ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

PART OF SPEECH TAGGER FOR TIGRIGNA LANGUAGE

BY
TEKLAY GEBREGZABIHER ABREHA

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

November, 2010

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

PART OF SPEECH TAGGER FOR TIGRIGNA LANGUAGE

BY
TEKLAY GEBREGZABIHER ABREHA

Signature of the Board of Examiners for Approval

	Name	Signature
1.	Dr.Sebsbie Hailemariam,Advisor	_____
2.	_____	_____
3.	_____	_____

Dedicated to:

1. My Mother Mebrat Gebremicheal

***Mom**!! You were dedicated to change my life! You have walked all the way you can to change and to educate me though you did not see your effort! May your soul rest on heaven!*

2. My Father Gebrgzabiher Abreha

***Dad**!! You made me strong, motivated and passionate person. I am here because of your inspiration during my childhood though you are not lucky to see your passion, courage and motive in your SON. May your soul rest on heaven!*

3. **Ashenafi Taddesse**

I wish I could call you everything for me! My brother, my father my mother… you have helped me in my entire career starting from my high school to university in everything.

Acknowledgments

Let all the praise and thanks be to the supernatural power and creature of the entire universe, almighty God for helping me to realize this work. Then, I was very lucky enough to have had the support of many people and without these people the completion of this thesis work would have been very thorny. Firstly I would like to thank my advisor **Dr. Sebsbie Hailemariam**. I thank you **Dr**. for your encouragement, guidance, understanding and motivation throughout this thesis work. You have showed me how researches are produced and how to tackle a problem by investing your invaluable time. I really thank you! Throughout this thesis work, you have been consistently sharing me constructive ideas and comments about this work in a friendship mode which I love most. Really you are a model advisor. Special thank goes to **Abreham Girmay** who has been always with me during the entire data preparation for this work; without him, it would be impossible to finish this work. **Solomon Asres,** you have shown me how to proceed with my work from the very beginning. I thank you a lot. **Beza M.**, I thank you for your invaluable comments.

TT, I thank you a lot for your support and understanding in my career. You are the positive and passionate person who has thought me what life is. It is also my pleasure to express my gratitude to many of my friends who have helped me in this thesis work. Especially, many thanks to my classmates: **Abebe A. Abel T., Desta B., Getasew T.**, **Mandefro K. Selama G. Moges A. Tesfaye G.,** and my colleagues: **Abreham H. Kewani W., knife T., Mohammed A. and Tulu T.**

I am so tongue tied to express my thanks to **Ashenafi Taddesse** (with his mother **Asegedech G**., his father **Taddesse G.,** his sister **Frey T.** and his brother **Measho T.**) who has helped me to be the man here. He is the ideal person of mankind. He is my father, mother, sister, brother… whom I have met him in one circumstance but who became everything for me later. Finally, I would like to thank everyone who has contributed negative and positive impacts to the successful realization of this thesis work, as well as expressing my apology that I could not mention individually one by one.

TABLE OF CONTENTS

Contents Page

List of Figures iv
List of Tables v
Acronyms and Abbreviations vi
Abstract vii
CHAPTER ONE 1
INTRODUCTION 1
1.1 Background 1
1.2 Statement of the Problem 3
1.3 Objective of the Study 4
1.3.1 General Objective 4
1.3.2 Specific Objectives 4
1.4 Methodology 5
1.4.1 Identification of POS Tagset for Tigrigna Language 5
1.4.2 Data Collection 5
1.4.3 Modeling 6
1.4.4 Tools and Implementation 7
1.4.5 Experimental Analysis 7
1.5 Application of Results and Beneficiaries 7
1.6 Scope of the Study 8
1.7 Organization of the Thesis 8
CHAPTER TWO 9
LITERATURE REVIEW AND RELATED WORKS 9
2.1 Literature review 9
2.1.1 Rule Based Approach 10
2.1.2 Stochastic Approach 11
2.1.2.1 Hidden Markov Model 14

2.1.3 Artificial Neural Networks (ANN) 16
2.1.4 Hybrid Approach 18
2.2 Related Works 18
2.3 Summary 25
CHAPTER THREE 27
TIGRIGNA LANGUAGE AND TAGSET PREPARATION 27
3.1 Overview 27
3.2 Tigrigna Sentence Structure 27
3.3 Word Classification of Tigrigna 29
3.3.1 The need for categorizing Words 29
3.3.2 Tigrigna Word Classes 29
3.4 Tigrigna tags and Tagsets 35
3.5 Summary 41
CHAPTER FOUR 43
DESIGN OF TIGRIGNA POS TAGGER 43
4.1 Introduction 43
4.2 Approaches and Techniques 43
4.3 Design Goals 44
4.4 Designing Hidden Markov Model (HMM) Tagger 44
4.4.1. Lexical Model 46
4.4.2. Contextual Model 46
4.5 Designing Rule Based Tagger 49
4.5.1. Transformation-based error-driven learning 49
4.5.2. The initial state Tagger 51
4.5.3. Rules 52
4.5.4. Learning Phase 52
4.5.5. Brill Tagger Architecture 55
4.6 Hybrid Tagger Architecture 56
4.7 Summary 58
CHAPTER FIVE 60
IMPLEMENTATION OF TIGRIGNA PART OF SPEECH TAGGER 60

5.1 Introduction 60
5.2 Corpus Preparation 60
5.3 Implementation of Preprocessing components 62
5.4 Implementation of Hybrid Tagger 63
5.5 Summary 64
CHAPTER SIX 65
EXPERIMENTS AND PERFORMANCE ANALYSIS 65
6.1 Introduction 65
6.2 Experiments with HMM tagger 65
6.3 Experiments with Rule based tagger 66
6.4 Experiments with Hybrid tagger 68
6.5 Performance Analysis 69
6.6 Summary 75
CHAPTER SEVEN 77
CONCLUSION AND RECOMMENDATION 77
7.1 Conclusion 77
7.2 Recommendation 78
References: 79
Appendices 84
Appendix A : sample corpus 84
Appendix B: transliteration from Tigrigna Fidel to Latin characters 85
Appendix C: Brill tagger learned rules 86

List of Figures Page

Figure 1.1 English SNoW based part of speech tagger 3
Figure 3.1 Tigrigna sentence structure 28
Figure 3.2 prepositions as a separate word 32
Figure 3.3 different adverbs of Tigrigna 33
Figure 3.4 tagset concept hierarchy 36
Figure 4.1 the HMM tagger trainer model 47
Figure 4.2 HMM tagger and evaluating process 48
Figure 4.3 Transformation-error driven learning of Brill tagger 51
Figure 4.4 Adapted Brill Tagger for Tigrigna 56
Figure 4.5 Hybrid tagger high level view 57
Figure 4.6 technical description of Hybrid tagger 57
Figure 5.1 Hybrid tagger algorithm 64
Figure 6.1 HMM tagger performance curve analysis 66
Figure 6.2 Rule based tagger performance curve analysis 67
Figure 6.3 performance analysis of hybrid tagger 68

List of Tables Page

Table 3.1 Tigrigna Tagsets 42
Table 6.1 HMM tagger performance 65
Table 6.2 Rule based tagger performance using different initial state taggers 66
Table 6.3 Hybrid tagger performance on different threshold values 68
Table 6.4 POS tags frequency 69
Table 6.5 Rule based tagger confusion matrix 70
Table 6.6 HMM Tagger Confusion matrix 72
Table 6.7 Hybrid tagger Confusion matrix 74
Table 6.8 performance improvement by the hybrid model compared to HMM model 76

Acronyms and Abbreviations

ANN	Artificial Neural Network
HMM	Hidden Markov Model
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
POS	Part of Speech
SNoW	Sparse Network of Winnows
TEL	Transformational Error driven Learning

Abstract

Due to many sophisticated and advanced technologies like the Internet, the world has become a single village. It is possible to get a vast amount of digitized information that are generated, propagated, exchanged, stored and accessed through the internet and other media like mobile network each day across the world. The accumulation of digital data is making information acquisition increasingly difficult, with natural language becoming critically an obstacle. The step towards tackling this obstacle is Natural Language Processing. And part of speech tagging is one and preliminary among the many steps that are used for information acquisition and other advanced NLP applications. It is a technique of labeling each word in a text/sentence with its corresponding part of speech category that best suits the definition of the word as well as its context in the particular position of the sentence in which it is used.

As far as the researcher‟s knowledge is concerned, there is no part of speech tagger developed for Tigrigna language though there are many part of speech taggers developed using different approaches for many languages such as English, Arabic, Amharic, etc. Therefore, this work proposes a hybrid approach, HMM tagger combined with rule based tagger, for Tigrigna part of speech tagger. Tigrigna literatures on grammar and morphology are reviewed to understand nature of the language and also to identify possible tagsets. As a result, 36 broad tagsets were identified and 26,000 words from around 1000 sentences containing 8000 distinct words were tagged for training and testing purpose. Since there is no readymade standard corpus, the manual tagging process to prepare corpus for this work was challenging and hence, it is recommended that a standard corpus is prepared. Raw Tigrigna text is first tagged by the HMM tagger; afterwards the rule based tagger is used as a corrector of the HMM tagger. Viterbi algorithm and Brill Transformation-based Error driven learning are adapted for the HMM and Rule based taggers respectively. Different experiments are conducted for HMM based, rule based and hybrid based approach taking 25% of the whole data for testing. The HMM and rule based approach shows an accuracy of 89.13% and 91.8% respectively whereas, the hybrid model improve the accuracy to 95.88%. Hence, it is found that that the hybrid of the two taggers outperforms the individual taggers.

Keywords: Tigrigna, POS tagger, NLP, Brill Tagger, Hidden Markov Model, Hybrid POS tagger

CHAPTER ONE
INTRODUCTION

1.1 Background

Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications in a computer [25]. There are multiple methods or techniques from which to choose in order to accomplish a particular type of natural language processing. NLP can deal with any natural language mode, type etc. Natural languages exist in various forms such as written, spoken and signed. The above definition refers to all possible forms of natural language as text but this paper entirely uses this term consistently to mean a written form of natural language.

NLP usually involves one or more level of linguistic analysis such as word level, phrase level, sentence level, semantic level, etc. There are processes made when humans produce or comprehend language. It is thought that humans normally utilize all of these levels since each level conveys different types of information. But, various NLP systems utilize different levels, or combinations of levels of linguistic analysis, and this is observed as a difference amongst various NLP applications. NLP refers to Human-like language processing which reveals that it is a discipline within the field of Artificial Intelligence (AI). Moreover, NLP is the means for accomplishing different types of tasks and/or applications. Such tasks include Part of Speech (POS) Tagging, Named Entity Recognition (NER), Information Retrieval (IR), Speech Recognition, Machine Translation, Question Answering etc [25].

Part-of-Speech (POS) tagging is a technique of assigning each word of a written text with an appropriate parts of speech tag. The significance of part-of-speech (also known as word classes, morphological classes, or lexical tags) for language processing is that it gives large amount of information about a word and its neighbour. POS tagging can be used in Text to Speech (TTS), Information Retrieval (IR), shallow parsing, Information Extraction (IE), linguistic research for corpora [7, 11] and also as an intermediate step for higher level NLP tasks such as parsing,

semantic analysis, machine translation, and many more [7]. POS tagging, thus, is a necessary application for advanced NLP applications in Tigrigna or any other languages.

Users are in need of using Information Technology in such a way that it can simplify their life. This striking interest of users brings to the idea of NLP, and Part of speech tagger is at the very heart of developing many NLP applications. For the development of many advanced NLP applications for a specific natural language, it will be a helpful tool to develop Part of Speech Tagger. For example, if information is needed to be extracted from a text of some pages, it would be very appropriate to know the words" part of speech category in the text. That is, it is better if there is a classification model of the words that map words to their part of speech categories such as Noun, pronoun, verb, adjective etc in the given text before trying to extract the information in need. The same holds true also in Information Retrieval (IR), Question answering and parsing.

Nowadays Part of Speech tagger is developed for different languages and it remains an intensive research area for other different languages. Among the languages with POS tagger developed are English, German, Dutch, Chines, Arabic, Bangla etc [8, 9, 11, 23, 26]. As to the best of the researcher"s knowledge, Tigrigna is then a language which does not have any POS Tagger developed so far.

The POS tagger for languages such as Arabic, German, Portuguese, Czech, Greek, Russian, Dutch, Hungarian, Spanish, English, Italian, Finnish, French, and Polish is implemented and is available on the web for use by some other researches for other languages [23]. For example, it is possible to see that of the English Sparse Network of Winnows (SNoW) based POS tagger [39], given a text „**Teklay has eaten his lunch.**" It generates POS information as shown in figure 1.1. The SNoW based English POS tagger [39] is a learning architecture that has used around fifty tagsets for classifying words in a given text into their correct part of speech. In this demo NNP is a tagset for Proper singular noun, VBZ for verb 3rd person singular, VBN for past tense verb, and PP for Possessive Pronoun.

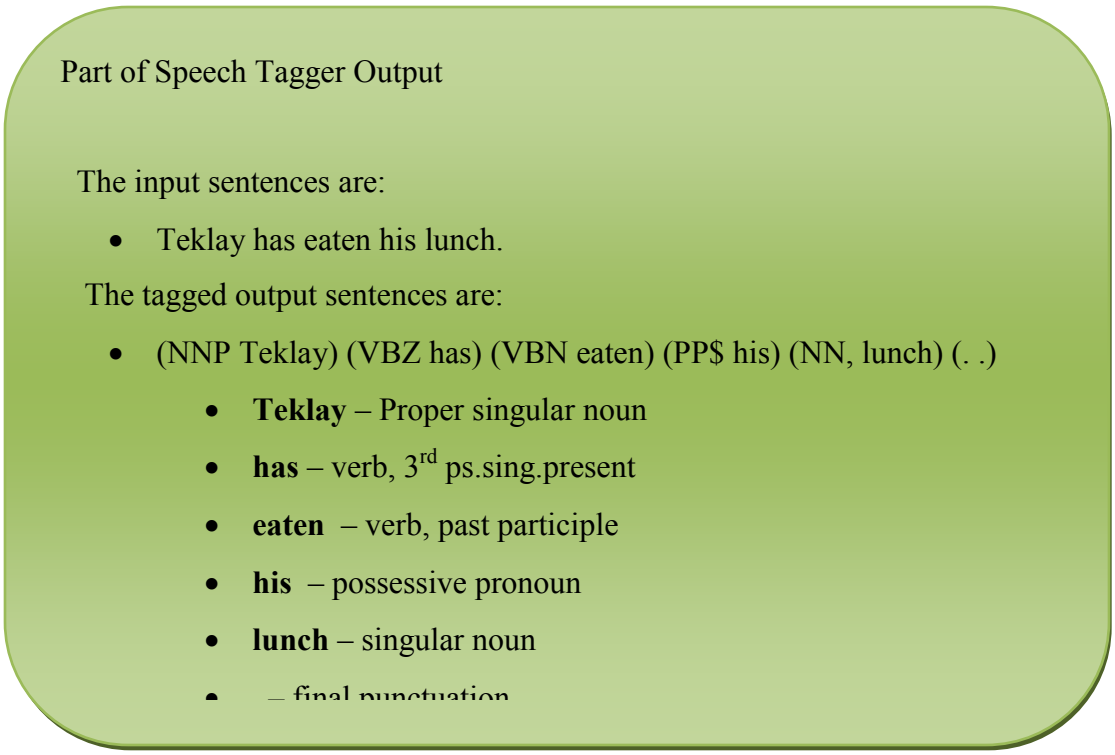


Figure 1.1 English SNoW based part of speech tagger [30]

Tigrigna (ትግርኛ, tigriññā), also spelled Tigrigna, Tigrina, Tigriña, less commonly Tigrinian, Tigrignan, is a Semitic language spoken by the Tigray people in Tigray region [Northern Ethiopia] and in central Eritrea, where it is one of the two dominant languages of Eritrea. Tigrigna is the official language of Tigray region and it is spoken among groups of emigrants from these regions, including some of the Bet-Israel now living in Israel. Tigrigna is also spoken by the Jeberti (Muslim Tigrigna) in Eritrea [43]. It is written with a version of the Ge'ez script and first appeared in writing during the 13th century [41]. The Tigrigna language has its own distinct way of grammar construction, character representation called ፊደል (fidel) and sentence formation as revealed by [41, 49, 50].

1.2 Statement of the Problem

As far as the researcher‟s knowledge on NLP is concerned, researches made in the area of Tigrigna natural language processing are very limited in number. In fact, the morphological analyser developed for Tigrigna language by Indiana University, school of Informatics and computing [12] is the functional NLP application that is found. Apart from this, there is also a

spell checker as an NLP application developed for Tigrigna language in Eritrea [2]. In addition, there is also an NLP research conducted entitled "A stemming Algorithm Development for Tigrigna Language Text documents" by Girma Berhie in Addis Ababa University that reduces words into their stem (root or base) form [15]. However, there is no research conducted on POS Tagger development for Tigrigna language which is becoming a barrier for researches of higher level NLP applications in the language like Information Extraction for Tigrigna that can be done using different techniques though it may not be efficient without using POS tagger. In fact, there are part of speech taggers developed for local languages like Amharic. But, these part of speech taggers can not be used for Tigrigna.

Researches in the area of POS tagging will contribute a lot in the effort of Natural language processing of Tigrigna language. The absence of POS Tagger system limits researches concerning the NLP of Tigrigna language such as parsing (syntactic and semantic), machine translation, sentence grammar checker, spell checker, speech synthesis etc as it is used as a pre-processing component for the aforementioned NLP applications. Hence, conducting research and developing an automatic Part of Speech Tagger for Tigrigna language worth paramount significance.

1.3 Objective of the Study

1.3.1 General Objective

The general objective of this research work is to develop Part of Speech Tagger model for Tigrigna and analyze the performance of the model.

1.3.2 Specific Objectives

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

- Review, Analyze and Study the basic word category of the Tigrigna language with the aim of identifying the POS tags that in turn helps get the tagsets for the ease of

simplifying computer representations of the structural or linguistic information contained in the Tigrigna language.

- Review, Analyze and study the structure of the Tigrigna sentence.
- Study the morphological property of the language to identify properties useful to POS Tagger.
- Collect and design corpus for training and testing of the system
- Asses the various approaches for the development of POS Tagger for the language.
- Design and model a POS Tagger for the language.
- Develop a Tigrigna POS Tagger Prototype.
- Test and Analyze the system performance
- State conclusion and recommendation based on the experimental results

1.4 Methodology

1.4.1 Identification of POS Tagset for Tigrigna Language

So far, there is no readymade tagset for Tigrigna that can be used in this thesis work. Hence, the Tigrigna word classes and grammar construction are assessed from the works of [49, 50] and discussions were made with the language experts in order to set tagsets for this language. In addition, there is a work [14] done on the annotation of Amharic news text which is used as a reference in the identification of tagsets for Tigrigna as both the languages are Semitic group and are closely related.

1.4.2 Data Collection

Tigrigna news texts were collected from different sources both in softcopy and in hardcopy. Such sources include websites of Dimtsi Woyane Tigray, Mekalh Tigray, Woyen Newspaper, FM Mekelle, Tigray Development Association and Ethiopian Radio and Television Agency Tigrigna Department as the news collected from these sources can be representative of the language from the perspective of the news domain. After a thorough review on different works of part of speech tagger, it was concluded that large data set is needed and hence around one

thousand Tigrigna sentence with around 26, 000 words containing around 8000 distinct words were collected. Though this is not as large as that of the Brown corpus which has around one million words and Amharic corpus [14] which has around two hundred fifty thousand words, it is possible to say that, as this is the first work in Tigrigna, it is sufficient for this thesis work as the performance learning curve of the model showed small difference of performance using this data set during data size testing. After the raw news texts are collected, the first two hundred sentences were given to language experts to annotate it manually on paper using the identified tagsets. These tagged texts were used as a corpus for training the prototype tagger. Then this trained tagger takes untagged text as an input and tags the words based on the knowledge that it has acquired during the training and gives tagged text as an output. The output of the tagger is taken and given to the language professionals for correction and approval. After the corrected and approved tagged text is obtained the corpus is updated which is used in turn for training of the final POS tagger model. This process is repeated until the desired size of the corpus is attained. Putting it altogether, an incremental approach is used for the preparation of the tagged corpus. Finally the corpus is divided randomly into training set and testing set.

