ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Development of Part of Speech Tagger for Ge'ez Language

MULATA KEBEDE ABERA

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

October 2017

Addis Ababa University

College of Natural Sciences

MULATA KEBEDE ABERA

Advisor: YAREGAL ASSABIE(PhD)

This is to certify that the thesis prepared by Mulata Kebede, titled: *Development of Part of speech tagger for Ge'ez language* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name **Signature** **Date**

Advisor: **Yaregal Assabie (PhD.)**

Examiner: _____

Examiner: _____

Dedication

I would like to dedicate this paper to my God, Saint Mary and my family.

Acknowledgments

I would like thanks to the **Almighty God** and his Mother **Blessed Virgin Marry** for helping me to realize this work. Many people have contributed in one way or another on this thesis work. First of all, I would like to express my gratitude to my advisor Dr. **Yaregal Assabie** for his guidance and support. I also would like to thank Ato **Pawlos Zemariam** for giving me deep lecture on Ge'ez language and support me anywhere any time. He was always with me during the entire data preparation for this work; without him, it would be impossible to finish this work. I also would like to thank **Ato Negasi Gidey** for helping me on corpus preparation. I would also like to extend my family who always supports me in every way.

Since I cannot list all, I thank all who helped me in the accomplishment of my work directly or indirectly.

Abstract

Part of Speech tagging is the process of assigning part of speech or other lexical class markers to each word in a sentence or literature. Most other tasks and applications heavily depend on it. Much of the research in natural language processing has been dedicated to resource rich languages like English, French and other major European and Asian languages. Among the languages for which POS tagger is developed are Tigrigna, Amharic**,** Kafi-Noonoo, Arabic, Afaan-Oromo, etc. The objective of this research work is to develop POS tagger for Geez using hybrid approach that combines Trigrams 'n' Tags tagger, human written rule, regular expression and unknown word guessing.

Among those diverse statistical taggers, we adopt TnT tagger to the hybrid tagger. Because it enables to the tagger to perform morphological analyzer and maintains several internal frequency distribution and conditional frequency distribution instances based on the training data. Even though TnT is preferred tagger among those statistical taggers for Ge'ez language, still it has shortcoming. TnT does not deal with prefix pattern of unknown words. Regular expression can solve slightly the drawback of TnT tagger. However, the combination of TnT and Regular expression tagger is not still sufficient to get acceptable accuracy, because, Ge'ez language is morphologically complex language and follow free grammar which can follow subject-object-verb, object-subject -verb or subject-verb-object order without change the meaning of the sentence. Consequently, human written rules and unknown word guessing are combined to the hybrid tagger. The hybrid tagger performs better than the individual component of the taggers taken alone.

There was no readymade standard corpus for Ge'ez language. As a result, 26 broad tag sets were identified and 15,154 words from around 1,305 sentences collected from one genre i.e., holy Bible. Then, those words ware manually tagged by Ge'ez language professionals for training and testing purpose. Different experiments are conducted for the three types of taggers namely the TnT tagger, TnT with Regex tagger and Hybrid tagger. We obtained 77.87%, 82.23% and 94.32% performances for TnT tagger, TnT with Regex tagger and Hybrid taggers respectively. As a result, it is possible to conclude that the hybrid tagger performs better than the TnT tagger and TnT with Regex tagger used individually.

Keywords: Ge'ez, POS tagger for Ge'ez, NLP, TnT, Hybrid POS tagger

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and Abbreviations

CRF	Conditional Random Fields
MEM	Maximum Entropy Model
NLTK	Natural Language Toolkit
POS	Part of Speech
Regex	Regular expression
SVM	Support Vector Machine
TnT	Trigrams n Tag

: **Introduction**s a crucial component of our lives. [1]. According to Noam Chomsky [2], Language can be categorized as natural and artificial language.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. NLP is the means for accomplishing different types of tasks and/or applications [3]. Such tasks include POS tagging, named entity recognition (NER), information retrieval (IR), speech recognition, machine translation, question answering etc. [3].

POS tagging is the process of assigning POS like noun, verb, preposition, pronoun, adverb, adjective or other lexical class markers to each word in a sentence or literature. POS tagging is the first step to understanding a natural language. Most other tasks and applications heavily depend on it [4]. The significance of POS (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 neighbor. POS tagging is considered as one of the basic necessary tools. The accuracy of many NLP applications depends on the accuracy of POS tagger [5]. POS tagging can be used in text to speech (TTS), IR, shallow parsing, information extraction (IE), linguistic research for corpora [6], as an intermediate step for higher level NLP tasks such as parsing, semantic analysis, machine translation etc. [6] . POS tagging, thus, is a necessary application for advanced NLP applications in Ge'ez or any other languages.

