2.

George Yule, (2006). The Study Of Language Third Edition, Cambridge university press The Edinburgh Building, Cambridge cb2 2ru, UK, Published in the United States of America by Cambridge University Press, New York

Getachew Mamo, (2009). Part-Of-Speech Tagging For Afaan Oromo Language. Masters Thesis, Department Of Information Technology, Jimma University, Jimma, Ethiopia

Gezehagn Gutema, (2012). Afaan Oromo Text Retrieval System, School of Information Science Addis Ababa University master's Thesis

Gumii Qormaata Afaan Oromoo, (1995). Caasluga Afaan Oromoo, Jildi. In Komishinii Aadaaf Turizmii Oromiyaa, Finfinnee, Ethiopia.

Jurafky. D, and Martin. H. James, (2006). Speech and Language Processing, An introduction

to natural Language Processing, Computational Linguistics, and speech recognition. 2nd ed. New Jersey, Prentice Hall.

Kanak Mohnot, Neha Bansal, and et.al, (2014). Hybrid approach for Part of Speech Tagger for Hindi language.

Karthik Kumar G, Sudheer K, and Avinesh Pvs, (2006). "Comparative Study of Various Ma chine Learning Methods for Telugu Part of Speech Tagging", *In Proceeding of the NLPAI Machine Learning Competition*.

Kothari, C.R, (2004). Research methodology methods and Techniques , 2nd revised edition For mer principal,college of commerce university of Rajasthan,Jaipur(India). New Age Inter national (P) Ltd., Publishers Published by New Age International (P) Ltd., Publishers.

Levent Altunyurt, Zihni Orhan and Tunga Güngör, (2007). Towards combining rule-based and statistical part of speech tagging in agglutinative languages, computer enginering Boga zici University, Istanbul, Turkey.

Linda Van Guilder, (1995). "Automated Part of Speech Tagging: A Brief Overview", Handout for LING361, Georgetown University.

Manning, C.D. and Schiutze, H, (2000). Foundations of Statistical Natural Language Pro cess ing, 2nd Ed. The MIT Press Cambridge, Massachusetts London, England.

Manoj Kumar C, (2005). "Stochastic Models for POS Tagging", IIT Bombay

Mark Aronoff and Kirsten Fudeman ,(2011). Fundamentals of linguistics, What is morphology? 2nd ed. Wiley-blackwell,A John Wiley & Sons, Ltd., Publication

Martha Yifiru Tachbelie, Solomon Teferra Abate and Laurent Besacier, (2011). Part-of-speech tagging for under-resourced and morphologically rich languages – the case of Amharic. Conference on Human Language Technology for Development, Alexandria, Egypt, 2-5 May 2011.

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

Mekuria Bulcha, (1994). The Language Policies of The Ethiopian Regimes and The History Of Written Afaan Oromo: 101, Vol 1, No 2.

Mesfin Getachew, (2001). Automatic Part of Speech Tagging for Amharic: An Experiment Using Stochastic Hidden Markov (HMM) Approach. Masters thesis, Addis Ababa University.

Mikheev, A, (1996), "Learning part-of-speech guessing rules from lexicon: extension to non-concatenative operations", Proceedings of COLING, pp. 770-775.

Mohammed-Hussen, (2010). Part Of Speech Tagger for Afaan Oromo Language using Trans formational error driven learning (TEL) approach. Master's thesis, Addis Ababa Uni versity.

Mosteller, F. and Wallace, D. L, (1964). *Inference and Disputed Authorship: The Federalist*. Springer-Verlag. A second edition appeared in 1984 as *Applied Bayesian and Classical Inference*.

Nadja Nesselhauf, (2011). Corpus Linguistics: A Practical Introduction

Prajadhip Sinha1, Nuzotalu M Veyie and et.al, (2015). Enhancing the Performance of Part of Speech tagging of Nepali language through Hybrid approach, Asst. Professor, De part ment of Computer Science, Kohima Science College, Kohima, Nagaland Profes sor, Depart ment of Computer Science, Assam University, Silchar, Assam, India.

Python Programming Language: http://www.python.org/ visited on may 15, 2016

Rippaabiliika Dimookiraatawaa Fedaraalawaa Itiyoophiyaatti Ministeera Barnootaa, (2005). Barnoota Afaan Oromoo. Kitaaba Barataa Kutaa 12ffaa.

Robin, (2009). Natural Language Processing.Article on Natural Language Processing.

Sandipan Dandapat, (2009). Part-of-Speech Tagging for Bengali. .

Schiitze, Hinrich, and Yoram Singer.(1994). Part-of-speech tagging using a variable memory Markov model. In *ACL 32*, pp. 181-187.

Solomon Asres, (2008). Automatic Amharic Part-of-Speech Tagging Using Hybrid Approach (Neural Network and Rule-Based). Masters thesis, Addis Ababa University.

Source: http://en.wikibooks.org/wiki/Afaan_Oromo/Alphabet visited may 5, 2016

Steven Bird, (2015). NLTK Documentation Release 3.0

Steven Bird, Ewan Klein, and Edward Loper, (2009). Natural Language toolkit and Python, First Edition. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

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

Teklay Gebregzabiher, (2010). Part Of Speech Tagger For Tigrigna Language, Addis Ababa University,Ethiopia.

Teubert, W, (2001). Corpus Linguistics and Lexicography. International Journal of Corpus Linguistics, Special Issue, pp.125-153

Thomas C. Rindflesch, (1996). Natural Language Processing, Sematic Knowledge Repre sentation research paper, USA.