The desired size of corpus can be obtained through tracking the curve analysis that is made during the experiment. At the very beginning, the testing set is fixed. Then 10% of the training set is taken to train the system. The system is then tested on the testing set to measure its performance. After getting low performance of the system, it is retrained using 20% of the training set. Again, the system is tested on the fixed testing set and measured its performance. The experiment was continued by adding the size of the training set until no performance increment of the tagger is obtained regardless of adding the training data. This implies that a sufficiently large data set is obtained for training and testing provided that the performance curve remains the same regardless of the increment in the size of the training set.

1.4.3 Modeling

In this thesis work, three different models were experimented namely the rule based tagger, HMM based tagger and the hybrid tagger (combination of the rule based tagger and HMM based tagger). The rule based approach as its name indicates relies on rules which are either

handcrafted or machine learned rules. The rules are the important elements for annotating words in the rule based approach. Brill Transformation error driven learning approach is adapted for the rule based tagger where rules are automatically learned from a manually annotated corpus. The HMM based Tagger relies on the statistical property of words along with 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. Both models, the rule based and HMM based taggers, have their own pros and cons and in order to optimize the disadvantages of the two, a hybrid model that takes the attractive properties from both models is proposed for this thesis work. The hybrid is done as the HMM tagger followed by the rule based as a corrector.

1.4.4 Tools and Implementation

In conducting the research on POS Tagger for Tigrigna text, an open source Natural Language ToolKit (NLTK) and Python programming language are 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 [3, 35, 38]. Python is an easy to learn but powerful programming language especially for text processing in NLP applications. It has efficient high level data structures and a simple but effective approach to object-oriented programming [3, 28, 35].

1.4.5 Experimental Analysis

Once part of speech tagger model is developed, testing is conducted on each model. The model is trained on 75% of the entire collected data and the remaining is used for testing purpose. The performance of the model on each category of part of speech is evaluated. A confusion matrix is generated for each model and analysis of the confusion matrix is made on each model. Accuracy is taken as the performance measure of the model. Accuracy is nothing but the closeness of the agreement between the test result and the accepted reference value (the manually tagged text (gold data) of the test set).

1.5 Application of Results and Beneficiaries

As the researcher has stipulated out in section 1.1, there are many advantages of developing POS Tagger for a specific language. In the first place, it is the basis for developing other higher level

applications of NLP such as parsing, information extraction, information retrieval, question answering etc. These applications can be used in different areas of the Tigrigna language. Accordingly the beneficiaries of this study are:

- Researchers who want to conduct on higher level application of NLP for this language such as Parsing, Spell checking, grammar checker, speech recognition, Question Answering etc.
- People who want to learn Tigrigna as a second language; it may help them to discover the word categories and grammar construction.

1.6 Scope of the Study

The corpus developed in this thesis work is domain specific corpus, a text corpus that is collected from a single domain in this case news domain only. Moreover, during the development of the corpus, the tagsets used are meant to give information of words about their word class category but not about the issues like gender, number, tense etc. The basic problem while doing this research work was the lack of data corpus needed for building a model and testing. Moreover, there are limited NLP researches done for Tigrigna language and hence there have been difficulties of using previous works as a reference.

1.7 Organization of the Thesis

The whole thesis is organized into seven chapters. The first chapter describes the introductory part. The second chapter is all about literature review and related work. It describes the methods used so far for POS tagging and works that are done using Hybrid approach, combination of rule based and stochastic based. Chapter three focuses on study and assessment of the nature and structure of Tigrigna sentences, word classifications and tagset preparation for Tigrigna. After the thorough study of the word classifications and sentence structure of the language, chapter four deals with architecture and design of the POS Tagger for the language. Afterwards, the fifth chapter deals with corpus preparation and implementation of the preprocessing tools and algorithms for the architecture stipulated out in chapter four. Chapter six manly focuses on the experimental analysis of the part of speech tagger. Finally, the last chapter is all about drawing conclusion that comprises both summary of the work done and future work.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

2.1 Literature review

So far many POS tagging researches have been done and different approaches have been used for POS tagging, where the well-known ones are rule-based, stochastic, Artificial Neural Networks and Hybrid Approach. Rule-based taggers, as their name implies, [4, 10, 11] strive to assign a tag to each word using a set of handwritten rules that might be specified by language experts or machine learned rules. These rules could determine, for example, that a word following a determiner and an adjective must be a noun. In this approach, what researchers should do is set and check rules properly with the help of language professionals or to make the taggers learn rule during the training phase of the system as it is in the case of the Brill tagger [4, 10]. The stochastic (statistic or probabilistic) approach [11, 26] uses a training corpus to pick the most probable tag sequence for a word sequence in a given sentence to be tagged. Some stochastic methods are based on first order or second order Markov models and some are based on a few other techniques which use probabilistic approach for POS Tagging, such as the Tree Tagger [18]. Artificial Neural Network (ANN) uses a training corpus and adaptively learns properties of words to pick the appropriate tag for a word in a given sentence [34].

Finally, the hybrid approach may either combine the rule-based approach and statistical approach or the rule based approach and the Artificial Neural Network. The hybrid of rule based and stochastic approach for example may pick the most likely tag based on a training corpus and then applies a certain set of rules to see whether the tag should be changed to another tag or not. Besides it saves any new rules that it has learnt in the process, for future use. One example of an effective tagger in this category is the Brill Tagger [4, 10, 11, 27, 45].

This chapter mainly deals with the most common approaches to POS Tagging and related woks to this thesis work. Accordingly the approaches used so far are described in detail in the following sections.

2.1.1 Rule Based Approach

The rule based approach uses rules to disambiguate tags of words. The rules are based on knowledge of the specific language which may consist of a large number of morphological, lexical and syntactical information. These rules can be obtained manually that are handcrafted by linguistic professionals or through machine learning [10, 34]. The former way of getting rules is tedious, time taking since it requires linguistic professionals to manually set rules. Moreover, it is inconsistent and subjective as it is determined by the understanding of one or more linguistic specialists and their skill and knowledge of the specific language [34], while the later way of obtaining rules as it is explained in the work of [10], transformation-error driven approach, is from a training corpus. That means a model is made to automatically learn and store rules (also called Brill transformations) from the training corpus to be provided. There is no way of specifying the rules manually by linguistic professionals, what is needed is the tagged corpus as an input to automatically drive its own rules so called transformations in Brill tagger [10].

In addition, the rule based approach also uses contextual information to assign tags to unknown words. These rules are often known as context frame rules [24]. A context frame rule can be, for example, if an unknown word is preceded by determiner and followed by a noun, its correct tag is adjective. Moreover morphological information can be used as a rule to aid in the tagging process. One particular example in this case is, if a word ends with „-ing" and is preceded by a verb the most probable tag of the word is verb [24].

In fact, adding a rule to a system may involve over-generation, i.e., one extra rule can result in more harm to the accuracy of general tagging in machine-learning rule-based approach. The manual rule-based tagging system has also the aforementioned limitations [34]. Therefore, in general, rule-based approaches are time-consuming and require a great knowledge of a specific language being tagged. But it is also possible to find some advantages of the rule based approach that is listed in the work of [4]. These are: Robustness, a vast reduction in stored information required the lucidity of a small set of meaningful rules, ease of finding and implementing improvements to the tagger, and better portability from one tag set or corpus type to another. Generally the basic characteristics of the rule based approach are clarified as advantages and disadvantages below:

Advantages:

- requires only small amount of training data
- useful for limited domain
- Can be used with both well-formed and ill-formed input
- High quality based on solid linguistic

Disadvantage:

- Most of the time, it relies on hand-constructed rules that are to be acquired from language specialists and construction of these rules is tedious and time consuming.
- development could be very time consuming
- not easy to obtain high coverage of the linguistic knowledge
- some changes may be hard to accommodate

2.1.2 Stochastic Approach

The stochastic approach also called statistical approach is based on a probabilistic pattern to assign a probable part of speech tag to a given text from a given training text corpus. The goal of any stochastic approach is to pick the most probable tag for a word from its context and its neighbors [13, 34]. They can build a probability matrix that stores the probability of an individual word belonging to a certain part of speech and its distributional probability. They can use this distributional probability to tag words that are in the input sentence but not in the training corpus. The probabilities are estimated from a tagged training corpus or an untagged corpus. Stochastic tagging techniques can be of two types depending on the training data. Supervised Statistical tagging techniques use tagged corpus for their training though it requires large amount of tagged data so that high level of accuracy can be achieved. Unsupervised Statistical techniques, 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. tagsets), and based on these automatic groupings, they calculate the probabilistic values needed by statistical taggers.

The basic idea of this approach is to find the probability (p) of a word along its tag from a given sentences or text which can be represented mathematically as:

$$p(Wi,Ti|<S>)$$

Where W_i, T_i are the i^{th} word and the i^{th} tag in the input sentence or text <S>.
The stochastic approach may use the most frequent tag, N-gram analysis or Hidden Markov Models to disambiguate tag of words which can be derived from the above mathematical representation.

The most frequent tag model as its name implies tries to pick the most frequent tag for a given word in a sentence. This model is the simplest model in the stochastic approach as it simply finds the most frequent tag from the training corpus. This can be done by counting the occurrence of the specific word W_i associated with a tag T_i and dividing it by the total occurrence of the word W_i in the training corpus which can be represented mathematically as:

$$P(T_i|W_i) = \frac{count\ of\ (Ti,Wi)}{count\ of\ Wi}$$

This implies that the most frequent tag model computes the probabilities of observing each word attached with every part of speech tags during the training phase. Then in the tagging process of new text, it will pick the tag with most probable tag for that word. This model has very clear limitations as it does not try to look into the sentence structure. This can be solved using the N-gram model that deals with local context of words in a sentence.

The N-gram model is a mechanism of dealing local context of words in a given sentence. It is originally conceived as a technique of predicting the next element of a sequence given only the N-1 previous elements [7]. The elements can be sequence of words, tags or both words and their corresponding tags. This implies that it can be used for finding the tag sequence probabilities (probability of a tag T_i given the previous N-1 tags ($P(T_i|T_{i-1},T_{i-2}, \ldots, T_n)$) or word sequence probabilities (($P(W_i|W_{i-1},W_{i-2}, \ldots, W_n)$). Moreover, the n-gram can also be used for finding the probability of the tag of a current word given the previous n words ($P(T_i,W_i|W_{i-1}, Wi_{-2}, \ldots, W_n)$). It solves the problem of the most frequent tag model which ignores the local context of words. This model decides the appropriate tag for a word by computing the probability that it occurs

within the n-previous tags, where the value of N is considered to be 1, 2, or 3 for practical purposes [13]. These are known as unigram, bigram and trigram models respectively.

The Hidden Markov Model (HMM) is the most widely used model for part of speech tagging under the stochastic approach [31, 46, 47]. The main idea of the HMM is to find the sequence of tags for a given sequence of words. This can be done by combining the most frequent tag and the N-gram model that considers the tag sequence probabilities i.e. considering the lexical category of a word using its most frequent tag and its local context in the sentence from the training corpus during the training phase. Generally speaking, from a statistical point of view, the task of the HMM model is to find the most likely POS sequence $T_1, T_2, \ldots T_n$ for a given word sequence $W_1, W_2, \ldots W_n$. In other words, the model has to maximize the conditional probability $P(\overrightarrow{T}, \overrightarrow{W})$ of the tag sequence given the word sequence over all possible tag sequences T_n [7]. Putting it all together, this approach tries to maximize the probability $P(T_1, T_2, ...,T_n \ldots\ldots T_n \mid W_1, W_2, ...,W_n)$ which can be achieved by the help of N-gram model and most frequent tag model. The details of HMM model is explained in section 2.1.2.1

Generally, the characteristics of the stochastic method are listed below as advantages and disadvantages:

Advantages:

- researchers may not need language specialists, expertise
- Coverage depends on the training data

Disadvantages:

- requires large amount of annotated training data (very large corpora)
- some changes may require re-annotation of the entire training corpus in the supervised statistical learning
- Not easy to work with ill-formed input as both well-formed and ill-formed are still probable

2.1.2.1 Hidden Markov Model

The most common model for stochastic approach is the Hidden Markov Model (HMM). It is the probabilistic function of Markov Process, a process which moves from state to state, from left to right on the states, to find optimal state sequence. An HMM is characterized by the following criteria [16]:

- A finite set of states each of which is associated with a probability distribution
- Transitions among the states are governed by a set of probabilities called transition probabilities.
- In a particular state an outcome or observation can be generated according to the associated probability distribution. The observation is visible and the states are hidden to the observer and hence the name Hidden Markov Model.

HMM is defined formally as a set $\{S, O, A, B, \Pi\}$ Where [16]:

- S denotes the set of states
- O denotes the set of observation symbols.
- $A = \{a_{ij}\}$ is a set of state transition probabilities represented in transition probability matrix in which each a_{ij} represents a probability of moving from state S_i at time t into state S_j at time t+1.

The state transition probabilities can be defined as $a_{ij} = P(S_{i+1} = j|S_i = i)$ for $1<=i<=n$ where n is the total number of states, $a_{ij>=0}$ and $\sum_{i=1}^{n} a_{ij} = 1$

- $B = b_j(k)$ is an emission or observation probability distribution in each of the state S. bj(k) is the observation probability of observation k at the j^{th} state.

The emission/observation probabilities $b_j(k)$ can be computed as $b_j(k) = P(O_i = k |S_i = j)$ for $1<=j<=n$ and $1<=k<=m$. $b_j(k)$ is the probability of state j taking the symbol O_i and it should be greater or equal to zero.

- The initial state distribution $\pi = \{\pi\}$ which is the probability of the first observation at a given state S_i.

Generally an HMM is the set containing $\{S, O, \lambda\}$ where:

$S = \{S_1\ S_2\ S_3\ S_4, ..., S_n)$

$O = \{O_1\ O_2\ O_3\ O_4, ..., O_m)$

$\lambda = \{A, B, \pi\}$

HMM takes three assumptions into consideration. The first assumption is called Markov assumption that states the first order transition probability can be extended to the k^{th} order transition probability. It implies that, if it is possible to find the probability of state S_i given the previous state S_{i-1} ($P(S_i|S_{i-1})$), it would be also possible to find the probability of state S_i given the previous K states ($P(S_i|S_1, S_2, ...S_k)$) [19]. The second assumption is called the stationary assumption that says: state transition probabilities are independent of time and the output. It takes the due consideration of the actual time at which the state transition takes place and the output symbol that can be emitted being on the particular state [19]. As a result, it considers that state transition probabilities are independent of the actual time and output symbol, which can be represented mathematically as:

$$P(S_{t1+1}|S_{t1}) = P(S_{t2+1}|S_{t2}) \text{ for any time } t_1 \text{ and } t_2.$$

The third assumption is the output independence assumption that states: the current observation is statistically independent of the previous observations. This means the probability of observing an output symbol O_i being in state S_i is independent of the probability of observing O_{i-1} being in state S_i [19]. This can be represented mathematically as $P(O_i|S_1, S_2, S_3 \ldots S_n)$. Because of these assumptions, HMM fails to accurately find the most likely sequence of states for a given sequence of observations.

When the HMM model is taken to the application of POS Tagging, the hidden states are the POS tags (tagsets) and the sequences of words are the sequence of observations. The transition probability in POS tagging is the probability of moving from one tag to the next tag and the emission probability is the probability of getting a word W_i being in tag T_i. The main problem of the HMM from the POS tagging point of view is, finding the sequence of tags thereby maximizing their probabilities given a sequence of words in a text. This can be done using the Viterbi Algorithm that will take the joint probabilities of the lexical probability and the n-gram probability [16]. The explanation of Viterbi Algorithm is given in section 4.4.2 of chapter four.

2.1.3 Artificial Neural Networks (ANN)

Here a brief description of ANN is given. To begin with, the definition of ANN is as follows according to the two studies mentioned below:

According to [52]:

> *A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.*

Again according to [51]:

> *A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*
>
> 1. *Knowledge is acquired by the network through a learning process.*
> 2. *Interneuron connection strengths known as synaptic weights are used to store the knowledge*.

Though the field of Artificial Neural Network was established before the advent of computers, the ANN simulations appear to be a recent development. This implies that the information processing of ANN was known in biological nervous system before actually applying it in information processing using computers applications [20]. Hence, the basic idea of ANN here is to process information in a similar way that the biological nervous system process pattern [20]. The key element of the ANN information processing scheme is the novel structure of the information processing system which is composed of, like the biological nervous system, a large number of highly interconnected elements called neurons working in union to solve a specific problem [20, 27, 34]. Artificial Neural Networks learn from example by configuring for a specific application such as pattern recognition or data classification. The ANN can learn by

adapting different behavior on the basis of the data that is given to the network. It is possible to call the ANN learning an adaptive learning as the network is able to find properties from the presented data. It is not necessary to tell the network how to react to each data input separately like the conventional programming.

The common types of ANNs consist of three layers of units namely a layer of input units, a layer of hidden unites and a layer of output unites [1, 20, 34]. The input layer which is connected to the hidden layer represents the raw information that is fed to the network as an input so that it can learn and adapt properties. The middle layer, so called hidden layer, connected to the output layer is determined by the activities of the input unit and the weights on the connections between the input and hidden units. The output layer represents the result of the learning properties from the input layer and hidden layer.

Taking the ANN approach to the application of part of speech tagging, first preprocessing activities are performed before dealing actually with the ANN based part of speech tagger. Such preprocessing activities can be tokenizing, feature extraction like POS information, word information, POS category and order information etc. The result of the preprocessing activities are given to the input layer of the network from which the network can learn pattern. As mentioned above, the input layer is connected to the hidden layer and in this layer different algorithms like error back-propagation algorithm, an algorithm based on an error-correction learning rule specifically on the minimization of the mean squared error which is a measure of the difference between the actual and the desired output, can be used for training the system [1, 34].

This technique of tackling the problem of assigning part of speech tags to words has some disadvantages compared to the HMM and rule based approaches. Some of these are: The HMM method assigns the sequence of tags for the sequence of words in the entire sentence i.e. it takes the due consideration of the sentence structure while most ANNs take the word to be tagged only. The same thing is true with rule based approaches which tend to take the sentence structure and generally the linguistic patterns into consideration [32].

2.1.4 Hybrid Approach

As its name implies, this approach takes the combination of either stochastic approach and rule based approach or ANN and rule based by taking the advantages from both approaches to improve the performance of the system. Works like [4, 8, 10, 48] have used the hybrid approaches (rule based + stochastic) as a result they have got better results than the corresponding uncombined approaches. Another work [34] by Solomon Asress has used a hybrid approach (Neural Network and Rule based) and hence has got a better performance than the individual performance of the Neural Network and Rule based.

2.2 Related Works

In this section, different works on part of speech tagger are presented starting from Semitic languages followed by Cushitic languages and others. The order of presentation of the works in this section is selected based on the similarities of the languages with Tigrigna.

The first Amharic Part of Speech Tagger using hybrid (Neural Network and rule based) Approach is developed by Solomon Asress [34]. 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. Moreover, he has also used relatively better Amharic tagsets and large size corpus than the previous works for the Amharic Language. Mesfin [26] has used 23 taggsets and one page long text while Solomon has used 30 taggsets and 210,000 words of text corpus. He 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 ($P(T_i|W_i)$) that can be calculated by dividing the occurrence of the number of appearances of the W_i and T_i by the number of occurrences of W_i 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 [34] has stated the tagging of POS to be tackled in two levels: at sentence level and at word level. He has stated it as follows:

> *In the case of word level tagging, the problem can be posed as a classification problem. Whereas, in the case of sentence level tagging, a series of tags corresponding to the sequence of words in the sentence need to be found. In this case, the context provided by the whole sentence influences the tag assignment. In this thesis work, the Multi Layer Perception-Tagger with back propagation algorithm addresses the classification problem posed by the word level POS tagging. The input to this network is the set of words that fall into a window of pre-specified size centered on the target word to be tagged. The output of the network is the corresponding tag for the target word. The network learns the word-tag mapping as a complex function; F (target word, context) = tag. The context refers to the set of words in the immediate neighborhood of the target word, i.e., it performs tagging using a fixed length context.*

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.

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 earlier work for Amharic [26] is developed by Mesfin Getachew 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 facilitate 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 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.

A hybrid Part of Speech Tagger for another Semitic language called Arabic, was proposed by Tlili-Guiassa Yamina [48] in 2005. This work combines rule based and Memory-Based Learning.