Much of the research in natural language processing has been dedicated to resource rich languages like English, French and other major European and Asian languages [7] [8] [9]. African languages have, however, received far too little attention. In fact, most are being spoken by few people. Nowadays POS tagger is developed for different languages and it remains an intensive research area for other different languages. Among the languages developed with POS tagger are Tigrigna [10], [11], Amharic [4]**,** Kafi-Noonoo [12], Arabic [9], Afaan-Oromo [13], etc. As to the best of the researcher's knowledge, Ge'ez is the language which does not have developed POS tagger so far.

Ge'ez is the classical language of Ethiopia among the Semitic language family. It is grouped under north Ethiopian Semitic along with Təgrä and Təgrinya [14]. Ge'ez or Ethiopic was the spoken language until the end of the Axum Empire in the ninth century [15]. Today, this language is used only for religious writings and liturgical services in the Ethiopian Orthodox Tewahido Church ,

Eritrean Orthodox Tewahido Church, Ethiopian Catholic Church and the Beta Israel Jewish community.

1.1 Motivation

Ge'ez is the language of many Ethiopian literatures and manuscripts. Several ancient manuscripts, arts, scriptures, heritages, historical, ethical and religious chronicles that can be used as a primary source of knowledge are found in Ge'ez language [16]. The ancient philosophy, tradition, history, knowledge etc. of Ethiopia was written in Ge'ez and there are also different books which are written in this language. These resources can be used as source of philosophy, creativity, knowledge and civilization both to Ethiopia and the rest of the world. To use, keep these resources, and transfer these identities to the next generation, the citizens must understand the semantic and syntactic of these written books/documents. If they do not know the idea in the documents, they will not give any attention for these heritages. If someone who is proposed to conduct a research on issues related to the classical custom, history, politics, tradition, and religion of Ethiopia, must explore the works handed down from the previous generations to the current generation. In addition to this, as the language is the ancestor of other modern Ethio-Semitic languages like Tigrinya and Amharic [15], professionals of these languages should also know the linguistic nature of Ge'ez language to earnestly understand and investigate the nature of these modern ones. To use these resources, one must know the language itself or else these literatures should be translated into either of the currently spoken languages manually, which may take a long time. To solve this problem, studying the nature of the language computationally and finally releasing the resources out with the help of information technology (IT) to be used by everyone of this era is a critical assignment that deserves research. As a result, it is worth conducting research as to develop a POS tagger for Ge'ez to contribute to the complete usage of the language by the new generation [17].

1.2 Statement of the Problem

There are POS taggers that have been developed for international languages like English [18], Arabic [9], Hebrew [19], etc. and Ethiopian languages like Amharic [4], Afaan-Oromo [13], Kafi-Noonoo [12], Tigrigna [10] etc. Rule based, stochastic and hybrid approaches are used to develop POS tagger for various languages. Hybrid approaches is the combination of different tagging techniques such as hidden Markov models (HMMs) and rule based tagging techniques. However, the way they are applied depends on the characteristics of the languages. As a result, these POS

taggers cannot be applied directly to Ge'ez language. On the other hand, to our best knowledge there is no research conducted on POS tagger development for Ge'ez language which is becoming a barrier for research and development works on higher level NLP applications of the language. Hence, the absence of POS tagger system limits researches concerning the NLP of Ge'ez 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 on developing an automatic POS tagger for Ge'ez language is paramount significance.

1.3 Objectives

General Objective

The general objective of this research work is to develop a POS tagger for Ge'ez language.

Specific Objectives

To achieve the above general objective, the research accomplishes the following specific objectives:

- Review techniques for POS tagging.
- Study the structure of Ge'ez language
- Review, analyze and study the basic word category and tag set for Ge'ez language
- Prepare corpus for training and testing the system
- Design and model a POS tagger for Ge'ez language
- Develop a Ge'ez POS tagger prototype
- Test the system performance

1.4 Methods

Literature Review

Literature review has a vital role for identifying the component of POS tagger, comparison of the approaches, detail understanding of problems, finding gaps, identifying methodologies, etc. Moreover, in order to understand the problem books, articles, journals and other publications will be reviewed. In addition, we will review Ge'ez language components such as phonemes, word classification, sentence structure and morphology of Ge'ez language.

Data Source

So far, there is no readymade tag set for Ge'ez language that can be used for this thesis work. Hence, we are to construct a new corpus and discussions will be made with the language experts to prepare tag sets for the language.

Tools

To achieve the research objectives, a relevant tools and methods or approaches will be used. To develop the system, on implementation phase, were selection of the programming language and other related components.