Appendix A: Some of Extracted Lexical Rules

AX->PR if Word:hin@[0]

NN->VV if Word:.@[1]

NN->VV if Word:Faantuu@[0,1]

JJ->NN if Pos:JJ@[-1] & Pos:JJ@[1]

NN->NNP if Word:hoteela@[-1]

JJ->NN if Word:dhiigaa@[1,2]

JN->NN if Pos:NN@[-1]

NN->JJ if Word:dha@[1,2]

AX->CC if Word:taanaan@[-1,0]

AX->PR if Word:haa@[-1,0]

JC->NC if Word:jireenyaafi@[-1,0]

NN->AD if Word:gadi@[-1,0]

AD->NN if Pos:NNP@[2]

NN->VV if Pos:NNS@[-2,-1]

AD->NNP if Pos:NNP@[1,2,3]

PS->NN if Pos:JJ@[1,2,3]

NN->VV if Pos:PP@[-1] & Pos:AX@[1]

Appendix B: Some of Extracted Contextual Rules

AX -> PR if the Word of this word is "hin

NN -> VV if the Word of the following word is "."

NN -> VV if the Word of words i+0...i+1 is "Faantuu"

NN if the Pos of the preceding word is "JJ", and the

NN -> NNP if the Word of the preceding word is "hoteela"

NN if the Word of words i+1...i+2 is "dhiigaa"

JJ if the Word of words i+1...i+2 is "dha"

NN if the Word of words i-1...i+0 is "Bakka"

CC if the Word of words i-1...i+0 is "taanaan"

AX -> PR if the Word of words i-1...i+0 is "haa"

JC -> NC if the Word of words i-1...i+0 is "jireenyaafi"

NN -> AD if the Word of words i-1...i+0 is "gadi"

JN -> NN if the Pos of the preceding word is "NN"

AD -> NN if the Pos of word i+2 is "NNP"

NN -> VV if the Pos of words i-2...i-1 is "NNS"

AD -> NNP if the Pos of words i+1...i+3 is "NNP"

PS -> NN if the Pos of words i+1...i+3 is "JJ"

NN -> VV if the Pos of the preceding word is "PP", and the

AX -> JJ if the Word of the preceding word is "hidhataa"

AX -> VV if the Pos of word i+2 is "JJ"

VV -> NN if the Pos of the following word is "JJ", and the

AX -> NN if the Word of the following word is "ilaala"

Appendix C: Sample of Training Data

Bakka/PR buutuun/JJ Biirichaa/NN Aadde/NN Faantuu/NN kaleessa/AD meeshaa/NN qorannoo/JJ dhiigaa/NN gargaarsa/VV waldaa/JJ misiyoonota/NN addunyaan/JJ hojjechaa/VV jira/AX ./PN

Amajjii/NN ooluuf/VV hanga/PR ammaatti/AD ajjeechaan/VV raa'wate/VV hedduu/JJ hordofee/VV dhugaan/NN jala/PR ka'udhaan/VV argama/AX ./PN

Gumaa/NN daa'imman/NN adnyaa/NN dammaqinaan/AD to'annaa/VV poolisootiNN godhanVV qofaadha/VV ./PN

Dhalattoota/NN uumamni/NN hirmaataa/VV biyyasaanii/JJ loogiin/VV raaw'atuVV hawaasichi/NN barumsa/NN barbaachisaVV/./PN

Biyyeefi/NC bishaanii/NN isheetiin/PP taasifamne/VV kanuma/PP dhuumamuu/VV seenaniiVV yaadatama/AX ./PN

Sareen/NN yeroo/AD dheeraaf/JP halkan/NN dutti/VV walirraa/PP baasuun/VV xiqqaanAD isaatti/PP kanneen/PR ayyaanichaafi/NC badhaadhina/NN jiru/AX /.PN

Billachi/NN isaanitti/PP namite/vV maraguuf/VV dhandhamatti/VV fixuun/VV ./PN

Midhaan/NN fagaachuun/VV oomishamu/VV qopheesadhu/VV ./PN

Yoo/CC wal/PP hin/AX lolan/VV waraana/NN of/PP harkaa/NN qabu/AX taanaan/AX ofirraa/PP garagalchanii/VV wal/PP rukutu/VV malee/CC ittiin/PS wal/PP waraanuun/VV safuu/AD dhorkaa/VV tureedha/AX ./PN

Bifti/JJ qonnaa/NN kanaa/PP boodaa/PR akka/PR geedaramuuf/VV jiru/AX bu'awwan/JJ qorannoo/NN saayinsii/JJ addeessu/VV ./PN

Waajirichi/NN dhaabbilee/NN 23/JN keessattiPR jijjiirama/VV hojii/NN qoratee/VV hojjeessuuf/VV sochiirra/VV akka/PR jiru/AX beeksisaniiru/VV ./PR

Bara/AD 90/JN eegalee/PR lalisaa/VV jira/AX ./PN

Bishaan/NN jireenyaafi/JC waan/PR barbaachisaa/AX guddaa/JJ akka/PR ta'e/AX beekamaadha/AX ./PN

Waggaa/NN 12/JJ booda/PR ,/PN dubartii/NN jalqabaa/JJ sanyii/JJ gurraachota/NN kessa/PR pirezidaantii/NN yuunvarstii/JJ taatee/AX hojjate/VV ./PN

kan/PR duulee/JJ hin/AX beekne/AX hidhataa/VV bula/AX ./PN

Waanni/PP cimaan/JJ ammoo/CC ciminasaatiin/JJ yoo/CC itti/PP fufe/VV kan/PR maqaan/JJ isaa/PP tolee/JJ mul'atu/AX ./PN