> *The Memory based learning contains two components: The first one is a learning component in which memory storage is done without abstraction or restructure. The second one is a performance component that does similarity-based classification.*

The basic idea of the researcher, in the proposed part of speech tagger, is to apply rules to determine the tag type of each word in an Arabic sentence and then to refer to memory based in order to check whether it is an exceptional case or not. When the rule is applied to a predicted tag T_i for a specific Arabic word W_i in the sentence, T_i is compared with the correct tag in the training corpus. If the predicted tag T_i is equal with the tag in the training corpus it is correct, as a result the tag of the word W_i will be T_i. Otherwise, it is considered as an exception and the type of error is determined according to the correct tag in the training corpus and the wrong predicted tag. For each rule executed, the number of exceptional cases is stored in a library file. Putting these altogether the researcher has stated as follows:

Firstly, the rules are applied to determine the tag, and it is checked as an exceptional case of rules. Secondly, it is presented to memory based reasoning, its similarity to all examples in memory is computed using a similarity metric, and the tag is determined again.

The researcher has decided to use rule based as there are several signs in the Arabic Language like affix that indicate the category of word and hence such signs have a significant impact on the Part of Speech Tagging. The Memory based learning is important to determine the most frequent tag of a given word. While finding the most frequent tag, the context and form features are looked up for each word in the text. Moreover information about the surrounding words is used. The researcher has used K-Nearest Neighbour (K-NN) algorithm to find the most frequent tag for a sequence of words in a text.

The researcher has collected data from educational books as well as Qur"anic texts that was tagged using small tagset and retagged with more detailed tagset. They have used 131+ tags that are proper to Arabic Language. The 131 tags were taken from the previous similar work in [33]. The researcher has conducted an experiment on the extracted data on three different systems namely the rule based, Memory based learning and hybrid (rule based and memory based learning); hence he has got a result that gives the following advantages of the hybrid over the individual ones:

- Make the tagging process more robust
- Involves the disambiguation of a word on the basis of information coming from both sources which resulted in performance increment over the individual taggers

A transformational error driven learning approach was used in the work of [27] by mohammed-hussen abubeker in Addis Ababa University in 2010 for Afaan Oromo language. In this work, the researcher has adapted the Brill Transformational error driven learning with some modifications on the tagger template. This approach can be considered as a hybrid (rule based + stochastic) as it learns rules automatically during the training that makes it rule based and it uses supervised learning method (it uses part of speech tagged training corpus) to get the statistical property of words like lexical probability and contextual probability from the training corpus that

makes it stochastic approach. The researcher has used 233 sentences (1708 distinct words) of Afaan Oromo language which he has divided it into training set and testing set. He has used 18 tagsets to tag the 233 sentences. He has conducted different experiments to test the performance of the tagger for the original and modified Brill tagger. Accordingly, he has got a better performance for the modified Brill tagger to be 80.08% accurate. The performance of the original Brill tagger on Afaan Oromo text was found to be 77.64%. The performance increment of the modified tagger, as the researcher has stated, is due to the adjustment done for the Afaan Oromo learning template.

Another earlier work on Afaan Oromo language part of speech tagger was done by Getachew Mamo in Addis Ababa University in 2009 [13]. In this work, the researcher has used one of the known models, which is the generalization of the stochastic approaches, HMM for tagging Afaan Oromo Texts. He has collected 159 Afaan Oromo sentences (with 1621 distinct words) from different sources and he has used 17 tagsets to annotate these sentences. He has divided these sentences as training set and test set. The HMM based Afaan Oromo part of speech tagger was trained on the training set in order to compute and store the lexical and contextual probabilities of words in the training. The tagger then takes untagged Afaan Oromo text as an input and tokenizes the sentences into words before actually assigns the part of speech tags sequence. After this, each token in the sentence is assigned with a correct part of speech tag sequence that is done using unigram and bigram models of the Viterbi algorithm by taking the knowledge from lexical and contextual probabilities gained during the training session. The researcher has tested the performance of the tagger by conducting experiments and as a result he has got an accuracy of 87.58% and 91.97% for the unigram and bigram models respectively.

A hybrid Approach, by applying combination of the Rule based and statistical approaches has been introduced for Turkish Language part of speech tagging [45]. The researchers in this work, [45] 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 analyser with stochastic methods in order to improve the accuracy of the system as morphological analyser 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 analyser 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%.

Another work [8] which has used a Memory based learning approach that combines the attractive properties of rule based and stochastic approaches was done for Dutch Part of speech tagging. In this work [8], the researchers have stated that, in Memory-Based approach a set of example cases is kept in memory in which each case consists of a word (or lexicon representation for the word) with preceding and following context and the corresponding category for that word in that context. Tagging of a new sentence is performed by selecting for each word in the sentence the most similar case in memory, and extrapolating the category of the word from these nearest neighbours. This paper stipulated out that the construction of POS tagger for a corpus is achieved as follows: Given an annotated corpus, three data structures are automatically extracted:

- A lexicon (associating words to possible tags)

- A case base for known words (words occurring in the lexicon)
- A Case base for unknown words.

During tagging each word in the text to be tagged is searched in the lexicon and if it is found, the word is disambiguated from the most similar cases in the known words case base. If the word is not found in the lexicon, its lexical representation will be compared on the basis of its form, its context and the resulting pattern is disambiguated using extrapolation from the most similar cases in the unknown words case base. Experimental results for this study shows that the researchers have used 27,651 Dutch sentences (roughly 610,806 words) as a corpus which is divided into the training set and testing set. They have used 5763 sentences for the testing set and the rest sentences for the training set. In their experiment they have restricted the tag sets to be 12 and accordingly they have got an accuracy of 97.1% for known words and 71.6% for unknown words.

Eric Brill [4, 10] proposed another approach for part of speech tagging of an English text: transformation based-error driven learning approach to part of speech tagger. In fact, his original work was done using rule based approach. He has proposed the system to be able to automatically recognize and remedy its weaknesses, thereby incrementally improving its performance [4]. The basic idea of the tagger is to simply tag each word by its most likely tag estimated from a large tagged text corpus regardless of its context [4] and an ordered list of transformations is then learned to improve tagging accuracy based on context, that is, in order to change from one tag T_i to another tag T_j based on some transformation rules and/or grammars. The rules and/or grammars are induced directly from the training corpus without human intervention or expert knowledge [4]. The only components necessary for the tagger are a manually and correctly annotated corpus - the training corpus - which serves as input to the tagger and in addition to the templates. The tagger did not take any hand crafted rules by linguistic professionals rather it was made to learn rules from the input corpus for training. Moreover the training corpus was also used for driving the lexical and contextual information that are important in assigning the most appropriate tag for a word.

There had been many possible transformations in the process of tagging and they have tried all these transformations and the transformation that resulted in the greatest error reduction is chosen. It is mentioned already that the tagger is based on transformation-based error driven learning that implies through out the tagging process the system tries to learn new rules and/or transformations. Learning will be stoped when no transformations can be found whose application reduces errors beyond some pre-specified threshold. The processing of the tagger was done from left to right and for each transformation application in the process all triggering environments are found from the corpus which are helpful variables to trigger and thereby carrying out the transformation application [10].

The Tagger was tested using 1.1 million words of the Penn Treebank Tagged Wall Street Journal Corpus. Of these, 950,000 (350,000 words were used to learn rules for tagging unknown words and 600,000 words were used to learn contextual rules) words were used for training and 150,000 words were used for testing. Accordingly the system has learned 243 useful rules for unknown words and 447 contextual tagging rules. Finally, the researcher has got a result that can be mentioned as satisfactory. The overall tagging accuracy became 96.6% on the test corpus.

2.3 Summary

Recently Part of Speech tagging, a technique of assigning each word of a written text with an appropriate parts of speech tag, has become a hot research area as it is significant for language processing by giving large information about words in a written text. Moreover it is also important in Text to Speech (TTS), Information Retrieval (IR), shallow parsing, Information Extraction (IE), linguistic research for corpora and also as an intermediate step for higher level NLP tasks such as parsing, semantics, translation, and many more. As a result many POS tagging researches have been done and different approaches have been used for POS tagging, where the well-known ones are rule-based, stochastic, ANN, and Hybrid Approaches.

The Stochastic, ANN and rule based approaches for tackling part of speech tagging have got their own disadvantages and advantages. Hence, there have been many researches done by combining the rule based and stochastic or ANN and rule based so that it would be possible to take the advantages from both approaches thereby improving the performance of the tagger.

It is also indicated that the smaller the dataset used for training and testing the part of speech tagger, the more difficult it is to accept the result obtained as it may not reflect the reality. The larger the dataset, however, shows the better representation of the reality and hence it is easy to accept the result obtained by the model. Moreover, as far as the researchers knowledge is concerned, there is no part of speech tagger developed for Tigrigna and hence this is the first attempt to develop POS tagger fir Tigrigna language.

CHAPTER THREE

TIGRIGNA LANGUAGE AND TAGSET PREPARATION

3.1 Overview

Tigrigna is the designated official language of Tigray region in Ethiopia and is the language most widely spoken in this region. It is also one of the national languages of Eritrea. And hence, it is a de facto official language in Eritrea and Tigray. Besides being a lingua franca, it is a medium of instruction of primary schools in the aforementioned areas and other Tigrigna speaking people. It has become the primary language of oral and written communication in Eritrea and Tigray. In some institutions of Tigray like Tigray Teachers training institution, all the subjects are taught in Tigrigna for elementary teachers. It is also thought as a subject in the junior and senior secondary schools of Eritrea and Tigray [36, 42]. Nowadays, many translation works are being done, and so many science, mathematics and social science terms are being coined into Tigrigna [36].

Tigrigna belongs to the Semitic language family where it shows the characteristic features of a Semitic language [36, 42]. It uses a special writing system called the Ge"ez or Ethiopic alphabet. The normal syllable in Tigrigna is considered to be a consonant followed by a vowel. If a consonant ends a syllable, the sixth, neutral vowel is used with it. Most consonants are written in seven different forms corresponding to the seven different vowels [36, 41, 42].

3.2 Tigrigna Sentence Structure

Language is a structured system of communication. Segments and features of Tigrigna form syllables which in turn form words. Any Tigrigna sentence, as in any other language, can be divided into two largest immediate constituents namely subject and predicate [36]. Let us look at the following examples. The examples are written in Tigrigna alphabets followed by the transliterated form of the alphabets to their corresponding Latin characters. The transliteration details of the Tigrigna characters are given in appendix B which is taken from the work of [12].

A) ሓውካ መፂኡ። /Hawka me^Si^u.
Your Brother came (has come).

B) ሓውካ ትማሊ ካብ ሸረ መፂኡ። /Hawka tmali kab xre me^Si^u.
Your brother came (has come) from shire yesterday.

C) እቲ መፅሓፍ ኣብ ልዕሊ ዓራት ግበሮ። /^ti me^SHaf ^ab l`li `arat gbero.
Put the book on the bed.

In the first two sentences ሓውካ /Hawka (your brother) functions as a subject, while the rest parts of the sentences are used as a predicate መፂኡ/me^Si^u (came (has come)) and ትማሊ ካብ ሽረ መፂኡ/tmali kab xre me^Si^u (came (has come) from shire yesterday). The subject can be thought as a noun phrase which is used to mention some objects or persons and the predicate is used to say something true or false about the subject [36, 42]. These two elements of sentences are known as noun phrase (NP) and verb phrase (VP) which are headed as noun and verb respectively. In the third sentence, the subject is entrenched with the verb ግበሮ/gbero which is you 3rd person singular that acts as the noun phrase (NP) and the sentence ,,እቲ መፅሓፍ ኣብ ልዕሊ ዓራት ግበሮ/^ti me^SHaf ^ab l`li `arat gbero'' acts as verb phrase (VP) which in turn is divided into noun phrase and verb phrase. The parse trees for the above three sentences is given in figure 3.1.

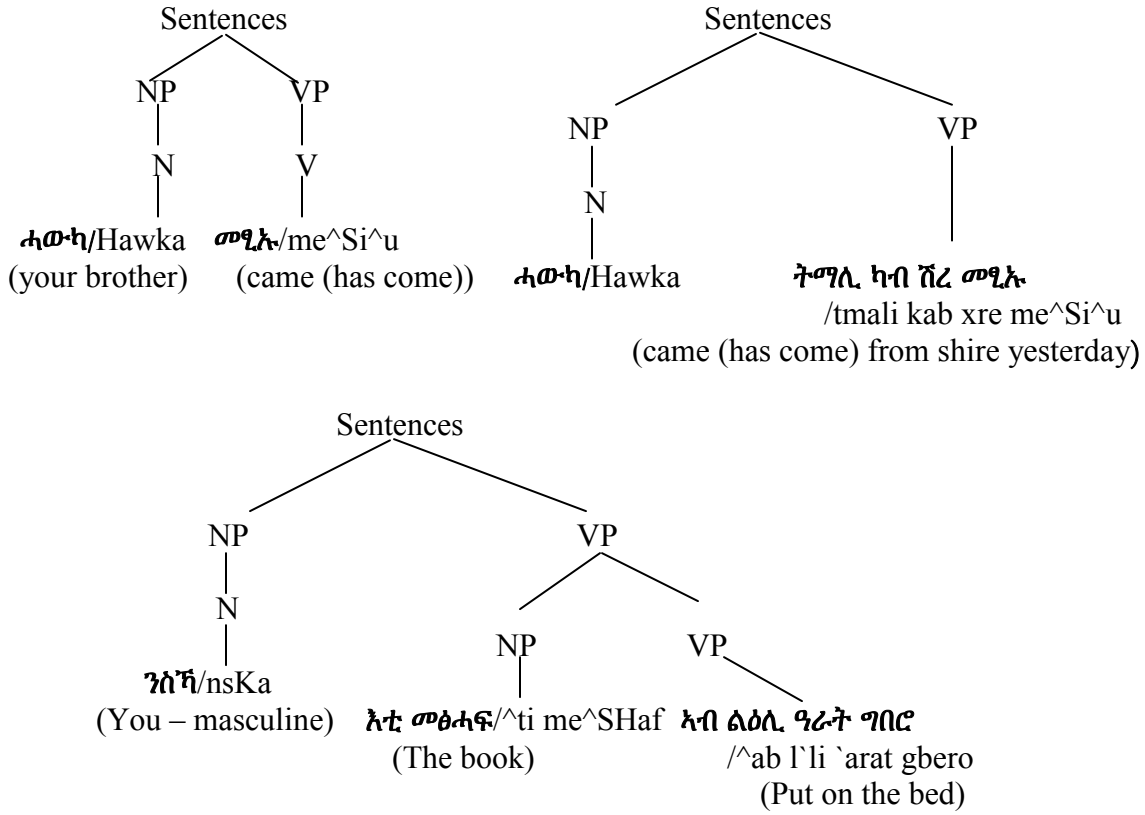


Figure 3.1 Tigrigna sentence structure [36, 42,50]

Sentences in Tigrigna can be classified as simple sentence and complex/compound sentences [36, 49, 50]. Simple sentences have only one verb while complex sentences have more than one

verb. The three aforementioned sentences are all simple sentences while the following sentence is a complex.

እቲ ጤል ዝሰረቐ ሰብኣይ ተኣሲሩ።/^ti TEl zsereQe seb^ay te^asiru.(The man who stole a goat is jailed.)

This sentence contains two verbs: ዝሰረቐ/zsereQe (stole) and ተኣሲሩ/te^asiru (jailed). In other words, it contains a subordinate and main clause which have one verb each. A subordinate clause is a clause which is dependent to another independent clause called the main clause [36, 49, 50]. In this case the main clause is እቲ ሰብኣይ ተኣሲሩ /^ti seb^ay te^asiru (The man is jailed) ad the subordinate clause is ጤል ዝሰረቐ /TEl zsereQe (who stole goat).

3.3 Word Classification of Tigrigna

3.3.1 The need for categorizing Words

As it is stipulated out in the previous sections, a sentence is composed of two components and these two components in turn consist of words. Therefore, it is possible to conclude that words are the basic components of every sentence. The meaning of a sentence is analyzed from the meaning of individual words and the way they are arranged. This shows that words are rarely used alone. Words more often work together in small groups which together make up whole sentence that possess a single coherent meaning. Moreover, the same word can be used in different sentences and belong to a different word class category.

Three basic criteria are considered in order to categorize words in a language. They are: the meaning of the word, the form or shape of the word, and the position or the environment of the word in a sentence. These can be taken as the main criteria to determine the categories of a given word [34, 36, 49].

3.3.2 Tigrigna Word Classes

Words of Tigrigna in general can be classified into two: open and closed. Open classes and closed classes are so called because new members are always added to the former and are

unlimited in number; whereas, members of the closed classes are relatively fixed and few in number [36].

As far as the number and types of classes are concerned, languages differ from one another in closed than in open classes. Some languages may have a dozen or more closed classes whereas others may have extremely few [36]. Tigrigna word classes were studied by some researchers and are briefly discussed in this section.

The first attempt found about Tigrigna word classification is the PhD work by Tesfaye in 2002 [36]. He has classified Tigrigna words into two broad categories: open and closed in general and into eight classes in particular. The three classes, Noun, Verbs and Adjectives, are in the category of open classes and the rest five namely Pronouns, Determiners, Adverbs, Prepositions and Conjunctions, are in the category of closed classes. Interjections are words without syntactic functions. In this categorization, interjections are not considered as part-of-speech/word class.

The second attempt found in Tigrigna word classification is the work by Daniel [50]. He reduced the categorization of Tigrigna words into five. These are: preposition, noun, verb, adjective, and adverb. Pronouns and conjunctions are put under noun and preposition categories respectively. In this categorization, interjections, like Tesfaye‟s classifications, are not considered as part-of-speech/word class. A brief description of each of the classes in Tesfaye‟s and Daniel‟s classification is given in the following topics.

Tigrigna Noun Class

A noun is a word that is used for labeling things, such as a real thing (for example, cat), an imaginary thing (for example, ghost), an idea (for example, love), person name (for example, “Teklay”). Tigrigna nouns, like English, are words used to name or identify a class of things, people, places or ideas. They typically function as arguments, subjects, objects of transitive verbs or complements of prepositions.

Tigrigna Verb Class

Tigrigna has generally a subject, object, verb (SOV) word order. The Verb is a word that tells us the state of doing or being. Tigrigna verbs carry inflections of aspect and mood and hence are morphologically the most complex POS in Tigrigna. A lot of words with other POS are derived primarily from verbs. There are two major approaches to identify verbs from other word categories: syntactical and morphological approach. In the former case, verbs function as predicates in a simple sentence and they are found at the end of a sentence. In the later case, they reflect grammatical categories such as aspect, mood and agreement.

Tigrigna Adjective Class

Most languages appear to identify two open classes namely nouns and verbs. However Tigrigna has got an additional open class, the Adjective class [36]. The numbers of Tigrigna Adjectives are too many and their numbers increase from time to time [36].

Adjectives are words that describe or add extra information to a noun. Adjectives in Tigrigna usually precede the nouns that they modify or describe. Here is a simple example.

ነዋሕ ጓል ፈትየ።/newaH gWal fetye. (I loved tall girl.)

In this example, the adjective **ነዋሕ**/newaH (tall) precedes the **ጓል**/gWal (Girl) which it modifies. But this does not mean that a word is an adjective just because it precedes a noun. For instance, in the sentence **እታ ጓል ቆንጆ እያ።**/^ta gWal qonjo ^ya (The girl is beautiful.), the word **እታ**/^ta (the) precedes the noun **ጓል**/gWal (girl). Although the word **እታ**/^ta (the) functionally shares the feature of an adjective, modifier, it is a demonstrative pronoun.

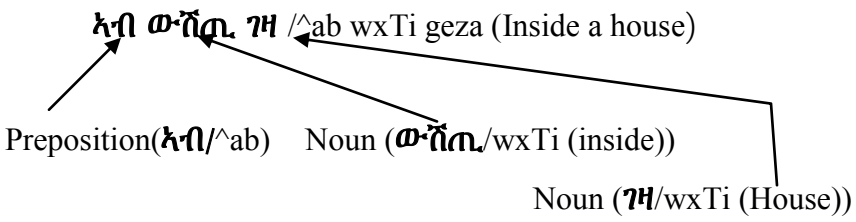
Tigrigna Pronoun Class

A pronoun is a word which is regarded as a subclass of a noun According to Daniel [50]. It can be taken as noun because it can function as a noun and can take the position of a noun. However pronouns are closed classes for two main reasons: first they are few in number and second the number of their members does not increase [36]. Examples of Tigrigna pronouns are: **ንሱ**/nsu (He), **ኣነ**/^ane (I) **ንስኻ**/nsKa (you (masculine)), **ንስኺ**/nsKi (you (feminine)), **ንሳ**/nsa (she) etc.