Evaluation

To evaluate the performance of the system, the prototype of the system will be developed and tested with samples tagged sentences. The performance of the system will be measured against the manually prepared corpus.

1.5 Scope and Limitations

The aim of this study is to develop a POS tagger for Ge'ez words based on the corpus into their appropriate category. The corpus that we develop for this thesis work is domain specific corpus, a text corpus that will be collected from a single domain, in this case it is the holy Bible domain only. During the development of the corpus, the tag set use will have meant to give information of words about their word class category but not about the issues like gender, number, tense etc. Moreover, there are limited NLP researches done for Ge'ez language and hence there have been difficulties of using previous works as a reference. However, the text of the corpus will be written language words.

1.6 Application of Results

There are many advantages of developing a 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, text to speech, etc. These applications can be used in different areas of the Ge'ez language. Accordingly, the beneficiaries of this study are:

- Researchers who want to conduct on higher level application of NLP for this language such as spell checker, grammar checker, speech recognition, etc.
- People who want to learn Ge'ez as a second language; it may help them to discover the word categories and grammar construction.
- It can be used as an input for a full parser

- It can be used in text-to-speech system to correct the way of pronunciation
- It can be used for surface linguistic analysis

1.7 Organization of the Rest of the Thesis

This thesis is organized as follows. Chapter 2 presents literature review for POS tagging approaches and Ge'ez language that focuses on the study and assessment of Ge'ez phonemes, word classes and sentence structure. Chapter 3 presents different related works on POS tagger. Chapter 4 presents the design of POS tagger for Ge'ez language. Test results are shown and discussed in Chapter 5. Finally, in Chapter 6, conclusion and future works are presented.

Chapter 1 : Literature Review

2.1 Introduction

POS tagging means assigning grammatical classes i.e., appropriate POS tag to each word in a natural language sentence. Assigning a POS tag to each word of an unannotated text by hand is very time consuming, which results in the existence of various approaches to automate the job. So, automated POS tagging is a technique to automate the annotation process of lexical categories. The process takes a word or a sentence as input, assigns a POS tag to the word or to each word in the sentence, and produces the tagged text as output. POS tags are also known as word classes, morphological classes, or lexical tags. The significance of these is the large amount of information they give about a word and its neighbors. In this chapter, we present structure of the Ge'ez language and different approaches for POS tagging.

2.2 Ge'ez Language

As we have discussed in background of the study of chapter one, Ge'ez language, also spelled Geez, is the classical language of Ethiopia within the Semitic language family. It is grouped under north Ethiopian Semitic along with Təgrä and Təgrinya [14]. It also belongs the South Arabic dialects and Amharic. Both Ge'ez and the related languages of Ethiopia are written and read from left to right, in contrast to the other Semitic languages. Extinct as a vernacular language, Ge'ez is the ancestor of the modern Tigrinya and Tigré languages of Eritrea and Ethiopia. Ge'ez or Ethiopic was the spoken language until the end of the Axum Empire in the ninth century [15]. Today, this language is used only for religious writings and liturgical services in the Ethiopian Orthodox Tewahido Church, Eritrean Orthodox Tewahido Church, Ethiopian Catholic Church and the Beta Israel Jewish community.

Some centuries before the Christian era, a Semitic people who spoke a pure Semitic language, were used Ge'ez language for communication which they called lisane Ge'ez "the tongue of the free [20]. The language commonly called Ethiopic is the language in which the inscriptions of the kings of the ancient Aksumitic (Axumite) empire and most of the literature of Christian Abyssinia are written. It is called lesana Ge'ez, "the tongue of Ge`ez," by the Abyssinians themselves, most probably because it was originally the dialect of the Ge`ez tribe, who in antiquity must have dwelt in or near Aksum (Axum) [21] . Their language, which we now call Ethiopic, remained the spoken

tongue till the beginning of the seventeenth century, when it was superseded by Tigre, Tigrigna, and Amharic. Since then, however, it has persisted as the language of the Church and of literature.

The Ethiopic tongue is more closely related to Arabic than to any other Semitic language, but its affinities to Assyrian and Hebrew are also close [22]. Most of extant Ethiopic literature, except for some inscriptions, has been handed down in manuscript form, and is ecclesiastical in character, the chief being versions of the books of the Old and the New Testaments. There is a considerable amount of theological, poetical (religious), and liturgical literature, and some historical, chronological, legal, mathematical, and medical material. Although the people of Europe first became interested in the language and literature of Ethiopia as early as the sixteenth century, very little was done until the time of Ludolfus who published a Grammatica Aethiopica in 1661 [23], and a Lexicon Aethiopico-Latinum a few years later. Ludolfus may well be called the father of Ethiopic studies. An unfortunate lull followed his efforts, and the study of Ethiopic was neglected until the time of Hupfeld in 1825 [24]. Since then such men as Tuch, Ewald, and especially Dillmann [25], Praetorius, Littmann and Wolf Leslau [14] have done good work.

2.2.1 Ge'ez Phonemes

Phoneme represents the smallest unit of natural languages. The Ge'ez letters can be called as syllabary. The letters can also be consonants that are not vocalized that is to say in the sixth order. In this concept, the letters can also be called alphabets.

The Alphabet

Ge'ez is written with Ethiopic or the Ge'ez abugida, a script that was originally developed specifically for this language. In languages that use it, such as Amharic and Tigrinya, the script is called Fidäl, which means script or alphabet. Formerly they were written from right to left like Hebrew, Aramaic, and Arabic [22]. Latter it read/write from left to right. The Ge'ez script has been adapted to write other languages, usually ones that are also Semitic. The most widespread use is for Amharic in Ethiopia and Tigrinya in Eritrea and Ethiopia. It is also used for Sebatbeit, Agew and most other languages of Ethiopia. In Eritrea it is used for Tigre, and it is often used for Bilen, a Cushitic language. It has twenty-six consonants [26] and adding obligatory vocalic diacritics to the consonantal letters. The diacritics for the vowels, u, i, a, e, ə, o, were fused with the consonants in a recognizable but slightly irregular way, so that the system is laid out as a syllabary, the letters can also be consonants that are not vocalized that is to say in the sixth order. The original form of

the consonant was used when the vowel was ä (/ə/), the so-called inherent vowel. The resulting forms are shown below in their traditional order. For some consonants, there is an eighth form for the diphthong -wa or -oa, and a ninth for -yä. It also uses four symbols for labialized velar consonants, which are variants of the non-labialized velar consonants. Unlike the other consonants, these labiovelar ones can only be combined with five different vowels. Additionally, it has also, its own symbols for representing numbers such as ፩ /one, ፪/two, ፫/three, ፬/four, ፭/five, ፮/six, ፯/seven, ፰/eight, ፱/nine, ፲/ten, ፳/twenty, ፴/thirty, ፵/forty, ፶/fifty, ፷/sixty, ፸/seventy, ፹/eighty, ፺/ninety, ፻/hundred etc.

Finally, it has also punctuation [27], much of it modern, includes,※ section mark,፡ word separator, ። full stop (period), ፣ comma, ፤ colon, ፥ semicolon, ፦ preface colon, ፧ question mark and ፨ paragraph separator.

2.2.2 Word Classification of Ge'ez

Like in other natural languages, Ge'ez also have word class in the grammar. According to the class, they play in a sentence. As stated by Alemayohu [26], the main word classes in Ge'ez are Noun, Adjective, Pronoun, Verb, Adverb, Preposition Conjunctions and Interjection.

Noun

Nouns are a POS which refer to person, place, thing, animal or idea. Ge'ez nouns possess gender, number, and case. Ge'ez nouns fall into four distinct classes [26]:

Common Nouns*:* Common refer to a person, place, or thing in a general sense which are with no special characteristics, e. g mother, lion, love, city etc. All are common nouns because they name a person, animals, thing, or place. Whereas Addis Ababa is a proper noun because it signifies a specific city in Ethiopian.

Proper Nouns: Proper nouns used to identify unique individuals, things, events, or places. A proper noun is a noun (or nominal content word) that is the name (or part of the name) of a specific individual, place, or object. Unlike English Ge'ez does not use capitalize for proper nouns to show their distinction from common nouns.

Collective Nouns: In general, collective nouns are nouns that refer to a group of something in a specific manner. Often, collective nouns are used to refer to groups of animals, a class of students, an army of soldiers, a choir of singers etc.

Abstract Nouns: More ethereal, theoretical concepts use abstract nouns to refer to them. Concepts like freedom, love, power, and redemption are all examples of abstract nouns. They support us for our freedom. All you need is love. We must fight the power. In these sentences, the abstract nouns refer to concepts, ideas, philosophies, and other entities that cannot be concretely perceived.

Pronoun

Pronouns are words that stand in the place of nouns in order to avoid unnecessary repetition. In Ge'ez, because they stand in for nouns, pronouns also have gender, case, and number. In addition, pronouns can be divided into classes according to their function in the sentence. Most of these classes can in turn be subdivided into subclasses. The detail description of pronouns is presented in Table 2.1. Ge'ez pronouns fall into two classes:

Independent Pronoun: One that stands by itself and which represents a specific person or object. As such, independent pronouns possess person, gender, case and number and used in nominative position.