Tigrigna Preposition Class

Prepositions are small set of words, which will have meanings only when they are attached or used together with other words such as nouns, verbs, pronouns and adjectives. They can express relationship between person, thing, or event etc and another. Tigrigna prepositions have the following central property:

- They take nouns or adjectives as their complements: e.g. **ብ**-ሃንደበት/**b**-handebet (Suddenly)
- They can stand alone as a separate word: e.g. **ናብ ቤት ትምህርቲ** /**nab** bEt tmhrti (To School)



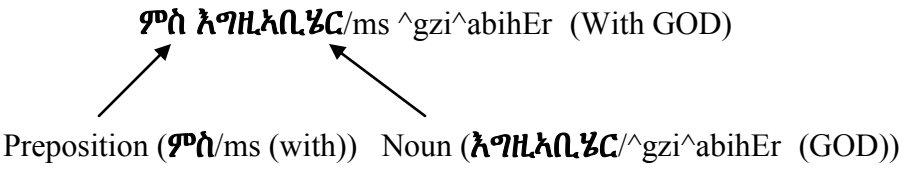


Figure 3.2 prepositions as a separate word [36, 49, 50]

- They are not inflected for gender, person, number etc.

Tigrigna Adverb Class

An adverb is a word that modifies a verb, adjective, sentences or clauses and other adverbs. In many languages adverbs are classified as open classes; however, Tigrigna‟s adverbs are classified as closed classes [36].

Modifiers of verbs or verb phrases usually express time, place, manner etc. Modifiers of adjectives and adverbs commonly express degree while adverbs functioning as sentence modifiers usually express the speakers‟ attitude regarding the event spoken.

e.g. **ትማሊ ቀልጢፉ መፂኡ።**/tmali qelTifu me^Si^u. (Yesterday he came quickly.)

Adverb of time adverb of manner verb

ንየማን ገፅ ኪድ።/nyeman ge^S kid. (Go to the right side)

Adverb of place Noun verb

ትማሊ ኣዝዩ ቀልጢፉ መፂኡ።/tmali ^azyu qelTifu me^Si^u. (yesterday he came very quickly.)

Adverb modifying the adverb **ቀልጢፉ**/qelTifu (quickly)

ብኣጋጣሚ, ኣብታ መኪና ዝነበሩ ሰባት ድሒኖም።/b^agaTami ^abta mekina zneberu sebat dHinom. (Fortunately, the people in the car were saved)

A sentence modifier

Figure 3.3 different adverbs of Tigrigna [36,49,50]

Tigrigna Conjunction Class

Conjunctions are words that link words, phrases and clauses to create larger grammatical units. Conjunctions were treated as a separate category [36, 49]. However, conjunctions are treated also as a subclass of prepositions [50]. Conjunctions can be distinguished as two categories: coordinating or subordinating. They coordinate words, phrases, clauses and sentences.

e.g**. ሰብኣይ - ን ሰበይት - ን**/seb^ay-**n** sebeyt-**n** (husband and wife)

Most of the items that function as prepositions in Tigrigna can also function as conjunctions. Moreover the coordinating conjunctions such as - **ን**/n (and), **ወይ**/wey (or) can form structural units with the nominal and prepositional phrases they precede or follow.

Determiner in Tigrigna

These are the structures used as demonstratives or function to indicate definiteness. They are also called demonstrative pronouns.

e.g. **እቲ ሰብኣይ**/^ti seb^ay (that man) **እዛ ጓል**/^za gWal (this Girl)

Interjection in Tigrigna

Interjections are words that function to suggest sudden, often unexpected, emotion. Tigrigna has many words or phrases used to express such emotions as sudden surprise, pleasure, annoyance and so on. Such words are called interjections. These Tigrigna interjections can stand-alone by themselves outside a sentence or can appear anywhere in a sentence.

e.g. **ዓሸ!**/`ax! (wow!)

ዓሸ! ንፉዕ ዝወደይ/`ax nfu` zwedey (wow! you (my son) excellent!)

ዋይ ኣነ!/way ^ane (oops!)

Numerals in Tigrigna

In Tigrigna there are words representing numbers that are called Numerals. The numerals can be classified as ordinal numbers and cardinal numbers. The cardinal numbers are numbers like **ሓደ**/Hade (one), **ክልተ**/klte (two), **ሰለስተ**/seleste (three), **ዓሰርተ**/`aserte (ten), **ዓሰርተ ሓደ**/`aserte Hade (eleven) and the corresponding ordinal numbers are **ቀዳማይ**/qedamay (first) **ካልኣይ**/kal^ay (second) **ሳልሳይ**/salsay (third), **ዓስራይ**/`asray (tenth), **መበል ዓሰርተ ሓደ**/mebel `aserte Hade (eleventh) etc.

There are also special numerals in Tigrigna that correspond to the English „half", „quarter" etc. Examples of these include **ፍርቂ**/frqi (half), **ርብዒ**/rb`I (quarter") and **ሲሶ** /siso (one third).

.

3.4 Tigrigna tags and Tagsets

In the previous sections, the broad categories of Tigrigna word classes are explained from the works of [36, 50]. In this section, the actual tags used in this thesis work are discussed. Tags are the labels used for adding more information concerning the lexical category of each word in a sentence and tagsets are the collection of the tags used for developing the Tigrigna part of speech tagger.

As far as the researchers‟ knowledge is concerned, there is no ready made tagset, unlike that of the Amharic language, for Tigrigna that researchers can make use of it. This implies that identifying and developing tagsets for this thesis work is mentioned to be crucial. As a result, the researcher has made continues discussion with Tigrigna Language professional like Abreha Girmay, Daniel Teklu, Mulu Hailesslassie, and other people who have knowledge of the language. Abreha Girmay and Mulu Hailesslasie are Tigrigna teachers who are currently pursuing their Masters degree in linguistics in Addis Ababa University and Bahrdar University Respectively. Daniel Teklu has finished his master‟s degree and now he is pursuing his doctorate degree in Addis Ababa University. Since the preparation of tagsets for a language is tedious and time consuming task, which needs in fact, human experts in the language of interest, the researchers have developed, after continuous discussions with the aforementioned professionals, broad categories of tagsets. Moreover, the work on Amharic [14] is used as a reference due to the similarities of the languages as both are Semitic derived from the same ancestor (Geez). This work [14] was done for the annotation of Amharic text news collected form Walta Information Center. The researchers [14] have identified 30 Tagsets for the annotation of the text news that Solomon Asres has used for his thesis work [34].

The tagsets that are discussed below are classified as a basic class and subclasses of the basic class where noun, pronoun, verb, adjective, preposition, conjunction, adverb, interjection are considered to be the basic classes. In addition, numeral and punctuation are also included as basic classes in the process of identifying the tagsets. The concept hierarchy of the total tagset that are identified is represented in figure 3.4 in terms of the basic classes and their respective subclasses.

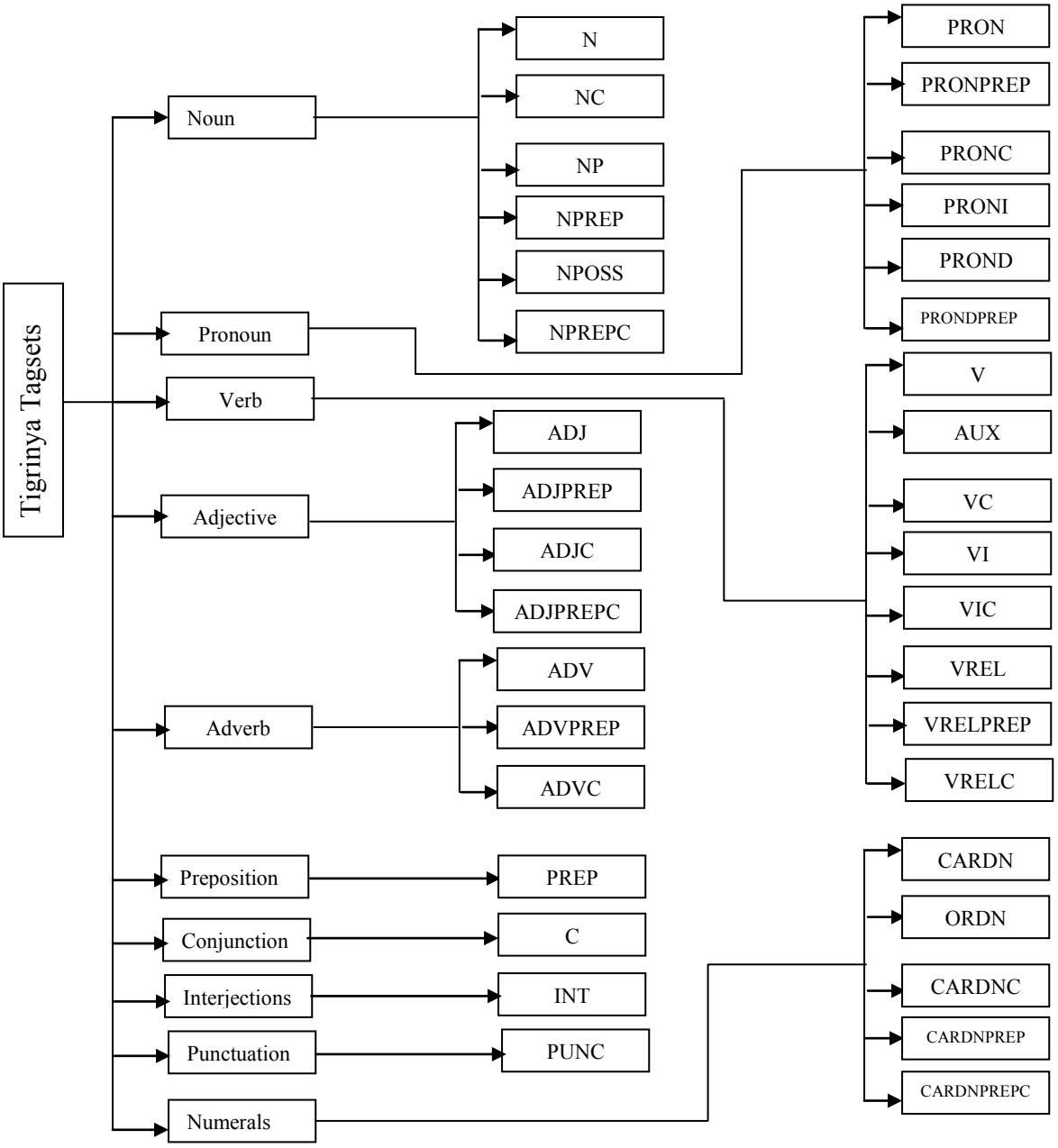


Figure 3.4 tagset concept hierarchy

The Noun Basic Class and its subclasses

Nouns as described in the above sections represent things in the real word. In Tigrigna, as it is in Amharic, English and Afaan Oromo, nouns have different attributes like gender, number, case and definiteness which can be of proper nouns, common nouns or abstract nouns etc. But in this

study, the researchers did not take these entire attributes into consideration except for the proper nouns because the due consideration of all attributes makes the tagsets detail and complex. The identified subclasses of nouns are Proper Noun, Verbal Noun, Verbal Noun with prepositions, Verbal Noun attached with conjunctions, Noun including both preposition and conjunction, and verbal noun with possession indicator. The last four subclasses were identified as the experts were doing on a word level tagging. Due to the nature of the language, a word may be formed from two or more words concatenated or assimilated into one word. It is also not possible to split such word into its constituents. For example, a preposition may get attached with a noun. These subclasses are explained in the following examples.

- Nouns that represent the name of a specific person, place, thing, institutions, organizations, religions, etc are considered to be proper nouns subclass and are tagged by NP. Example ኢትዮጵያ/Ethiopia, ኣበበ/Abebe, ኣባያታ/Abayata etc are tagged as NP.
- Noun prefixed with preposition and when the preposition cannot be separated from the noun is considered to be the noun preposition subclass and tagged as NPREP. Example: **ብ**-ባንኪ/**b**-banki (by bank)
- Noun affixed with conjunction and when the conjunction cannot be separated from the noun, it is considered as noun conjunction subclass which is tagged as NC. Example: ፓርትታት-**ን** ውልቀ-**ን**/Parttat**n** wlqe**n** (Parties and individuals)
- Nouns that are prefixed and affixed with both prepositions and conjunctions which can not be separated from the noun are classified as noun preposition with conjunction subclass and tagged with NPREPC.
  Example: **ብ**-ምውላዕ-**ን**/**b**-mwla`-**n** (by lighting and)
- Sometimes, Nouns may be attached with possession indicator which cannot be detached and hence classified as Noun possession that can be tagged as NPOSS. Example ክልል-**ና**/kll-**na** (our region)
- Other forms of nouns such as common nouns, concrete nouns, abstract nouns etc that can be classified under the above subclasses are tagged as N. examples are: ምግቢ/mgbi (food, Common noun), ኣራዊት/^arawit (Concrete noun), ቁልዕነት/qul`net (Abstract noun)

Pronoun Basic class and its subclasses

According to Daniel [50] pronouns can be taken as a sub class of noun. But coming to the preparation of tagsets for Tigrigna, the researchers have taken them as a basic class as they frequently occur in documents and hence it is better to treat them independently. Pronouns, in the work of amharic corpus preparation [14], are also treated independently rather than subclassifying them under the basic class of nouns for the aformentioned reasons. The following subclasses of pronouns are identified in this research work:

- Pronouns attached with Prepositions which can not be separated are classified as pronoun prepostions and are tagged as PRONPREP. Example: **ን-ባዕልና**/**n**ba`lna (for us)
- Pronouns attached with conjunctions which can not be separated are classified as pronoun conjunctions and are tagged as PRONC. Example: **ኣነ-ን ንስኻ-ን**/^ane**n** nsKa**n** (me and you)
- Pronouns that can point or identify a noun and/or prounoun are classified as demonstrative pronouns and are tagged as PROND. Example: **እቲ**/^ti (the)
- The demonstrative pronoun can be attched with preposions and classified as demostrative pronoun with prepostions and it is tagged as PRONDPREP. Example. **ኣብቲ**/^abti (on the)
- Pronouns that can be used to ask questions are classified as interrogative Pronouns and are tagged with PRONI. Examples: **እንታይ**/^ntay (what)
- Pronouns that can not be classifed in one of the above classfications are tagged as PRON.

Verb basic class and its subclasses

Verbs are possibly the most important part of any text almost in any language [36, 48, 49, 50]. A sentence without a verb cannot give a complete meaning. The researcher identified one general tag and seven tagsets for seven subclasses of verb in this work. The subclasses are explained with examples as follows.

- Auxillary verbs are one sub categories which are tagged by AUX. AUX is used to show such verbs in all forms without any distinction like number, gender, tense etc. Example: **እዮም** /^yom
- Verbs which are attached with conjunction and can not be separated are tagged as VC. Example: **በሊዑ-ን ሰትዩ-ን**/beli`u-**n** setyu-**n** (ate and drank)

- Verbs that show infinitive like that of the verb to be in english are tagged as VI. Example: **ንምርግጋፅ**/nmrgga^S (to assure)
- An infinitive verb attached with conjunction is tagged as VIC. Example: **ንምቅላል-ን** /nmqlal-**n** (to simplify and)
- Relative verbs of Tigrigna are verbs that are prefixed most of the time with **ዝ** or **ዘ** which indicate the subject of the sentence and are tagged with VREL. Example: **ዝጠመተ**/zTemete (focused)
- Relative verbs which are attached with prefix are tagged with VRELPREP. Example: **ን-ዝተሓቱ** /nzteHatu (for less capable of doing something)
- Relative verbs attached with conjunctions are tagged as VRELC. Example: **ዝተሳተፍሉ-ን** /ztesateflu-**n** (participated and)
- All other verbs which can not be classified in the above subclasses are tagged by the genral tag for verbs V.

Adjective basic class and its subclasses

As it is mentioned above, an adjective is another word category that is meant to add extra information to nouns. In the identification of tagsets for Tigrigna part of speech tagger, one general tag and the following three sub categories are stipulated out.

- Adjectives attached with prepostion that can not be separted are tagged with ADJPREP. Example: **ን-ሰለማዊ** /**n**-selemawi (for peaceful and)
- Adjectives attached with conjunction that can not be separted are tagged with ADJC. Example: **ማሕበራው-ን ቁጠባው-ን** /maHberaw-**n** quTebaw-**n** (social and economical)
- Adjectives that are attached with both prepostions and conjunctions are tagged as ADJPREPC. Example: **ን-ማሕበራው-ን** / **n**-maHberaw-**n** (for social and)
- Any other adjective which does not belong to these subcategories is tagged as a general tag ADJ.

Adverb Basic class and its subclasses

Tigrigna adverbs are words that modify a verb, adjective, clause or other adverbs. Though adverbs can be classified as adverbs of time, manner, place etc, these all are considered as a general adverb that can be tagged as the general adverb tag (ADV) and the following two categories.

- Adverbs attached with preposition that cannot be detached are tagged as ADVPREP. Example: ብ-ከመይ/b-kemey (in what manner)
- Adverbs attached with conjunction that cannot be detached are tagged as ADVC. Example: ብመጠን ቀቻታ-ን /bmeTenqeQta-n (warning and)

Prepositions basic class

Prepositions alone will not convey any meaning unless they are attached or used with nouns and other basic classes. Some of the prepositions are attached to the word, may be noun or something else, in such a way that they cannot be separated as this work focuses on the surface form of the words, and some of them can be found alone used with other words separately. If the preposition is separated from the word being used it is tagged as PREP. Example: ኣብ/ ^ab (on)

Conjunction basic class

Like that of prepositions, conjunctions are words that are either attached or used with some words for the purpose of linking in order to create coherent grammatical units from linguistics point of view. If the conjunctions are used with words such as nouns, adjectives as a separate word they are tagged as C. example: ድሕሪ/dHri (after)

Tigrigna Numerals

Tigrigna numerals like that of Amharic, English, and Afaan Oromo can be cardinal numbers and ordinal numbers which are tagged as CARDN and ORDN respectively. In Tigrigna the numerals can get attached with prepositions and conjunctions that may not be separated. If the cardinal numbers are attached with prepositions and conjunctions they are tagged as CARDNPREP and CARDC. Moreover if there are cardinal numbers that are attached with both prepositions and conjunctions they are tagged as CARDNPREPC.

Interjections

Interjections are words or phrases used to express emotions which can stand by themselves or can appear anywhere in a sentence. They are tagged as INT.

Punctuation Marks

All Tigrigna punctuation marks like **:-, :, ::, ?** and **!** are assigned the tag PUNC.

3.5 Summary

Though Tigrigna is the de facto language in Tigray region and Eritrea, there is no well agreed Tigrigna language word classes‟ classification. Different professionals have classified the word classes into eight according to Tesfaye [36], or into five according to Daniel [50] by giving their own reason. In fact the classification can vary depending on the objective of the study being made. For the development of the Tigrigna Part of speech tagger, the researchers have assessed the different word classes of the language from different sources [36,49,50] and all essential word categories of the language are included. The Tigrigna tagsets which are important in labeling Tigrigna word sequences from the word categories are extracted. Hence, 36 tagsets are set which are summarized in table 3.1 with examples for each tagset.