Suffixed pronouns: used as possessive pronouns for nouns, as objects for verbs, and with prepositions and certain particles. The suffixed personal pronouns are also employed to derive independent possessive pronouns and reflexive pronouns.

Table 1.1 Independent and suffixed pronouns

Pronoun	Class
Independent Personal Pronoun	One that stands by itself and which represents a specific person or object. As such, independent pronouns possess person, gender, and number. E.g., አነ/ene/I, ንሕነ/nəḥne/We, አንተ/ente/You (m.i), አንትሙ/entəmu/Y'all (m.), አንቲ/enti/You, አንትን/entən /Y'all (f.), ውእቱ/wə'ətu, እማንቱ/əmantu/they (f.) etc.
Reflexive	Indicates that the action of a verb is directed toward its own subject. Reflexive pronouns are formed with ርእስ /rais/ 'head' or, less frequently ነፍስ/nafs/'soul' and the suffixed personal pronouns: e.g., ቤዘወነ: ርእሱ {bezewene: raisu} (lit. 'he himself hath saved us').
Relative	The relative pronouns originated from the demonstrative pronouns: እንተ/ente, እለ/Ele and introduces a relative clause, which serves as an adjective modifying the antecedent of the relative pronoun; relative pronouns can be either definite or indefinite.

Interrogative	Used to make asking questions easy. There are different interrogative pronouns. Each one is used to ask a very specific question or indirect question. Some, such as መኑ/menu, for who, መኑ/menu, ምነ/mine/ for what, አይ/Ay, አያት/Ayat/, አየ/Aye, አያተ/Ayate for which, ሚ/mi/ for what? etc.
Demonstrative	The demonstrative pronouns, like the nouns, are marked for gender and number. This demonstrative is usually prefixed or affixed to the word next to it, e. g. ዝሕዝበ /zhizb 'this people 'or ዝንቱ፡ብእሲ፡ውእቱ፡ንጉሥ። zəntu bə'əsi wə'ətu nəguš 'this man is a/the king.'

Adjective

An adjective is a word used to modify a noun and to specify their properties or attributes. In Ge'ez, an adjective must agree with the word it modifies in gender, case and number. Adjectives can also exist in different states. In addition, adjectives can be divided into classes according to their function in the sentence. Table 2.2 describes different class of adjective.

Table 1.2 Adjective class

Adjective Class	Sentence Function
Cardinal	Describes the numbers አሀዱ/ahadu/*"one",* ክልኤቱ/kilEtu/*"two",* ሰለስቱ/selestu*"three",* etc.
Demonstrative	Specifies or points out the person or thing referred to, such as ዝ/*z*, ዘ/*ze*, ዝዝ/*zez*, ለዝ/*lez*, ዛ/*za*, ዛዘ/*zaze*, ለዛ/*leza*, etc. which is to refer 'this' or '*that'.*
Indefinite	Like an indefinite pronoun, such as *"many"* or *"some".*
Intensive	Indicates emphasis such as *"all"* or *"each"*
Interrogative	Implies a question, such as መኑ/*Menu*/ "*what"*
Numeral	Relates to numerical symbols (like *" ፩/1, ፪/2, ፫/3, ፬/4, ፭/5, ፮/6"* etc., as opposed to አሀዱ/ahadu/*"one",* ክልኤቱ/kilEtu/*"two".*
Ordinal	Expresses the numbers in ordinal format e.g., ቀዳማይ/qedamy/*"first",* ካልአይ/kaalay *"second",* ሳልሳይ/salsay*"third"* etc.

Possessive	Indicates possession e.g., የ/ye, ከ/ke, ክሙ/kmu, ሁ/hu, ሙ/mu/, ሆሙ/homu for *"my", "your" etc.*
Proper name	Derived from the name of a person or place e.g., ኢትዮጲያዊ/*Ethiopian*.
Relative	Has qualitative indefiniteness; *"such as," "as many as,"* and *"whatever"*.
Verbal	Derived from a verb

Verb

A verb is a word that expresses action or a state of being. In Ge'ez, the subject of the verb is implicitly expressed by the verb itself. Consequently, Ge'ez verbs have both person and number in addition to having tense, voice, and mood. Verbs are often associated with grammatical categories like tense, mood, aspect and voice, which can either be expressed inflectionally or using auxiliary verbs or particles. Many 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. Formation of verb in Ge'ez has three stages, which the verb must pass through;

1. Stem-formation;
2. Tense- and mood-formation;
3. Formation of persons, genders and numbers.

Adverb

Adverbs are words that typically modify verbs for such categories as time, place, direction or manner. In Ge'ez, adverbs exist in one of three degrees (which are included under the class category in Accordance) or as a negative.

Examples

- ህየ/Hiye 'there'
- ለፊ/lefie' thence', 'since'
- ሕቀ /Hiqe 'little'
- Interrogative adverbs: ሁ, often in composition, e. g. ቦሁ
- Totality adverbs: ፈድፋደ/fedfade/'much'or 'excessively'
- Negative adverbs: ኢ/i always prefixed

Preposition

A preposition is a word that indicates the relationship of a substantive (known as the object of the preposition) to a verb, an adjective, or another substantive. Prepositions can be free-standing or prefixed to another POS. The meaning of a preposition can vary depending on the case of its object. Some of the most frequent Ge'ez prepositions are:

በ /be/'in'; ለ/la/ 'to', 'toward'; እምነ/əmənna/'from','out of' etc.

Conjunction

Conjunctions are words that link other words, phrases or clauses. In Ge'ez language there are two types of conjunctions: coordinating and subordinating conjunctions.

Coordinating conjunctions coordinate or join two or more sentences, main clauses, words, or other POS which are of the same syntactic importance. Also, known as coordinators, coordinating conjunctions are used to give equal emphasis to a pair of main clauses. In Ge'ez Coordinating conjunctions are divided in to independent and dependent. Among the dependent, some are suffixed, others are affixed. A coordinating conjunction joins two identically constructed grammatical elements, and is belongs to one of the following subclasses:

Table 1.3 Coordinating Conjunction

Coordinating Conjunction Subclass	Sentence Function
Adversative	Expresses antithesis or opposition, such as አላ/Ala/ 'but'
Continuative	Expresses continuation, such as እንከ / enke/ 'then.'
Copulative	Connects coordinate words or clauses, such ሂ/hi/'and' also, always suffixed.
Disjunctive	Expresses contrast or opposition, such as ዳእሙ/da-emu/' but, however.'
Explanatory	Introduces an explanation, such as እምዘ/emze/'since.'

Subordinating conjunctions are conjunctions that links constructions by making one of them a constituent of the other. The subordinating conjunction typically marks the incorporated constituent which has the status of a (subordinate) clause. It introduces a dependent clause, and belongs to one of the following subclasses.

Table 1.4 Subordinating Conjunction

Subordinating Conjunction Subclass	Sentence Function
Causal	Expresses a cause or reason, such as ለ/le/’for.’
Concessive	Expresses a concession or admission, such as እመኒ/emeni/, እመሂ/emehi/, ለእመኒ/le-emeni/, ለእመሂ/le-emehi/’though’ or ‘although.’
Conditional	Expresses a condition, such as እመ/eme/, አመ/ame/’if’’ or "unless."
Temporal	Expresses time, such as እምአመ/emame/’when, while,’

Interjection

Expresses strong feeling, emotion, or surprise. They are often capable of standing on their own. Examples

- ሐሰ/hasse/፣ ሓሰ /hasse/ ‘sign of aversion. ‘
- ሐዊሳ/hawisa/፣ሓዌሳ /hawisa/’sign of joy.’
- ለይልየ/leylye/፣ሌልየ/leylye/’woe is me.’
- ሶ/so/’sign of request, please!’

2.2.3 Structure of the Ge’ez Sentence

Ge'ez language is free in word order of the sentence. It may take subject- verb-object (SVO), object- verb- subject (OVS) or verb-subject- object (VSO) [28]. Most of the time the order of word class depends on the type of verb that used. Like other Semitic languages, only Biblical Ge'ez had two distinct sets of verbs form /aspects, called the imperfect and the perfect. The imperfect forms were used frequently for most purposes, while the perfect forms were used only occasionally for a few purposes. Importantly, the imperfect forms normally required VSO word order, the ordinary word order of the language is the perfect, a marked form, usually required a marked word order, SVO. Here are two examples from the book of the Gospel:

- ወባረኮ እግዚአብሔር ለአብርሃም በኵሉ /WeBareko Egziabher leAbriham Bekulu/ and the LORD had blessed Abraham in all things. Genesis 24:1
- እግዚአብሔር ባረኮ ለአብርሃም ጥቀ /Egziabher Bareko leAbriham Tiqe/ the LORD blessed to Abraham greatly. Genesis 24:35

As in any other language, Ge’ez has two chief members of the sentence namely subject and predicate [25], may be extended into larger groups of words.