Table 3.1 Tigrigna Tagsets

NO	Basic category/tag	Derived category/tag	Description	Example
1.	NOUN	N	Noun	ምግቢ/mgbi
2.		NP	Proper noun	ተክላይ/Teklay, ኣፍሪካ/Africa, ተክዘ/Tekeze
3.		NC	Noun + conjunction	ፓርትታትን ውልቀን/Parttatn wlqen
4.		NPREP	Preposition + noun	ብባንኪ/bbanki
5.		NPREPC	Preposition +noun+conjunction	ብምውላዕን/bmwla`n
6.		NPOSS	Noun + possession	ክልልና/kllna
7.	PRONOUN	PRON	pronoun	ኣነ/^ane, ንስኻ/ nsKa
8.		PRONPREP	Preposition + pronoun	ንባዕልና/nba`lna
9.		PRONC	Pronoun + conjunction	ኣነን ንስኻን/^anen nsKan
10.		PRONI	Infinitive pronoun	እንታይ/^ntay
11.		PROND	Demonstrative pronoun	እቲ/^ti
12.		PRONDPREP	Preposition + demonstrative pronoun	ኣብቲ/^abti
13.	VERB	V	verb	ጠመተ/Temete
14.		AUX	auxilary	እዮ /^yu
15.		VC	Verb + conjunction	በሊዑን ሰትዩን/beli`un setyun
16.		VI	Infinitive verb	ንምርግጋፅ/nmrgga^S
17.		VIC	Infinitive verb + conjunction	ንምቅላልን /nmqlaln
18.		VREL	Relative verb	ዝጠመተ/zTemete
19.		VRELPREP	Preposition + relative verb	ንዝተሓቱ /nzteHatu
20.		VRELC	Relative verb + conjunction	ዝተሳተፍሉን /ztesateflun
21.	ADJECTIVE	ADJ	adjective	ማሕበራዊ /maHberawi
22.		ADJC	Adjective _ conjunction	ማሕበራውን /maHberawn
23.		ADJPREP	Preposition + adjective	ንሰለማዊ /nselemawi
24.		ADJPREPC	Preposition + adjective + conjunction	ንማሕበራውን / nmaHberawn
25.	ADVERB	ADV	adverb	ከመይ /kemey
26.		ADVPREP	Preposition + adverb	ብከመይ/ bkemey
27.		ADVC	Adverb + conjunction	ብመጠን ቀቐታን /bmeTenqeQtan
28.	PREPOSTION	PREP	prepostion	ኣብ/ ^ab
29.	CONJUNCTION	C	conjunction	ድሕሪ / dHri
30.	NUMERALS	CARDN	Cardinal number	ሓደ/Hade, ክልተ/klte, ሰለስተ/seleste
31.		ORDN	Ordinal number	ቀዳማይ/qedamay, ካልኣይ/kal^ay, ሳልሳይ/salsay
32.		CARDNC	CARDN + conjunction	ክልተን ሰለስተን/klten selesten
33.		CARDNPREP	Prep + CARDN	ንክልተ/nklte
34.		CARDNPREPC	Prep + CARDN + C	ንክልተን /nklten
35.	PUNCTUATION	PUNC	Punctuation	:-, :, ::, ?, !
36.	INTERJECTION	INT	Interjection	ዓ ሽ ! /`ax!

CHAPTER FOUR
DESIGN OF TIGRIGNA POS TAGGER

4.1 Introduction

Assigning grammatical categories to words in a text is an important component of a natural language processing (NLP) system. Text collection tagged with Part of speech (POS) information are often used as a prerequisite for more complex NLP applications such as information extraction, syntactic parsing, machine translation or semantic field annotation etc. Tigrigna POS tagging is a method of assigning a specific Tigrigna part of speech tag to each word in a Tigrigna sentence to disambiguate the function of that word in the specific context. In this chapter, a detail description of design issues and techniques of the Tigrigna POS tagger are dealt. Moreover, the design of HMM, Rule based, and Hybrid taggers is discussed.

4.2 Approaches and Techniques

As it is mentioned earlier, Tigrigna POS tagging is a process of labelling a word in a given Tigrigna sentence with its correct part of speech category based on its context in the sentence. The process of tagging takes a sentence as input, assigns a POS tag to the word or to each word in a sentence or in a text collection, and produces the tagged text as output. The following can be taken as an example:

Input sentence**:** **ኣብ ዓራት ኮፍ ኢሉ።** /^ab `arat kof ^ilu. (He sat on the bed.)
Output sentence: **ኣብ**/PREP **ዓራት**/N **ኮፍ**/V **ኢሉ**/V **።**/PUNC

The input sentence is tagged as preposition, noun, verb, verb and punctuation respectively. The symbols PREP, N, V and PUNC are called Tagsets that are explained already in sections 3.4 and 3.5 of chapter three.

In the field of computational linguistics, the problem of labelling words in a given sentence with the correct parts of speech can be tackled using different techniques. Some of the notable ones are the statistical, rule based (knowledge based) and a combination of these two.

Rule based tagger also called knowledge based tagger needs rules to perform POS Tagging. It can often correctly analyse complex and long structures, but they are generally unable to provide tags for constructions that have not been recognized [31]. Statistical taggers use the probabilistic algorithms to disambiguate words in a sentence but they need to be trained in a corpus. Another technique for tackling the problem of POS tagging is combining the two techniques: rule based and statistical approaches called the hybrid approach.

From the definitions of the aforementioned techniques, it is possible to infer that they have an implication on the design process and development of the Tigrigna POS tagger system. Therefore, the design and development of this thesis work is based on the combination of statistical and rule based approaches in order to see performance improvements. The statistical tagger is trained first by a pre-tagged corpus in order to get the distributional probability of each word in the training corpus. Then, an input sentence is given to this tagger that will label each word of the input sentence based on the distributional probability of the words in the training corpus. The rule based tagger is then made to learn rules on the same training corpus and store these rules so that they will be applied on the output of the statistical tagger. This implies that, since both approaches have their own pros and cons, it is possible to optimize the drawbacks of the two approaches if their hybrid form is used (statistical followed by rule based) thereby minimizing the limitations of both approaches [22].

4.3 Design Goals

The general goal of this research work is to developing the Tigrigna POS tagger thereby attaining better accuracy by combining two different approaches. Moreover, it is also a brainstorm for NLP researchers for Tigrigna language..

4.4 Designing Hidden Markov Model (HMM) Tagger

A Hidden Markov Model is a technique most widely used in the statistical approach that is important for finding the optimal part of speech tag sequence for a given word sequence. HMM allows to talk about both the observed hidden markov model events (the words in the input sentence and features derived from the words), and the hidden events (the part of speech tag

sequence which are casual factors in the probabilistic model [5, 6, 7, 13, 37]). This model uses the following formula to find the optimal tag sequence for a given word sequence:

$$t_{seq} = \underset{\vec{T}}{\arg\max}\ P(\vec{T} \mid \overrightarrow{W})$$

where t_{seq} is the sequence of tags to be found, $\vec{T}$ is $T_1, T_2, \ldots, T_n$ and $\vec{W}$ is $W_1, W_2, \ldots, W_m$. argmax implies $\vec{T}$ such that $P(\vec{T} \mid \overrightarrow{W})$ is maximized. This equation can be simplified as follows since it is difficult for calculation taken as it is:

$$t_{seq} = \underset{\vec{T}}{\arg\max}\ P(\vec{T} \mid \overrightarrow{W})$$

$$t_{seq} = \underset{\vec{T}}{\arg\max}\ \frac{P(\overrightarrow{W} \mid \vec{T})P(\vec{T})}{P(\overrightarrow{W})}$$

For each tag sequence, the word sequence is constant and hence it is possible to reduce the denominator for further simplification as follows:

$$t_{seq} = \underset{\vec{T}}{\arg\max}\ P(\overrightarrow{W} \mid \vec{T})P(\vec{T})$$

Assuming that the probability of the word is dependent only on its part of speech tag for the first part and considering bigram for the second part of the above equation, it is possible to approximate as follows:

$$P(\overrightarrow{W} \mid \vec{T}) = \prod_{i=1,j=1}^{n,m} (Wj \mid Ti)$$

$$P(\vec{T}) = \prod_{i=1}^{n} (Ti \mid Ti-1)$$

This implies: $t_{seq} = \underset{\vec{T}}{\arg\max}\ \prod_{i=1,j=1}^{n,m} P(Wj \mid Ti)P(Ti \mid Ti-1)$

This equation has two parts: the lexical model (that gives lexical probability) and contextual model (that gives contextual probability). Here a brief description of the lexical model and contextual model followed by the HMM model is given.

4.4.1. Lexical Model

The goal of the lexical model is to prepare lexicon and the lexical probability of each word for each tag in the training set. The lexical probability can be calculated with relative frequencies using the following formula:

$$P(Wi \mid Ti) = \frac{count\ of\ (Wi, Ti)}{count\ of\ (Ti)}$$

Where W_i and T_i are the i^{th} word in the input sentence and the i^{th} tag in the tagset respectively. The relative frequencies for the lexical model can be found by counting every word with a specific tag and divide it with the number of occurrences for this particular tag, which gives the conditional probability of the word given the tag [13, 17]. For example the Tigrigna word **ዓራት**/`arat (bed) will have two probabilities P(**ዓራት**|N) and P(**ዓራት**|ADJ) for the two part of speech categories N and ADJ, where N and ADJ denotes the classes noun and adjective respectively.

4.4.2. Contextual Model

This model helps the HMM tagger gather the context of words in the training corpus as lexical model only deals with the probability of the word given the tag. I.e. relying only on the lexical model may degrade the performance of the tagger and hence it is important to take context of words into consideration [13, 17, 34, 47]. The contextual model also called N-gram Model that considers the sequence of part of speech tags is aimed to calculate the transitional probability of tags. It is said in the works of [21, 27] that if part of speech tagging problem is to be solved and the training data contains small training corpora, it would be convenient to use the N-gram model to be bigram or trigram model that considers the previous one or two tags respectively. Since this research work has small training corpora in comparison with the corpora for Amharic [14] and brown corpus of English, a bigram model is selected. Therefore the contextual probability is found via tracking the previous one tag which can be calculated using relative frequencies by the following formula.

$$P(Ti \mid Ti-1) = \frac{\#(Ti, Ti-1)}{\#(Ti-1)}$$

The relative frequencies can be calculated by counting the frequency of T_i and T_{i-1} and divide it by the number of occurrences of T_{i-1} in the training corpus.

The above two models are the core models used in the HMM tagger for this thesis work. The HMM strives to find the optimal sequence of part of speech tags for a sequence of words in an input sentence using Viterbi algorithm. The tagger gets the lexical probability and contextual probability from the training corpus.

The overall architecture of the HMM tagger building process is described in figure 4.1

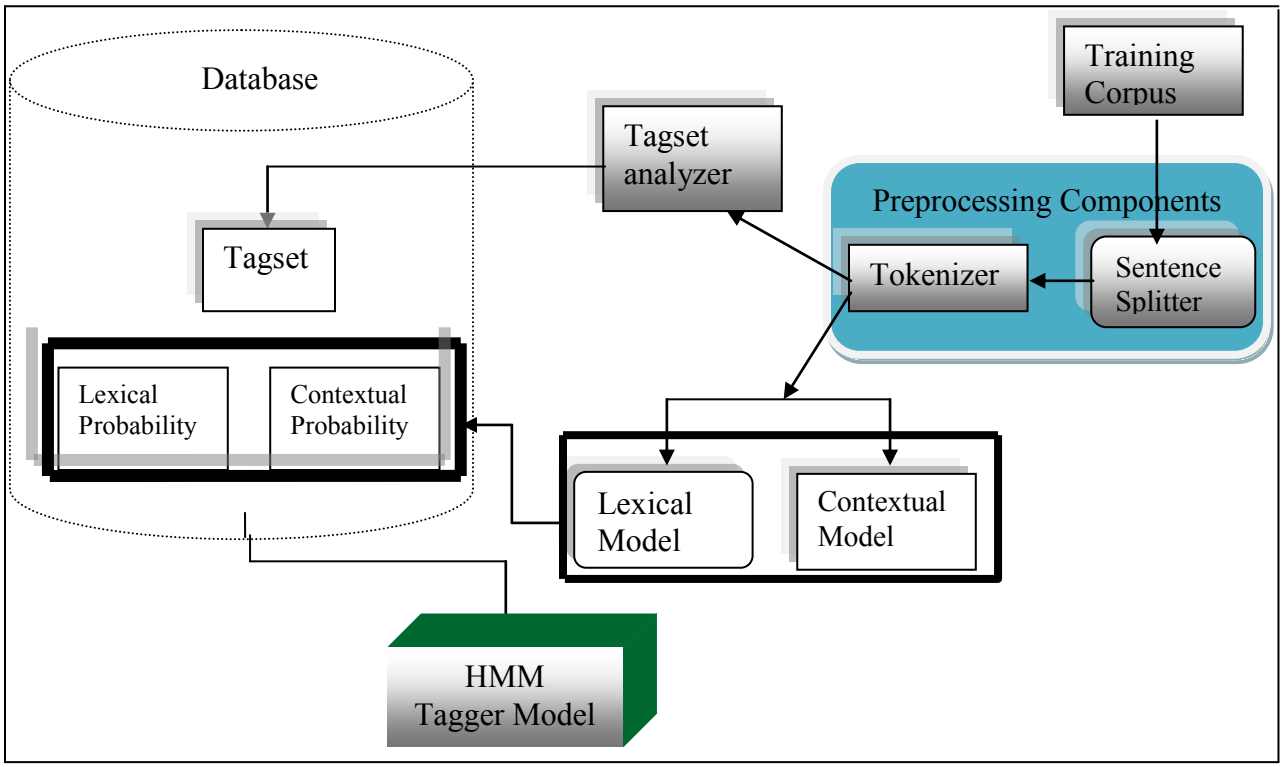


Figure 4.1 the HMM tagger trainer model

Figure 4.1 shows the HMM tagger trainer model. A supervised learning method is used for training the HMM model i.e. the training corpus is part of speech annotated Tigrigna text. The tagged corpus is an input to the model. The tagged corpus is given to the sentence splitter module in order to prepare it in a sentence level for training. The segmented sentences are given to the Tokenizer for splitting each sentence to a word level. After each sentence is tokenized into words, a tagset analyzer extracts the tags from the words and stores them in the database. The lexical and contextual models compute the lexical and contextual probabilities which are

important for finding a sequence of part of speech tags for the sequence of words in the input sentence.

After the tagger is trained, it is used for annotating untagged sentences which can in turn be evaluated against the manually tagged data (reference data) of the input untagged sentence. The graphical representation of the tagging and evaluation of the tagger is shown in figure 4.2.

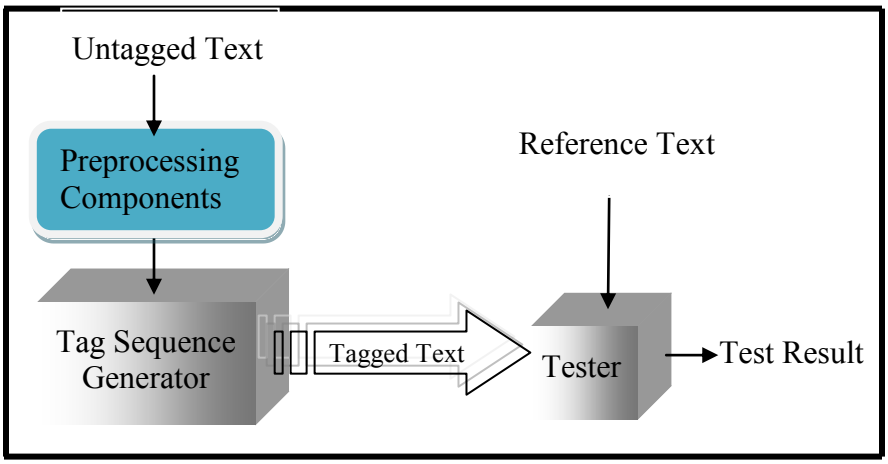


Figure 4.2 HMM tagger and evaluating process

The untagged text is given to the sentence splitter module and tokenizer preprocessing components so as to make ready for tagging by the Tag Sequence Generator. Afterwards, the Tag Sequence Generator selects an optimal part of speech tag sequence for the given word sequences and gives the tagged word sequences as an output. The tagged text is given to the tester component for comparison against a manually tagged (so called reference text) and this component gives accuracy of the tagging by counting the number of correctly tagged words.

The optimal sequence of part of speech tags for a given sequence of words in an input sentence to be tagged can be found using the Viterbi algorithm [7, 13, 47]. The Viterbi algorithm is a dynamic programming algorithm that finds the optimal path in the tagging process. It reduces the complexity of the HMM core issue, finding the best part of speech tag sequence for a given sequence of words in the input sentence, to polynomial time and the algorithm is linear in the number of words to be tagged [17]. In simple terms, the Viterbi algorithm calculates the probability of all possible paths of the word tag pairs in the input sentence. Afterwards, it will

select the path of the word tag pair with the highest probability to be the best path [7, 13]. It uses the lexical and contextual probabilities obtained from the lexical and contextual model to find the best path.

4.5 Designing Rule Based Tagger

Rule based part of speech tagger is another approach to tackle the problem of assigning part of speech tags to words in a sentence using rules. These rules can be manually designed rules by linguistic professionals or machine learned rules. In this thesis work, machined learned rules are selected as the handcrafted rules heavily rely on the skills of linguistic professionals and time consuming but machine learned rules can be obtained on the course training of the tagger. The works of [4, 27, 34] have used a machine learned rules to assign part of speech tags to sequence of words in an input sentence. Rules, also called Brill transformations, are derived on the supervised training of the tagger.

Brill"s Transformational-error driven Learning approach is adapted to drive the machine learned rules for designing the rule based tagger component of the Tigrigna POS tagger.

4.5.1. Transformation-based error-driven learning

The work of [10] developed by Eric Brill was based on rules, Brill transformations as he calls them, where the grammar is induced directly from the training corpus without expert knowledge. The only expert knowledge required is a correctly annotated corpus used for training which is the input to the tagger. The tagger can then learn and drive lexical and contextual information from the training phase which helps in labeling the likely POS tag for a word. After the tagger acquires the necessary knowledge in the training phase, it can be used for annotation of untagged corpora. The works of [27, 34] have adapted the Brill"s Transformation-Error driven Learning (TEL) to tackle the problem of part of speech tagging.

Brill"s rule based tagger learning also known as transformation based error driven learning is a framework that is based on rules that can be learned by detecting errors occurred in the previous steps. It has two phases: the initial state tagger and the learning phase. The TEL tagger takes

untagged corpus as an input and the initial state tagger assigns the likely tag for the words in the untagged corpus which then results in a new and temporary corpus as an output. The learning phase takes two input data namely the temporary corpus tagged by the initial state tagger and the goal corpus, manually tagged corpus assumed to be correct, which is used for comparison against the temporary corpus during rule derivation. The temporary corpus passes through the learner iteratively to derive Brill transformations. In each iteration, the learner derives a new rule, afterwards a comparison with goal corpus is done and the rule that improves the annotation is considered as a Brill Transformation and the temporary corpus is updated. The learning phase continues until no rules that can improve the tagging of temporary corpus (through comparison with the goal corpus) can be derived. By this process, the learner produces an ordered list of rules which can be applied for tagging untagged texts.

The learning phase of the Brill tagger has two sub-phases: the lexical rule learner, as its name implies, it derives lexical rules which are used for tagging unknown words and the contextual rule learner, learns the context of a word in a sentence and derives contextual rules which are used for the improvement of the accuracy of the tagger. Hence the TEL is used twice in the Brill‟s tagger learning phase: in the lexical rule learner and in the contextual rule learner. The lexical and contextual rule learner uses two corpora, the goal corpus and the temporary corpus. Then the goal of the tagger is to change the tags of the temporary corpus step by step in the learning phase to make it similar with the goal corpus as much as possible. The framework of the adapted Brill TEL tagger for this thesis work is shown in figure 4.3.

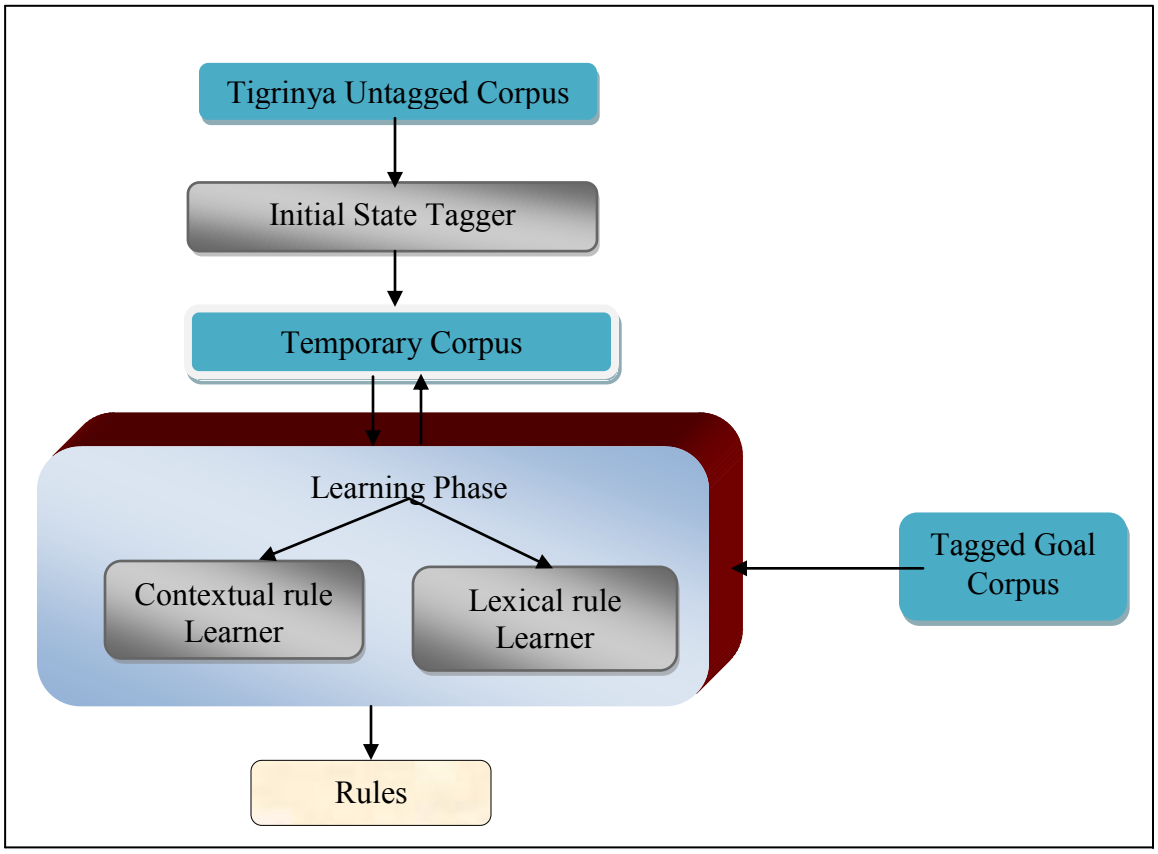


Figure 4.3 Brill Transformation-error driven learning tagger [10]

Sections from 4.5.2 up to 4.5.4 describe the details of the core components of the TEL Brill Tagger.