The Subject

Every sentence, which is not imperfect, must contain a subject. Such subject is usually a substantive or a pronoun representing a substantive; but it may also be an adjective if it is invested with the force of a Substantive, or even an Adverb, when through the stimulus of speech, the adverb is raised to the position of a noun-substantive. An entire sentence even may take the place of subject, particularly a relative or a conditional sentence, example. የእሰለኒ ፡ ዘረኩብኩ ፡ ሞገሰ” it is enough for me that I have found favor", just as in other languages.

The Predicate

The predicate of a sentence is usually a verb or an adjective (or participle). Certain adjectives, when used as predicates, are in all cases, or at least in certain cases, supplemented by a suffix. Those adjectives and participles also, which are formed by periphrasis with the relative pronoun.

2.2.4 Ge’ez Morphology

Morphology is a term based on the Greek words morphe (=form/structure) and logie (=account/study). In fact, the term can apply to any domain of human activity that studies the structure or form of something. In linguistics, morphology is the study of words, how they are formed, and their relationship to other words in the same language [29]. There are two types of complexity of word-structure: one is due to the presence of inflections and another due to the presences of derivational elements. Both operations add extra elements to what is known as the base [28].

Derivation refers to word formation processes such as affixation, compounding and conversion. Derivational processes typically induce a change in the lexical category of the item they operate on and even introduce new meanings. Inflection includes the grammatical categories/markers for number, gender, case, person, tense, aspect, mood and comparison. It is defined as “a change in the form of a word to express its relation to other words in the sentence”.

Based on rules, morphology can be classified into three main categories: derivational, inflectional, and compounding [30]. Derivational Morphology is a morphology concerned with the way in

which words are derived from morphemes through processes such as affixation or compounding. This derivation process usually changes the POS category.

Inflectional morphology deals with combination of a word with a morpheme, usually resulting in a word of the same class as the original stem, and serving the same syntactic function. They do not change the POS category but the grammatical function.

Compounding morphology is the process of forming a new word through combining two or more words. Compounding is a process of word formation that involves combining complete word forms into a single compound form.

In Ge'ez language, among POS: nouns, verbs and adjectives are the most frequent inflectional morphology. The reaming POS tags are less frequent. Hence, we will focus on those most frequent inflectional morphologies.

Ge'ez nouns Derivation: Ge'ez nouns can be derived from verbal roots by infixing vowels between consonants, stems by prefixing or suffixing bound morphemes, stem-like verbs by suffixing the bound morpheme, nouns by suffixing bound morphemes and compound words Dillmann [25].

Ge'ez nouns Inflection: In Ge'ez language, nouns can be inflected for gender, number, definiteness, and case. In Ge'ez language, there are three genders: masculine, feminine, and common. The masculine has no special termination. However, the feminine has the termination ት(t) e.g., ዐመት(amet) /year, ቁላት (qwelat) /valley or sometimes it has no ending. There are two numbers: singular and plural. The plural is formed in two ways:

- Strong plural: formed by means of the termination ን (n) for the masculine, e. g. ነግድ(negd) traveler, p. ነግዳን (negdan) and ት (t) for the female, e.g., ዐመት(amet)/servant, and for plural ዐመታት(ametat)/ servants.
- Broken plural: formed by vocal modification and by prefixes and suffixes. E.g., ቀተል (qetel)/killed, ቀትል(qetl), ቅትል (qtl) became አቅታል (aqtal).

In Ge'ez nouns, there are four cases: nominative, vocative, genitive, and accusative. The nominative and genitive have no distinct termination. The vocative is the same as the nominative, or is the nominative with a prefixed or suffixed አ(a), e. g. አግብር(agbr) /servant. The accusative is differing from the nominative because unlike nominative, it is formed by vocal change. E.g., the nominative መስል(mesl) become as ምስለ(msle)/image as accusative. In general, Table 2.5 shows summarized form of noun Inflection by using single word Camel.

Table 1.5 Examples of Morphological Changes for Ge'ez Noun፡ ግመል

No	Ge'ez	Transliteration	Meaning
1	ግመል	**Gmel**	Camel
2	ግመሉ	**Gmel**u	him Camel /the Camel
3	ግመላ	**Gmel**a	her Camel
4	አግማል	A**gm**a**l**a	Camels
5	ለአግማሊሁ	le-a**gm**a**l-ihu**	for him Camels
6	ለአግማሊከ	le-a**gm**a**l-ike**	for your Camels
7	ለአግማሊከኒ	le-a**gm**a**l-ike-ni**	also for your Camels
8	በአግማሊሆን	be-a**gm**a**l-ihon**	By their Camels
9	በአግማል	be-a**gm**a**l**	By Camels
10	አግማለ	a**gm**a**l-e**	The accusative case of Camels
11	አግማሊሁ	a**gm**a**l-ihu**	for him Camels
12	እምአግማለ	em-a**gm**a**l-e**	From Camels
13	ወለአግማሊከኒ	we-le-a**gm**a**l-ike-ni**	and also for your Camels
14	ለአግማሊሃ	le-a**gm**a**l-iha**	for her Camels
15	አግማሊየ	a**gm**a**l-iye**	My Camels
16	ኢአግማሊሆን	i-a**gm**a**l-ihon**	not their Camels