4.5.2. The initial state Tagger

The initial state tagger component takes untagged corpus as an input and tags this untagged corpus in some fashion. The initial state tagger can be chosen to be n-gram taggers (like unigram, bigram or trigram), default tagger, and/or sophisticated taggers etc. The choice of the tagger may depend on the final performance of the overall TEL Brill tagger. In this thesis work, different initial state taggers like the default tagger, unigram tagger, and bigram tagger are used for comparing the performance of the Brill tagger. The comparison of these initial state taggers is explained in section 6.3 of chapter six.

4.5.3. Rules

A rule is one component of the TEL Brill tagger that consists of two parts: a condition (the trigger and possibly a current tag), and a resulting tag. The rules are instantiated from a set of predefined transformation templates. They contain uninstantiated variables and are of the form:

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

Or

If Trigger, then change the tag to the tag Y

where X and Y are variables

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

4.5.4. Learning Phase

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

4.5.4.1 The Lexical Rule Learner

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

comparatively small set of words. The lexical rule learner component uses statistical methods to find the most likely tag of a word. Besides, the lexical rule learner deals with the morphology of the language in order to drive the set of all permissible rules and some of the Transformation Templates that are used in lexical rule learner are given below.

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

Template number 2 and 3 are for the original Brill tagger which implies adding/deleting of prefix/suffix of only up to 4 characters was considered. In this thesis work, after assessing the nature of Tigrigna words, the same trend as that of Brill templates 2 and 3 is found. It is possible to consider the following examples. But since the transliterated version of the texts is used, templates 2 and 3 are changed to be up to 6 suffixes.

ገሊፅኩምልና: /geli^Skumlna (You explained to us). It has four suffixes **ኩምልና**/kumlna (with six suffixes).

ተረዲኦምልካ:/teredi^omlka (They have understood you (Masculine)). It has three suffixes **ምልካ**/mlka (four suffixes).

Generally, the rule generating process takes an initially tagged temporary corpus (TC_0) which can be tagged by the initial state tagger, and finds the rule in PR which gets the best score when applied to TC_0. A best score for a rule means that the temporary corpus produced when applying the rule gives an annotation closer to the goal corpus. And this rule can be called as R1. Then R1 is applied to TC_0, producing TC_1. The process is now repeated with TC_1, i.e. it finds the rule in

PR which gets the best score when applied to TC_1. This will be rule R2 which then is applied to TC_1 producing TC_2. The process is done iteratively producing rules R3, R4, etc and corresponding temporary corpora TC_3, TC_4, etc until the score of the best rule fails to reach above some predetermined threshold value or until no rule can further improve the tagging of the corpus. The sequence of temporary corpora can be thought of as successive improvements closer and closer to the goal corpus. The output of the Lexical rule learner is the ordered list of rules R1, R2 ... which are used for tagging new unannotated texts.

The score for a rule R in PR is computed as follows: for each tagged word in TC_i the rule R gets a score for that word by comparing the change from the current tag to the resulting tag with the corresponding tag of the word in the goal corpus. Depending on the effect of the rule on the text to be tagged, the score of the rule R may be Positive, Negative or Zero. A positive score means that the rule improves the tagging of this word, and a negative score means that the rule worsens the tagging. If the condition of the rule is not satisfied then the score is zero. The total score for R, score(R), is then obtained by adding the scores for each word in TC_i for that rule. When the total score for each R is obtained the rule which has the highest score is added to the set of rules which have already been learned. Rules are ordered, i.e. the last rule is dependent on the outcome of earlier rules.

4.5.4.2 The Contextual Rule Learner

Once the tagger has learned the most likely tag for each word found in the tagged training corpus and the rules for predicting the most likely tag for unknown words, contextual rules are learned for disambiguation and better accuracy. The contextual rule learner finds rules on the basis of the particular environments i.e. the context of words. In order the contextual rule learner to generate rules, it needs the goal corpus and the initial temporary corpus TC_0 as an input. First, the learner generates the set of all permissible rules PR from all possible instantiations of all the predefined contextual templates. After it generates the contextual rules, it computes the score of each rule for a particular word. Then the learner can pick the rule R1 with the highest score and put on the rules component as an output. Afterwards the learner can take R1 and apply on TC_0 to get TC_1, on which the learning continues. The process is done iteratively putting one rule, the rule with

the highest score in each iteration, on the rule component as an output in each iteration until no rule achieves a score higher than some predetermined threshold value or until no rule can further improve the tagging of the corpus. Besides, the score of every Rule R is computed. Let R is a rule in PR, the score for R in TC_i can be calculated as follows: for each word W in TC_i, the contextual rule learner computes the score for R on this word W. Then the scores for all words in TC_i where the rule is applicable are added and the result is the total score for R. This can be done by comparing the tags of words in the tags of words in the TC_i with the correct tags of words in the goal corpus. If R is applied to the word W, thereby correcting an error, the score for W is +1. If applying R to the word W introduces an error then the score for W is -1. In all other cases, the score for the particular word W is 0. Therefore the total score for R is computed as follows.

score(R) = number of errors corrected - number of errors introduced.

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

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

4.5.5. Brill Tagger Architecture

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

architecture of the Brill tagger for this thesis work is given in figure 4.4 which is adapted from the work of [47] with a bit modification.

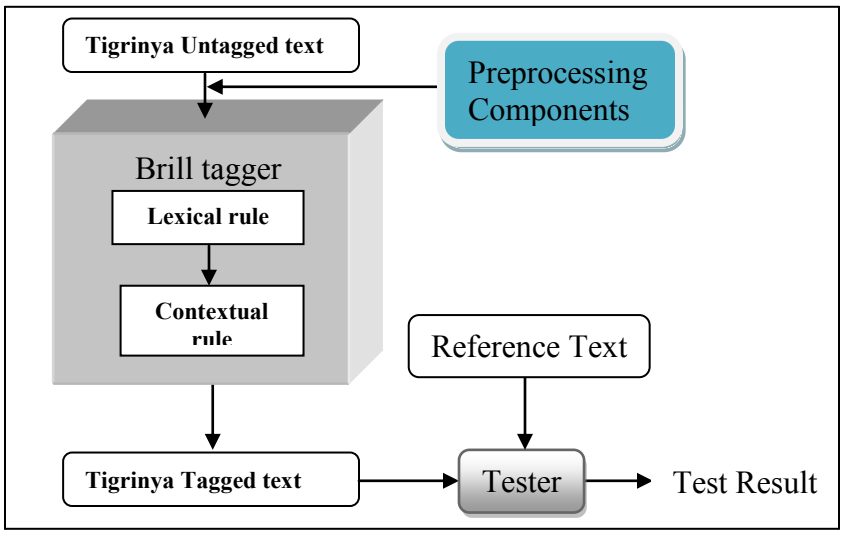


Figure 4.4 Adapted Brill Tagger for Tigrigna

The Brill tagger is given Tigrigna untagged text as an input then it tags this text using the rules that it has learned during the learning phase after applying the preprocessing components (sentence splitter module and Tokenizer) on the input untagged text. It first tags the text using the lexical rules. Since the lexical rules do not deal with contexts (environments of the word), the contextual rules are applied to look into the contexts of words in the tagged text and improve the performance of the tagger. Finally a tagged Tigrigna text is given as an output of the tagger. The output of the tagger is given to the tester component for comparison against the reference text to evaluate the performance of the tagger which gives the test result as an output.

4.6 Hybrid Tagger Architecture

The hybrid tagger consists of the HMM tagger and the Rule based tagger. The HMM tagger acts as an initial tagger for the raw text to be tagged and the rule based tagger corrects the output of the HMM tagger by applying Rules if the predetermined threshold value is not attained. The high level view architecture of the hybrid tagger for this thesis work is given in figure 4.5.

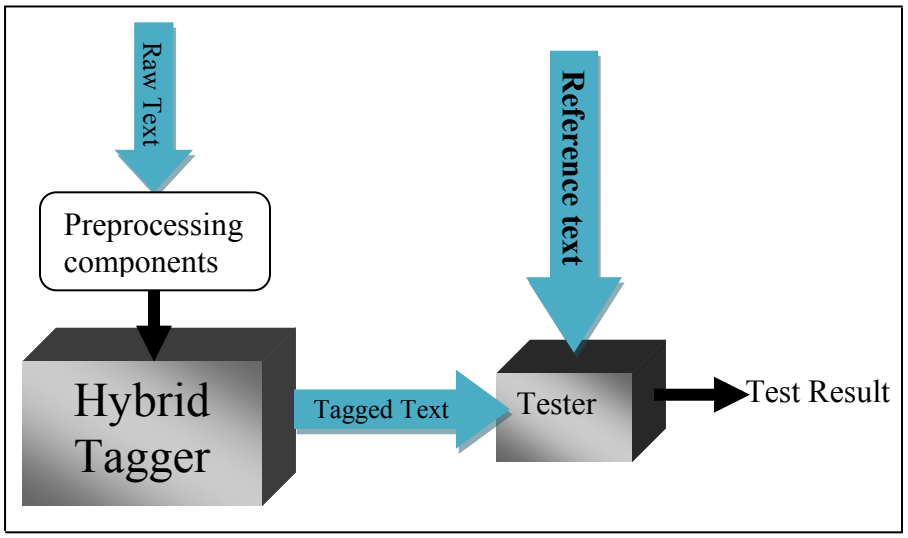


Figure 4.5 Hybrid tagger high level view

The higher view of the tagger is considered as a system that takes raw Tigrigna text, after processed by the sentence splitter module and Tokenizer preprocessing components, as input and gives the tagged Tigrigna text as an output. The hybrid tagger obtains rules and statistical information during the training of the rule based tagger and HMM tagger. The tagger uses these Rules and statistical information to tag the raw text with the appropriate tag and gives the tagged text as an output. The output of the hybrid tagger is then given for the tester component for comparison with the same text but manually tagged (reference text) which is assumed to be the correct one or the gold data to test the performance of the tagger. The technical description of the hybrid tagger architecture is shown in figure 4.6.

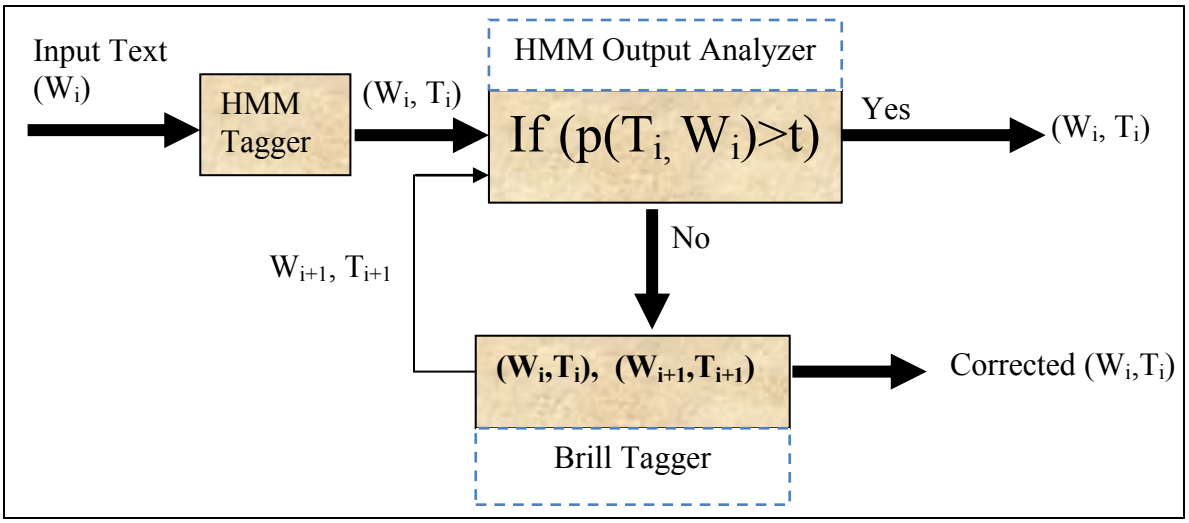


Figure 4.6 technical description of Hybrid tagger

When a word sequence W_i is given to the HMM tagger as an input, the HMM tagger assigns the tag sequences T_i using the aforementioned Viterbi Algorithm which implies that the output of the HMM tagger is the word tag sequence (W_i,T_i). This word tag sequence is given to the output analyzer that checks whether the predetermined threshold value for a word W_i 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 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 exhausted.

4.7 Summary

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

The statistical approach as its name implies extracts the statistical properties of words in the training phase to label words with their correct part of speech. One of the most widely used models in the statistical approach for part of speech tagging is the Hidden Markov Model. The main goal of this model is to assign an optimal sequence of part of speech tags to a sequence of words in a given sentence. The problem of finding the optimal sequence of part of speech tags to a sequence of words can be done using different algorithms of which the most used one is the Viterbi algorithm. Hence, the Viterbi algorithm is adapted for this Hidden Markov Model component of the tagger.

The rule based tagger is meant to disambiguate words to their correct part of speech tags using rules. These rules can be of handcrafted rules or machine learned rules. The handcrafted rules are tedious and time consuming as they are set by language experts while the machine learned rules can be obtained through training from a manually annotated corpus. As a result, the Brill Transformational Error driven learning tagger is adapted to derive the machine learned rules for the rule based component of the tagger.

A hybrid approach, rule based and statistical approach, combines the attractive properties of the two approaches so as to get better accuracy than its constituents. The HMM based tagger i.e. the statistical based tagger first tags the Tigrigna sentence then the HMM tagged word sequences are given for the rule based tagger for correction based on the fixed threshold value.

CHAPTER FIVE

IMPLEMENTATION OF TIGRIGNA PART OF SPEECH TAGGER

5.1 Introduction

Here the detail implementation of the different architectures that are dealt so far and corpus preparation is explained.

To begin with, Natural Language Toolkit (NLTK) and python are used in the entire implementation of the Tagger. The rationale behind the choice of these two tools is, they are suitable for processing different NLP tasks [3]. NLTK is an open source tool that contains open source python modules, linguistic data and documentation for research and development in natural language processing [3, 35, 44]. It supports many NLP tasks such as tokenizer, stemmer, part of speech tagger, classifier with distributions for Windows, Mac, and Linux for some languages such as English and German. Moreover, it contains different corpora for these two languages with their respective corpus reader module. Python is an easy to learn but powerful programming language especially for text processing in NLP applications [3, 28, 35]. It has efficient high level data structures and a simple but effective approach to object-oriented programming [35]. Its elegant syntax and dynamic typing, together with its interpreted nature make it an ideal language for scripting and rapid application development in many areas particularly in natural language processing on most platforms [35].

In the following sections, the details starting from the data preparation including incorporation of the data into NLTK as a module to the implementation of the hybrid tagger is discussed.

5.2 Corpus Preparation

Corpus, plural corpora, is a collection of text. It can be a flat text i.e. a text with no additional linguistic information or a text whereby each word in the text is attached with linguistic information [7, 13, 15,37, 40]. The corpus with additional linguistic information can be called as annotated/tagged corpus. Such linguistic information in the annotated corpus can be part of speech information, sentiment information that specify the word‟s word class category and

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

In fact, the tagged text i.e. the annotated corpus is thought to represent all the domains of the language. The domains can be text of news category, fiction category, editorial category, scientific category etc. A corpus with all possible categories is called a balanced corpus. Though it is difficult to prepare a balanced corpus, it is an important element in most natural language processing applications in general and part of speech tagging in particular [13].

Development of balanced corpus takes time, effort/skills of language experts and money as it needs data to be collected from different domains. Due to these constraints, a balanced corpus is not developed for this thesis work rather one category, news category, is selected as news texts are available and can be collected easily from different sources. 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. However, a category specific corpus contains words that are mostly used in that category and if a text from other category to be tagged is given to the tagger trained on this corpus, the performance of the tagger may be degraded. But if the text taken is from that category, the performance is assumed to perform as expected [13, 44].

As far as the researchers‟ knowledge is concerned, let alone a balanced corpus, there is no category based corpus developed for Tigrigna language. Therefore, an incremental approach is used for developing a category specific corpus. Tigrigna texts are first collected from different Tigrigna sources like Dimtsi Woyane Tigray, FM mekelle, Ethiopian Radio and Television Agency Tigrigna department, Woyen News Paper, Mekalh News paper and Tigray Development Association. Afterwards, these news texts are given to linguistic professionals for manual tagging. Since the manual tagging process is tedious and time consuming activity, around two hundred raw Tigrigna news text sentences are collected first which were tagged manually on

paper by the linguistic professionals. These tagged news text sentences were used for training the rule based tagger. Then, again additional raw Tigrigna news texts were collected from the aforementioned sources which are initially tagged using the trained rule based tagger. Afterwards, the tagged text is given to language experts for correction and approval. The tagged corpus used for training is updated to contain the new correctly tagged and approved text which is in turn used for training the tagger. This process is repeated until the desired size of the corpus for this thesis work is achieved. A sample corpus is shown in appendix A.

5.3 Implementation of Preprocessing components

In this thesis work, supervised learning approach is used for the Tigrigna POS tagger. The tagger takes tagged training corpus as an input which needs to be preprocessed by the sentence splitter module, Tokenizer and tagset analyzer components. Moreover, since Tigrigna uses Geez writing script, a python code that transliterates the Tigrigna tagged corpus to Latin characters is written so as to be processed by the tools used. The transliteration of the Tigrigna/Geez characters is listed in Appendix B from the work of [12]. It is the transliterated corpus that is used for training the system. On top of this, a piece of python code that reads the Tigrigna corpus in the NLTK tool is written. The corpus reader will read the contents of the corpus and give to the sentence splitter module which splits the corpus based on Tigrigna sentence end markers which can be one from the characters „፡ /.' '?' or „!'.

Afterwards, the output of the sentence splitter component is given to the Tokenizer component to reduce the sentences into word level. This component splits the sentences into words/tokens using the space character. The tokens/words, in this case, can be any Tigrigna word and punctuation marks.

The Tokenizer component prepares the words, in fact, during the training phase, the word comprises two parts namely the token/word and its corresponding part of speech, for finding the statistical properties of words and the part of speech tags. The tags can be extracted from the output of the Tokenizer by the help of the tagset analyzer component. The purpose of extracting tagsets is for the sake of physical storage for later use by the HMM tagger as an input during the tagging of new text. This component extracts the tag of a token by splitting the output of the

Tokenizer using the slash (/) character as this is the character that is used during the manual tagging of the corpus to separate the word and its corresponding tag. This preprocessing component is done only during the training phase of the tagger whereas the other two components namely the sentence splitter and Tokenizer components are done during the training as well as the testing phase.

5.4 Implementation of Hybrid Tagger

The hybrid tagger in this thesis work is implemented as a two step process. The first step is performed by the HMM 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. Sample rules are given in Appendix C that the tagger has learned during the course of its training 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 in figure 5.1

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. Fix sliding window size of 2
8. While there are HMM tagged words compare the probability of each tag for each word against θ:
9. If $P(T_i|W_i) < \theta$ then
    - 9.1. Apply rules to W_i and W_{i+1}
    - 9.2. Get $W_i|T_i$ as an output assumed to be the corrected one.
10. Else the selected tag suffices and considered to be correct and taken as an output.
11. End of the HMM tagged words

Figure 5.1 Hybrid tagger algorithm

5.5 Summary

In the Tigrigna POS tagger work, NLTK and Python programming language are used as implementation tools due to their ease of application in natural language processing [28, 35, 38]. They are easy to use and process text with different integrated components.

There is no ready made Tigrigna corpus that researchers can make use of it and hence a domain specfic corpus is developed using incremental appraoch for this thesis work.

The preprocessing components (sentence splitter, tokenizer and tagset anlyser) are developed for use by the HMM tagger and the rule based tagger. The viterbi algorithm is implemented for finding the optimal path in the HMM tagger and the TEL brill tagger is customised for the rule based tagger. The hybrid tagger is implemented as a combination of the two by making the HMM tagger as an initial tagger and the TEL brill tagger as a corrector.

CHAPTER SIX

EXPERIMENTS AND PERFORMANCE ANALYSIS

6.1 Introduction

Different experiments have been conducted on Tigrigna HMM tagger, Rule based tagger and Hybrid tagger. The corpus is divided into two sets: the training set and the testing set. The former one comprises 75% of the corpus while the remaining 25% are used for testing purpose. In this chapter, the detail experiments conducted for this thesis work are discussed briefly.

6.2 Experiments with HMM tagger

The NLTK HMM tagger tool with some modifications is used for conducting experiments on Tigrigna HMM tagger. Ten different experiments are conducted on the HMM tagger using different portions of the training set to see the goodness of the training set based on the observation that can be made on the learning curve. The researchers have started training the system using the 10% of the training set. After the tagger is trained, its performance is measured on the 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. In fact, the desired performance of the tagger is considered to be the performance measured from the learning curve shown in figure 6.1 with almost constant value regardless of the training data added for training the system. Table 6.1 shows the different experiments conducted using different portions of the training set with the corresponding performance of the tagger. The curve shows that the available training set is almost sufficient despite lack of already tagged corpus. Moreover, the performance obtained when all of the training data is used (100%) shows the attainable performance of the HMM model.

Table 6.1 HMM tagger performance

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance(%)	50.95	65.56	70.08	72.71	75.31	76.77	78.34	87.22	88.83	89.13
difference	50.95	14.61	4.52	2.63	2.6	1.46	1.57	8.88	1.61	0.3

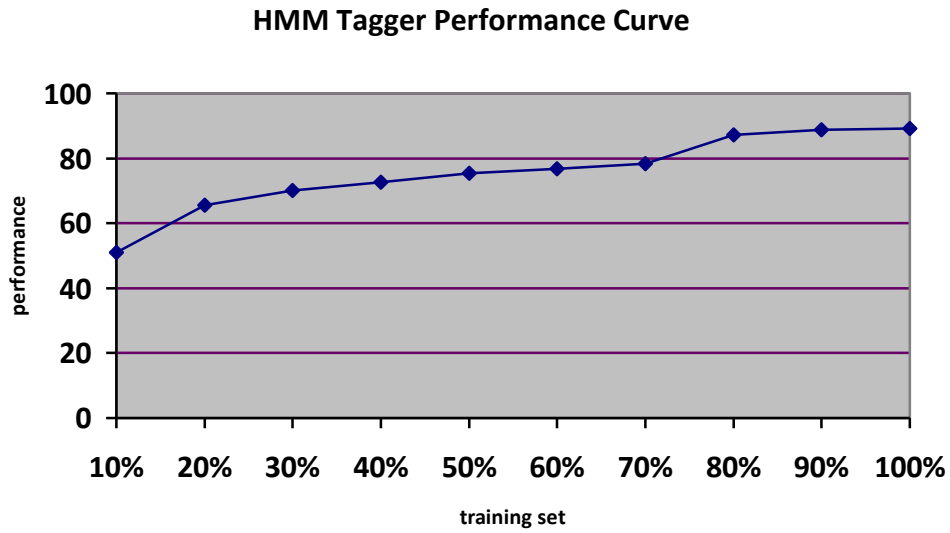


Figure 6.1 HMM tagger performance curve analysis

6.3 Experiments with Rule based tagger

The NLTK Brill tagger with modifications is used for conducting experiments in the rule based tagger. As it is done for the HMM tagger, ten different experiments are also conducted on the rule based tagger using different portions of the training set and different initial state annotators. Table 6.2 and figure 6.2 show the different experiments conducted using different portions of the training set with the corresponding performance of the rule based tagger for the different initial state annotators: default, unigram, bigram and trigram taggers.

Table 6.2 Rule based tagger performance using different initial state taggers

Training set		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance per different initial state taggers (%)	Defualt tagger	40	47	47	46	47.1	48	53	57	57.4	58
	Unigram tagger	72.7	78.1	79.8	80.9	82.5	83.2	84	86.2	91.4	91.8
	Bigram tagger	65	70	71.6	72.7	74.1	75.8	76.2	78.5	86.4	88.3
	Trigram tagger	58.1	61.7	63.9	65.2	65.8	66.6	67.2	69.3	80.3	82

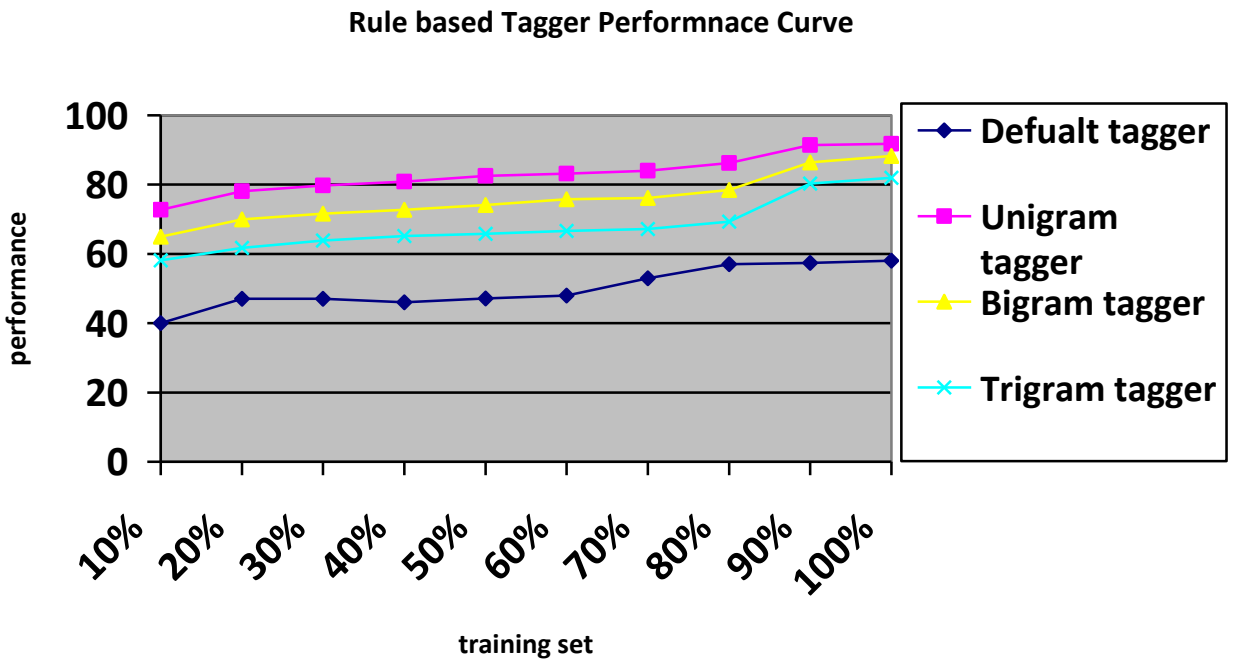


Figure 6.2 Rule based tagger performance curve analysis

The default tagger assigns a specified default tag for each word. In this thesis work, based on the performance of the rule based tagger when it takes the default tagger as an initial annotator, the Noun (N) part of speech is selected to be the default tag. The 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 [3, 38]. Higher performance of the rule based tagger is obtained when the initial state annotator is unigram tagger which implies that the training corpus used does not have many bigram and trigram words that can be found in the testing data. This is reflected by the performance change in bigram and trigram taggers even at the last column of the figure. However, the rule based tagger has 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 N (Noun) tag.

6.4 Experiments with Hybrid tagger

The hybrid tagger is composed of the HMM tagger and rule based tagger. 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. The threshold value, in this thesis work, is fixed to 0.5 since taking threshold value less than this value does not bring significant difference on the performance of the tagger. As the threshold value increases, the rule-based tagger corrects more words. As a result of this, the hybrid tagger scores the highest performance as the threshold value goes up, as it is indicated in table 6.3. Therefore, by fixing the threshold value to 0.5, an overall performance of 95.88% is obtained. But 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.

The different performances of the hybrid tagger on different threshold values are given in table 6.3 and figure 6.3.

Table 6.3 Hybrid tagger performances on different threshold values

Threshold value	0.02	0.05	0.08	0.11	0.14	0.17	0.5	0.9	1
Performance (%)	94.84	94.86	95	95	95	95	95.88	95.88	93.5

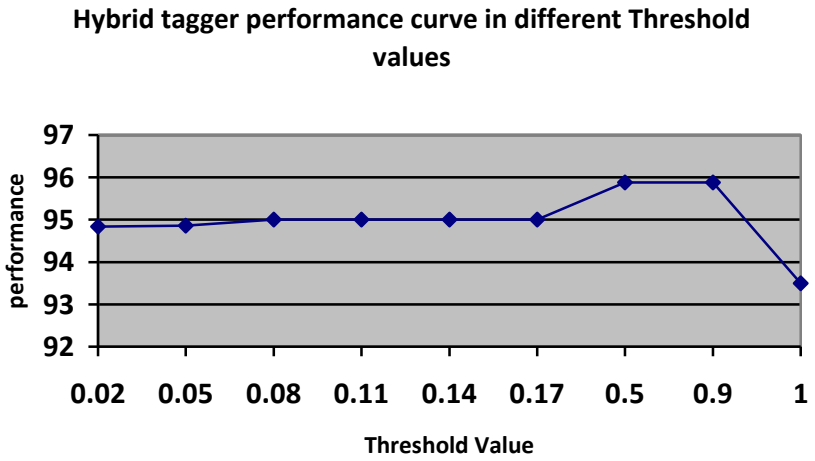


Figure 6.3 performance analysis of hybrid tagger

6.5 Performance Analysis

In order to analyze the performance of the three different taggers for the different part of speech tags, the frequency of the taggers in the entire corpus, training set and testing set is considered. Moreover, confusion matrix is developed for the three Tigrigna POS taggers. A total of 36 tags are identified in this research work and based on their frequency, they are divided into two groups namely the 15 most frequent tags and the rest as others. The frequency of the tags is given in table 6.4.

Table 6.4 POS tags frequency

tag	Entire Corpus Frequency	Training Frequency	testing	
			Frequency	%
N	8360	6658	1702	20.36%
VREL	2141	1686	452	21.11%
PREP	2021	1559	462	22.86%
V	1895	1559	336	17.73%
NP	1543	1301	242	15.68%
ADJ	1482	1189	293	19.77%
NC	1306	1022	284	21.75%
CARDN	837	665	172	20.55%
ADV	694	576	118	17.00%
C	622	516	106	17.04%
PROND	601	475	126	20.97%
AUX	576	462	114	19.79%
PRONDPREP	356	266	90	25.28%
VRELPREP	328	295	33	10.06%
NPREP	298	249	49	16.44%
Others	2940	1522	1421	48.33%
Total	26,000	20,000	6000	23.08%

The confusion matrix for the rule based Tigrigna part of speech tagger is given in table 6.5.

Table 6.5 Rule based tagger confusion matrix using the unigram tagger as an initial state annotator

		Test tags (Predicted tags)																	
Reference tags (Desired tags)		N	PREP	VREL	V	ADJ	NC	NP	CARDN	PROND	ADV	AUX	C	PRONDPREP	NPREP	VRELPREP	others	**total**	Performance (%)
	N	**1691**			2	4				1	2	2						1702	99.35
	PREP		**461**							1								462	99.78
	VREL	101		**351**														452	77.65
	V	69		1	**264**							1	1					336	78.57
	ADJ	33				**260**												293	88.74
	NC	58					**226**											284	79.58
	NP	19					1	**222**										242	91.74
	CARDN	8							164									172	95.35
	PROND	1								**125**								126	99.21
	ADV	21									97							118	82.20
	AUX	1			3							**110**						114	96.49
	C	1	1										**104**					106	98.11
	PRONDPREP	4												**86**				90	95.56
	NPREP	14													**35**			49	71.43
	VRELPREP	9														**24**		33	72.73
	others	122															**1299**	1421	91.41
	Total	2152	462	352	269	264	227	222	164	127	99	113	105	86	35	24	1299	6000	91.80

The rule based tagger confusion matrix shows that it assigns 5519 tags correctly and 481 tags wrongly to the tokens in the testing set. The performance of the rule based tagger varies for the different part of speech tags with a higher performance for PREP part of speech tag followed by N, PROND, C, AUX, PRONDPREP, CARDN, NP, ADV, NC, V, VREL, ADJ, VRELPREP, NPREP part of speech tags for the given testing set trained on the training set. Besides, more part of speech tags are wrongly assigned to be noun (N) due to the fact that the rule based tagger is modified to tag tokens with no transformation rules as nouns using a backoff method in NLTK. The backoff method is a mechanism of referring back for some ways of labeling tokens if the rule based tagger fails to find the correct way of tagging a given token. In fact, the reason behind not getting enough transformation rules indicates the lack of a standard corpus for Tigrigna language.

The confusion matrix for the HMM based Tigrigna part of speech tagger is given in table 6.6.

Table 6.6 HMM Tagger Confusion matrix

		Test tags (Predicted tags)																	
Reference tags (Desired tags)		N	PREP	VREL	V	ADJ	NC	NP	CARDN	PROND	ADV	AUX	C	PRONDPR	NPREP	VRELPRE	others	**total**	Performance (%)
	N	**1545**	6	9	14	11	19	27	15	4	1	9	1	2			39	1702	90.78
	PREP	3	**458**							1								462	99.13
	VREL	10	5	**389**	5		9	8	17		1	1	1				6	452	86.06
	V	9	3	1	**285**	2	6	7				4	1		1		17	336	84.82
	ADJ	6	3		3	**262**	5		5								9	293	89.42
	NC	5	4		3	2	**229**	11	2		2	3		1	1		21	284	80.63
	NP	5			1		2	**231**	2								1	242	95.45
	CARDN	2		7	1		3		**155**			1					3	172	90.12
	PROND			1			1			**123**							1	126	97.62
	ADV	7	1		2	1	4	1			**90**	2	1	1			8	118	76.27
	AUX			3	3							**107**					1	114	93.86
	C		2					1					**103**					106	97.17
	PRONDPREP	1			1			1	1					**84**			2	90	93.33
	NPREP	5	1	1	2			2	1			1			**31**		5	49	63.27
	VRELPREP	1	1			2	1				1					**21**	6	33	63.64
	others	164															**1257**	1421	88.46
	Total	1763	484	411	320	280	279	289	198	128	95	128	107	88	33	21	1376	6000	89.13

The HMM tagger confusion matrix shows that it assigns 5370 tags correctly and 630 tags incorrectly. The HMM tagger has confused the tags to other different part of speech tags; for instance out of 1702 nouns 157 are assigned wrongly to be different tags. 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 confusion is due to the fact that there is no standard and large corpus for training the tagger. Like that of the rule based tagger the performance of the HMM tagger is different for the different part of speech tags. The order of performance of the HMM tagger for the part of speech tags in descending order is: PREP, PROND, C, NP, AUX, PRONDPREP, N, CARDN, ADJ, VREL, V, NC, ADV, VRELPREP, and NPREP.

The confusion matrix for the hybrid Tigrigna part of speech tagger is given in table 6.6.

Table 6.7 Hybrid tagger Confusion matrix

		Test tags (Predicted tags)																	
Reference tags (Desired tags)		N	PREP	VREL	V	ADJ	NC	NP	CARDN	PROND	ADV	AUX	C	PRONDPR	NPREP	VRELPRE	others	**total**	Performance (%)
	N	**1641**	2		20	10	3	12		1	2	2		2	1		6	1702	96.42
	PREP		**461**							1								462	99.78
	VREL	1		**451**														452	99.78
	V	21			**310**	1						1	1		1		1	336	92.26
	ADJ	22	1		1	**266**		2									1	293	90.78
	NC	42	1		5	1	**228**	4									3	284	80.28
	NP	13			5	1	1	**222**										242	91.74
	CARDN	8							**164**									172	95.35
	PROND									**126**								126	100.00
	ADV	1	1		1						**112**		1			1	1	118	94.92
	AUX			1	5							**108**						114	94.74
	C		2										**104**					106	98.11
	PRONDPREP	1												**89**				90	98.89
	NPREP	1	1		2										**42**	1	2	49	85.71
	VRELPREP															**33**		33	100.00
	others	127															**1294**	1421	91.06
	Total	1878	469	452	349	279	232	240	164	128	114	111	106	91	44	35	1308	6000	95.88

The Hybrid tagger confusion matrix shows that it assigns 5651 tags correctly and 349 tags incorrectly. The confusion of tags by the hybrid tagger is less than the confusion made by the HMM tagger and rule based tagger which clearly shows performance improvement on the overall tags compared to the individual taggers. The errors of assigning tags to other undesired tags is attributed to different factors such as lack of standard corpus, linguistic patterns like some Tigrigna nouns can be adjectives and incorrect labeling of the words in the prepared corpus. Moreover, the confusion made by the hybrid is due to the fact that it takes probabilities and rules into consideration. Like that of the rule based and HMM taggers, the performance of the hybrid tagger is different for the different part of speech tags. The order of performance of the hybrid tagger for the part of speech tags in descending order is: PROND, VRELPREP, PREP, VREL, PRONDPREP, C, N, CARDN, ADV, AUX, V, NP, ADJ, NPREP, and NC.

6.6 Summary

Different experiments are conducted for the Tigrigna HMM, rule based and hybrid tagger. Accordingly, different performances are obtained: the rule based tagger performed better than the HMM and the hybrid (HMM tagger followed by the rule based tagger) performs better than the individual taggers. The performance of the Tigrigna HMM tagger and rule based tagger is 89.13% and 91.8% respectively while the overall performance of the hybrid tagger is 95.88%. These performances are attained on the prepared testing set; but if there is more data for training and testing, it may be possible to get higher performances than the achieved ones.

A performance comparison for each part of speech tag for the HMM and Hybrid models is given in table 6.8 to see the performance improvement through making a hybrid of the HMM and rule based taggers.

Table 6.8 Performance improvements by the hybrid model compared to HMM model

POS tag	Hmm performance (%)	Hybrid performance (%)	Difference
VRELPREP	63.64	100.00	36.36
NPREP	63.27	85.71	22.45
ADV	76.27	94.92	18.64
VREL	86.06	99.78	13.72
V	84.82	92.26	7.44
N	90.78	96.42	5.64
PRONDPREP	93.33	98.89	5.56
CARDN	90.12	95.35	5.23
PROND	97.62	100.00	2.38
ADJ	89.42	90.78	1.37
C	97.17	98.11	0.94
AUX	93.86	94.74	0.88
PREP	99.13	99.78	0.65
NC	80.63	80.28	-0.35
NP	95.45	91.74	-3.72

As shown in table 6.8, the hybrid model has significant effect on VRELPREP, NPREP, ADV, VREL, V, N, PRONDPREP, CARDN, PROND, ADJ, C, AUX, and PREP part of speech tags while it has negative impact on the two POS tags: NC and NP. This indicates that making a hybrid does degrade the performance for the last two tags. The performance decrement for the two tags could be due to the reason that there is no learned transformation rule for these tags during the learning phase of the rule based tagger. The lack of the transformation rules could be also due to the fact that there is no standard corpus and even there might be errors made while labeling the words in the prepared corpus for this thesis work.

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATION

7.1 Conclusion

Part of speech tagging is the process of labelling words with their corresponding part of speech categories. It is a hot research area in the field of natural language processing for different languages. Moreover, part of speech tagging can be conceived as the problem of assigning part of speech tags to a word in a sentence. This problem can be solved using different techniques among which the hybrid of the HMM and Rule based approach is assumed to perform better than the HMM and rule based taggers taken independently.

A hybrid approach, which is the combination of rule based and HMM tagger, is designed for Tigrigna part of speech tagger. The choice behind the hybrid tagger is, it performs better than the individual ones.

Data corpus is an important component in Natural language processing in general and part of speech tagging in particular. And hence, a corpus with a total of 26, 000 words is collected from different Tigrigna news broadcasting agencies. 36 part of speech tags are identified as tagsets 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 corpus is divided 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).

NLTK and Python2.6 are used in the implementation and experiment of the Tigrigna tagger. 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. Therefore, it is

possible to conclude that the hybrid tagger performs better than HMM tagger and rule based tagger used separately. In addition, this research work has got promising results on the experimented three models though a large size and balanced corpus is not developed.

7.2 Recommendation

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

Finally this research work suggests the following items as a future work:

- Comparative study of three different approaches (HMM based, rule based tagger , ANN based taggers for Tigrigna Language with more training and testing data)
- Extending this work by training in large corpus and using large tagsets that can identify gender, number, tense etc with different feature set
- Comparison of two hybrid approaches: the hybrid of rule based and HMM tagger and the hybrid of rule based and ANN for Tigrigna language
- Conducting similar researches for other local languages by adapting this work such as kunamigna and erop.

References:

[1]. Ankur Parikh, (2009 (unpublished)). Part-Of-Speech Tagging using neural network, International Conference on Natural Language Processing (ICON-2009), Report No: IIIT/TR/2009/232.

[2]. Biniam Gebremicheal, Wordlist and Spell checking for Amharic and Tigrigna, http://www.cs.ru.nl/~biniam/geez/index.php visited on October 02, 2009

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

[4]. Brill, Eric (1992) A simple rule-based part of speech tagger. In Proceedings of the Third ACL Applied NLP, , Trento, Italy, pages 152-155.

[5]. Carlberger and V. Kann (1999). Implementing an efficient part-of-speech tagger, John Wiley & Sons, Inc. New York, USA, Vol. 29, pp. 815-832

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

[7]. D. Jurafky and J. H. Martin, (2006). Speech and Language Processing, An introduction to natural Language Processing, Computational Linguistics, and speech recognition. 2nd ed. New Jersey, Prentice Hall.

[8]. Daelemans, W., J. Zavrel, and P. Berck, (1996). Part-of-speech tagging for Dutch with MBT, a memory-based tagger generator. In K. van der Meer, editor, Informatiewetenschap 1996, Wetenschappelijke bijdrage aan de Vierde Interdisciplinaire Onderzoeksconferentie Informatiewetenchap, The Netherlands. TU Delft, pages 33-40.

[9]. ElHadj, Y.O.M., I.A. Al-Sughayeir and A.M. Al-Aansari, (2009). Arabic part-of-speech tagging using the sentence structure. Proceeding of the 2nd International Conference on Arabic Language Resources and Tools, Apr., Cairo, Egypt. pp: 241-245

[10]. Eric Brill, (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: a Case Study in Part-of-Speech Tagging, Department of Computer Science, Association for Computational Linguistics, the Johns Hopkins University, Vol.21, No.

4, pp. 543-565.

[11]. Fahim Muhammad Hasan, Naushad UzZaman, Mumit Khan, (2006). Comparison of Different POS Tagging Techniques (n-grams, HMM and Brill‟s Tagger) for Bangla, International Conference on Systems, Computing Sciences and Software Engineering (SCS2 06) of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 06), pp: 4-14.

[12]. Gasser, M., (2009). Semitic morphological analysis and generation using finite state transducers with feature structures. Conference of the European Chapter of the Association for Computational Linguistics, Vol. 12

[13]. Getachew Mamo, (2009). Part-of-Speech Tagging for Afaan Oromo Language. Masters thesis, Addis Ababa University.

[14]. Girma Awgichew & Mesfin Getachew, (2006). Manual Annotation of Amharic News Items with Part-of-Speech Tags and its Challenges., ELRC Working Papers Vol. 2; number 1:, AAU Printing Press.

[15]. Girma Berhie, (2001). A Stemming Algorithm Development for Tigrigna Language Text documnets: Masters thesis, Addis Ababa University.

[16]. Gordana Llic Holen, (May 25, 2009). HMM tagger for Swidish.

[17]. Hall, Johan, (2003). A Probabilistic Part-of-Speech Tagger with Suffix Probabilities. Master‟s thesis, School of Mathematics and Systems Engineering, Växjö University.

[18]. Helmut Schmid, (1994). Probabilistic Part of Speech Tagger using Decision Trees, Institute of Natural Language Processing , university of Stuttgart, Stuttgart, Germany, pp. 172-176

[19]. http://jedlik.phy.bme.hu/~gerjanos/HMM/node2.html/ visited on June 21, 2010

[20]. http://www.doc.ic.ac.uk/~nd/surprise_96/ visited on May 07, 2010

[21]. J. Nivre, (2000). Sparse data and smoothing in statistical part-of-speech tagging, Journal of Quantitative Linguistic, Vol 7, pp:1–17.

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

[23]. Language-Specific Tools (Part of Speech Tagger): http://www.xrce.xerox.com/competencies/content-analysis/toolhome.en.html: visited on August 26, 2009

[24]. Levent Altunyurt & Zihni Orhan, (2006). Part of Speech Tagger for Turkish, Masters Thesis, Computer Engineering, Bo_azici University.

[25]. Liddy, E. D., (2003). Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. Marcel Decker, Inc.

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

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

[28]. Python Programming Language: http://www.python.org/ visited on may 15, 2010

[29]. Resource portal for the Amharic NLP, http://nlp.amharic.org/ visited on Aug. 30, 2009

[30]. Result and Demo of SNOW POS tagger for English: http://l2r.cs.uiuc.edu/~cogcomp/eoh/posdemo.html: visited on October 02, 2009.

[31]. Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu, (2004). A Hybrid Model for Part-of-Speech Tagging and its application to Bengali, International Journal of Information Technology Vol. 1.

[32]. Schmid H., (1994). Part-of-Speech Tagging With Neural Networks. Proc. COLING‟94, Kyoto, Japan, pp:172-176, 1994.

[33]. Shereen Khoja, (2001). APT: Arabic Part-of-speech Tagger. Proceeding of the Student Workshop at the 2nd Meeting of the NAACL, (NAACL‟01), Carnegie Mellon University, Pennsylvania, pp: 1-6

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

[35]. Steven Bird, Ewan Klein, and Edward Loper, (2009). Natural Language Processing with Python: Analysing Text with the Natural Language Toolkit, O'Reilly Media. 1st

ed.

[36]. Tesfaye Tewolde (PhD), (2002). A modern grammar of Tigrigna, Tipografia U. Detti – via G. Savonarola Roma.

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

[38]. The Natural Language Toolkit: http://www.nltk.org/ visited on May 15, 2010

[39]. The SNoW Learning Architecture**,** http://l2r.cs.uiuc.edu/~danr/snow.html, visited on October 02, 2009

[40]. Thomas C. Rindflesch, (1996). Natural Language Processing, Sematic Knowledge Representation research paper, USA.

[41]. Tigrigna Alphabet and pronunciation: http://www.omniglot.com/writing/Tigrigna.htm: visited on August 26, 2009.

[42]. Tigrigna Grammar, (1996). American Evangelical Mission, First Red Sea Press, Inc., Edition.

[43]. Tigrigna Language: http://en.wikipedia.org/wiki/Tigrigna_language: visited on August 26, 2009.

[44]. Tony McEnery & Andrew Wilson, (2001). Corpus Linguistic, Edinburgh University Press; 2nd ed.

[45]. Tunga Gungor, A composite approach for part of speech tagging in Turkish, Bogazici University, Istanbul, Turkey

[46]. Uí Dhonnchadha, E., (2009). Part-of-Speech Tagging and Partial Parsing for Irish using Finite-State Transducers and Constraint Grammar, School of Computing, Dublin City University: PhD Thesis.

[47]. Yahya O. Mohamed Elhadj, (2009). Statistical Part-of-Speech Tagger for Traditional Arabic Texts, Journal of Computer Science, Science Publications, Vol 5, pp: 794-800

[48]. YAMINA, T.-G., (2005). Tagging by Combining Rules-Based Methods and Memory-Based Learning for Arabic Text. Proceedings of world academy of science, engineering and technology, Volume 6 pp. 245-248.

[49]. ሰለሞን ገብረክርቶስ, (1985 ዓ.ም). ሓረጋት ትግርኛ፣ ኣስመራ

[50]. ዳኒኤል ተኽሉ ረዳ, (1996 ዓ.ም). ዘመናዊ ስዋሰውቋንቋ ትግርኛ፣ ኣ.ኣ

[51]. Simon Haykin, (1994). Neural Networks: A Comprehensive Foundation, 2nd ed. Prentice Hall PTR, Upper Saddle River, NJ.

[52]. Richard P. Lippmann, (1988). An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, Vol. 4 (1987), pp:4-22, at pp:15-18; DARPA, Darpa Neural Network Study (Fairfax, VA: Armed Forces Communications and Electronics Association [AFCEA] International Press), pp:78-80

Appendices

Appendix A : sample corpus

Original Tigrigna corpus

ኣብ/PREP ወረዳ/N ወልቃይት/NP ፅርየት/N ትምህርቲ/N ኣብ/PREP ምርግጋፅ/N ዘተባብዑ/VREL ለውጥታት/N ይምዝገቡ/V ምህላዎም/N ተገሊፁ/V ።/PUNC

ቤትፅሕፈት/N ርክብ/N ህዝብን/NC መንግስትን/NC እታ/PROND ወረዳ/N ከምዝገለፆ/VREL ድሕሪ/C እቲ/PROND ኣብ/PREP ፅርየት/N ትምህርቲ/N ኣድሂቡ/V ብደረጃ/NPREP ወረዳ/N ዝተሳለጠ/VREL ኮንፈረንስ/N ኣብ/PREP ምርግጋፅ/N ፅርየት/N ትምህርቲ/N እንታይ/PRONI ዓብዪ/ADJ ዘተታት/N ኣመዝኒብና/V ዝብል/VREL ሰፊሕ/ADJ ገምጋማዊ/ADJ መድረኽ/N ድማ/C ልዕሊ/PREP 100/CARDN ርእሳነመምህራንን/NC ሱፐርቫይዘራትን/NC ተሳቲፎም/V እዮም/AUX ።/PUNC

ብመሰረት/C እዚ/PROND ድሕሪ/C እቲ/PROND ዝተሳለጠ/VREL ኮንፈረንስ/N ሞዴል/N ከፍልታትን/NC መምህራን/NC ከምዝተፈጠሩን/VRELC ብዓቕሚ/NPREP ንዝተሓቱ/VRELPREP ተምሃሮ/N ፍሉይ/ADJ ሓገዝ/N ይዋሃቦም/V ከምዘሎን/VRELC ብምሕባር/ADV ብሓፈሻ/ADV ፅርየት/N ትምህርቲ/N ንምርግጋፅ/VI ዘተባብዑ/VREL ጅማሮታት/N ይምዝገቡ/V ኣለዉ/AUX ።/PUNC……..

Transliterated corpus

^ab/PREP wereda/N welqayt/NP ^Sryet/N tmhrti/N ^ab/prep mrgga^S/N zetebab`u/VREL lewTtat/N ymzgebu/V mhlawom/N tegeli^Su/V ./PUNC
bEt^SHfet/N rkb/N hzbn/NC mengstn/NC ^ta/PROND wereda/N kemzgele^So/VREL dHri/C ^ti/PROND ^ab/PREP ^Sryet/N tmhrti/N ^adhibu/V bdereja/NPREP wereda/N ztesaleTe/VREL konferens/N ^ab/PREP mrgga^S/N ^Sryet/N tmhrti/N ^ntay/PRONI `abyi/ADJ zetetat/N ^amezgibna/V zbl/VREL sefiH/ADJ gemgamawi/ADJ medreK/N dma/C l`li/PREP 100/CARDN r^sanememhrann/NC supervayzeratn/NC tesatifom/V ^yom/AUX ./PUNC
bmeseret/C ^zi/PROND dHri/C ^ti/PROND ztesaleTe/VREL konferens/N modEl/N kfltatn/NC memhran/NC kemztefeTerun/VRELC b`aQmi/NPREP nzteHatu/VRELPREP temharo/N fluy/ADJ Hagez/N ywahabom/V kemzelon/VRELC bmHbar/ADV bHafexa/ADV ^Sryet/N tmhrti/N nmrgga^S/VI zetebab`u/VREL jmarotat/N ymzgebu/V ^alewu/AUX ./PUNC ……

Appendix B: transliteration from Tigrigna Fidel to Latin characters

First order		Second order		Third order		Fourth order		Fifth order		Sixth order		Seventh order	
ሀ	he	ሁ	Hu	ሂ	hi	ሃ	ha	ሄ	hE	ህ	h	ሆ	ho
ለ	le	ሉ	Lu	ሊ	li	ላ	La	ሌ	lE	ል	l	ሎ	lo
ሐ	He	ሑ	Hu	ሒ	Hi	ሓ	Ha	ሔ	HE	ሕ	H	ሖ	Ho
መ	me	ሙ	Mu	ሚ	mi	ማ	ma	ሜ	mE	ም	m	ሞ	mo
ሠ	^se	ሡ	^su	ሢ	^si	ሣ	^sa	ሤ	^sE	ሥ	^s	ሦ	^so
ረ	re	ሩ	Ru	ሪ	ri	ራ	Ra	ሬ	rE	ር	r	ሮ	ro
ሰ	se	ሱ	Su	ሲ	si	ሳ	Sa	ሴ	sE	ስ	s	ሶ	so
ሸ	xe	ሹ	Xu	ሺ	xi	ሻ	Xa	ሼ	xE	ሽ	x	ሾ	xo
ቀ	qe	ቁ	Qu	ቂ	qi	ቃ	Qa	ቄ	qE	ቅ	q	ቆ	qo
ቐ	Qe	ቑ	Qu	ቒ	Qi	ቓ	Qa	ቔ	QE	ቕ	Q	ቖ	Qo
በ	be	ቡ	Bu	ቢ	bi	ባ	Ba	ቤ	bE	ብ	b	ቦ	bo
ቨ	ve	ቩ	Vu	ቪ	vi	ቫ	Va	ቬ	vE	ቭ	v	ቮ	vo
ተ	te	ቱ	Tu	ቲ	ti	ታ	Ta	ቴ	tE	ት	t	ቶ	to
ቸ	ce	ቹ	Cu	ቺ	ci	ቻ	Ca	ቼ	cE	ች	c	ቾ	co
ኸ	Ke	ኹ	Ku	ኺ	Ki	ኻ	Ka	ኼ	KE	ኽ	K	ኾ	Ko
ኀ	^he	ኁ	^hu	ኂ	^hi	ኃ	^ha	ኄ	^hE	ኅ	^h	ኆ	^ho
ነ	ne	ኑ	Nu	ኒ	ni	ና	Na	ኔ	nE	ን	n	ኖ	no
ኘ	Ne	ኙ	Nu	ኚ	Ni	ኛ	Na	ኜ	NE	ኝ	N	ኞ	No
አ	^e	ኡ	^u	ኢ	^i	ኣ	^a	ኤ	^E	እ	^	ኦ	^o
ከ	Ke	ኩ	Ku	ኪ	ki	ካ	Ka	ኬ	kE	ክ	k	ኮ	ko
ወ	We	ዉ	Wu	ዊ	wi	ዋ	Wa	ዌ	wE	ው	w	ዎ	wo
ዐ	`e	ዑ	`u	ዒ	`i	ዓ	`a	ዔ	`E	ዕ	`	ዖ	`o
ዘ	Ze	ዙ	Zu	ዚ	zi	ዛ	Za	ዜ	zE	ዝ	z	ዞ	zo
ዠ	Ze	ዡ	Zu	ዢ	Zi	ዣ	Za	ዤ	ZE	ዥ	Z	ዦ	Zo
የ	Ye	ዩ	Yu	ዪ	yi	ያ	Ya	ዬ	yE	ይ	y	ዮ	yo
ደ	De	ዱ	Du	ዲ	di	ዳ	Da	ዴ	dE	ድ	d	ዶ	do
ጀ	Je	ጁ	Ju	ጂ	ji	ጃ	Ja	ጄ	jE	ጅ	j	ጆ	jo
ገ	Ge	ጉ	Gu	ጊ	gi	ጋ	Ga	ጌ	gE	ግ	g	ጎ	go
ጠ	Te	ጡ	Tu	ጢ	Ti	ጣ	Ta	ጤ	TE	ጥ	T	ጦ	To
ጨ	Ce	ጩ	Cu	ጪ	Ci	ጫ	Ca	ጬ	CE	ጭ	C	ጮ	Co
ጰ	Pe	ጱ	Pu	ጲ	Pi	ጳ	Pa	ጴ	PE	ጵ	P	ጶ	Po
ፀ	^Se	ፁ	^Su	ፂ	^Si	ፃ	^Sa	ፄ	^SE	ፅ	^S	ፆ	^So
ጸ	Se	ጹ	Su	ጺ	Si	ጻ	Sa	ጼ	SE	ጽ	S	ጾ	So
ፈ	Fe	ፉ	Fu	ፊ	fi	ፋ	Fa	ፌ	fE	ፍ	f	ፎ	fo
ፐ	Pe	ፑ	Pu	ፒ	pi	ፓ	Pa	ፔ	pE	ፕ	p	ፖ	po
ኗ	lWa	ሟ	mWa	ሧ	^sWa	ሷ	sWa	ቿ	xWa	ቈ	qWe	ቍ	qWi
ቋ	qWa	ቌ	qWE	ቍ	qW	ቈ	Qwa	ቌ	QWE	ቍ	QW	ኗ	vWa
ቿ	cWa	ኈ	^hWi	ኋ	^hWa	ኌ	^hWE	ኍ	^hW	ኯ	TWa	ፗ	pWa
ኩ	kWi	ኳ	kWa	ኴ	kWE	ኵ	kW	ዅ	Kwi	ዃ	KWa	ዄ	KWE
ዅ	KW	ዟ	zWa	ዷ	dWa	ጇ	jWa	ጐ	gWe	ጒ	gWi	ጓ	gWa
ጔ	gWE	ጕ	gW	ሗ	HWa	ሯ	rWa	ቈ	Qwe	ቧ	bWa	ቷ	tWa
ኈ	^hWe	ኗ	nWa	ኟ	NWa	ኰ	kWe	ዀ	Kwe	ዧ	ZWa	ቋ	CWa
ፏ	PWa	ፗ	Swa	ፏ	fWa	።	.	፣	,	፤	;	፥	:

Appendix C: Brill tagger learned rules

Sample Lexical Rules

N -> ADJ if the word has suffix „wi‟
N -> NPOSS if the word has suffix „Na‟
V -> VREL if the if the word has prefix „kemz‟
ADJ -> N if the text of the following word is „maHber‟
N -> VREL if the text of word i+2 is 'kemzeyblen'
PREP -> N if the text of the preceding word is 'tgray', and the text of the following word is '^ab'
PREP -> PROND if the text of the preceding word is „dHri‟
ADJ -> N if the text of the following word is „^afeguba^E'
ADJ -> V if the text of words i+1...i+3 is 'hagerawi'
N -> C if the text of the preceding word is 'kab', and the text of the following word is '^tom'
PREP -> C if the text of the following word is '^Seb^Sab'
VREL -> AUX if the text of the preceding word is 'T`na', and the text of the following word is '^abalat'
…….

Sample Contextual rules

ADJ -> ADV if the tag of the preceding word is 'N', and the tag of the following word is 'PRON'
AUX -> V if the tag of word i-2 is 'PREP'
AUX -> VRELC if the tag of word i+2 is 'VRELC'
CARDN -> ADJ if the tag of the preceding word is 'ORDN', and the tag of the following word is 'N'
PREP -> PROND if the tag of the preceding word is „ADJ‟ and the tag of the following word is 'NC'
ADJ -> N if the tag of the preceding word is 'V‟ and the tag of the following word is 'V'

ADJPREP -> VRELPREP if the tag of the preceding word is 'N', and the tag of the following word is 'VI'

ADV -> ADJ if the tag of the preceding word is 'NPREP', and the tag of the following word is 'PREP'

ADV -> C if the tag of the preceding word is 'NP', and the tag of the following word is 'ORDN'

AUX -> V if the tag of word i-2 is 'PREP'

C -> ADV if the tag of the preceding word is 'VREL‟ and the tag of the following word is 'PRONDPREP'

C -> PREP if the tag of the preceding word is 'CARDN‟ and the tag of the following word is 'VREL'
……

Declaration

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

Declared by:

Name: Teklay Gebrgzabiher Abreha

Signature: _____

Date: _____

Confirmed by advisor:

Name: Sebsbie Hailemariam (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa, November, 2010.