Derivation Verb: Ge'ez verbal stems (from which various forms of verbs are formed) can be derived from verbal roots by affixing vowels, repeating penultimate consonants and affixing vowels. Can be also derived from verbal stems by affixing morphemes.

Inflection Verb: Ge'ez verbs are marked for person, gender, number, case, tense/aspect and mood. We use example of derivation and inflection of root verb ፍቅድ (fqd) that is adapted from the work of Desta [31] shown in Table 2.6.

Table 1.6 Examples of Morphological Changes for Ge'ez verb፣ fqd

No	Ge'ez	Transliteration	Meaning
1	ፈቀደ	feqede	He liked
2	ፈቀደኒ	feqedeni	He liked me
3	ፈቀደነ	feqedene	He liked us
4	ፈቀደከ	feqedeke	He liked you(2psm)
5	ፈቀደክሙ	feqedekmu	He liked you(2ppm)
6	ፈቀደኪ	feqedeki	He liked you(2psf)
7	ፈቀደክን	feqedekn	He liked you(2ppf)
8	ፈቀዶ	feqedo	He liked him
9	ፈቀዶሙ	feqedomu	He liked them(3ppm)
10	ፈቀዳ	feqeda	He liked her
11	ፈቀዶን	feqedon	He liked them(3ppf)
12	አስተፋቀደ	Estefaqede	He caused others to like each other
13	አስተፋቀዳ	astefaqda	He caused her to be liked with
14	አፍቀደ	afqede	He caused somebody to be liked
15	አፍቀደኒ	afqedeni	He caused me to be liked
16	አፍቀደነ	afqedne	He caused us to be liked
17	አፍቀደከ	afqedeke	He caused you(2psm) to be liked
18	አፍቀደክሙ	afqedekmu	He caused you(2ppm) to be liked
19	አፍቀደኪ	afqedeki	He caused you(2psf) to be liked
20	አፍቀደክን	afqedekn	He caused you(2ppf) to be liked
21	አፍቀዶ	afqedo	He caused him to be liked
22	አፍቀዶሙ	afqedomu	He caused them(3ppm) to be liked

23	አስተፋቀዶሙ	astefaqedomu	He caused them(3ppm) to like each
24	ተፈቅደ	tefeqde	He is liked by
25	ተፋቀደ	tefaqede	He is liked with somebody
26	አፍቀዳ	afqeda	He caused her to be liked
27	አፍቀዶን	afqedon	He caused them(3ppf) to be liked
28	አስተፋቀደኒ	astefaqedeni	He caused me to like with others
29	አስተፋቀደነ	astefaqedene	He caused us to like with others
30	አስተፋቀደከ	astefaqedeke	He caused you(2psm) to like with
31	አስተፋቀደክሙ	astefaqedekmu	He caused you(2ppm) to like with
32	አስተፋቀደኪ	astefaqedeki	He caused you(2psf) to like with
33	አስተፋቀደክን	astefaqedekn	He caused you(2ppf) to like with
34	አስተፋቀዶ	astefaqedo	He caused him to like with others
35	አስተፋቀዶን	astefaqedon	He caused them(3ppf) to like with

Ge'ez Adjectives Derivation: In Ge'ez, like verbs and nouns, adjectives also have derivational and inflectional morphology. Ge'ez adjectives can be derived from verbal roots by infixing vowels between consonants, nouns by suffixing bound morphemes, stems by suffixing bound morphemes and compound words of nouns and adjectives. Ge'ez Adjectives can be marked for number by affixation of morphemes or repetition of consonants, definiteness by affixation of morphemes or vowels based on number, gender, and/or ending of the adjective, gender by affixation of the morpheme and object case by affixation of the morpheme.

2.3 Approaches to POS Tagging

There are different approaches to the problem of assigning each word of a text with a parts-of-speech tag, which is known as POS tagging [32] . The most common ones are rule-based, stochastic, artificial neural network and hybrid approaches. Figure 2.1 [32, 33, 34] demonstrates the classification of different POS tagging approaches.

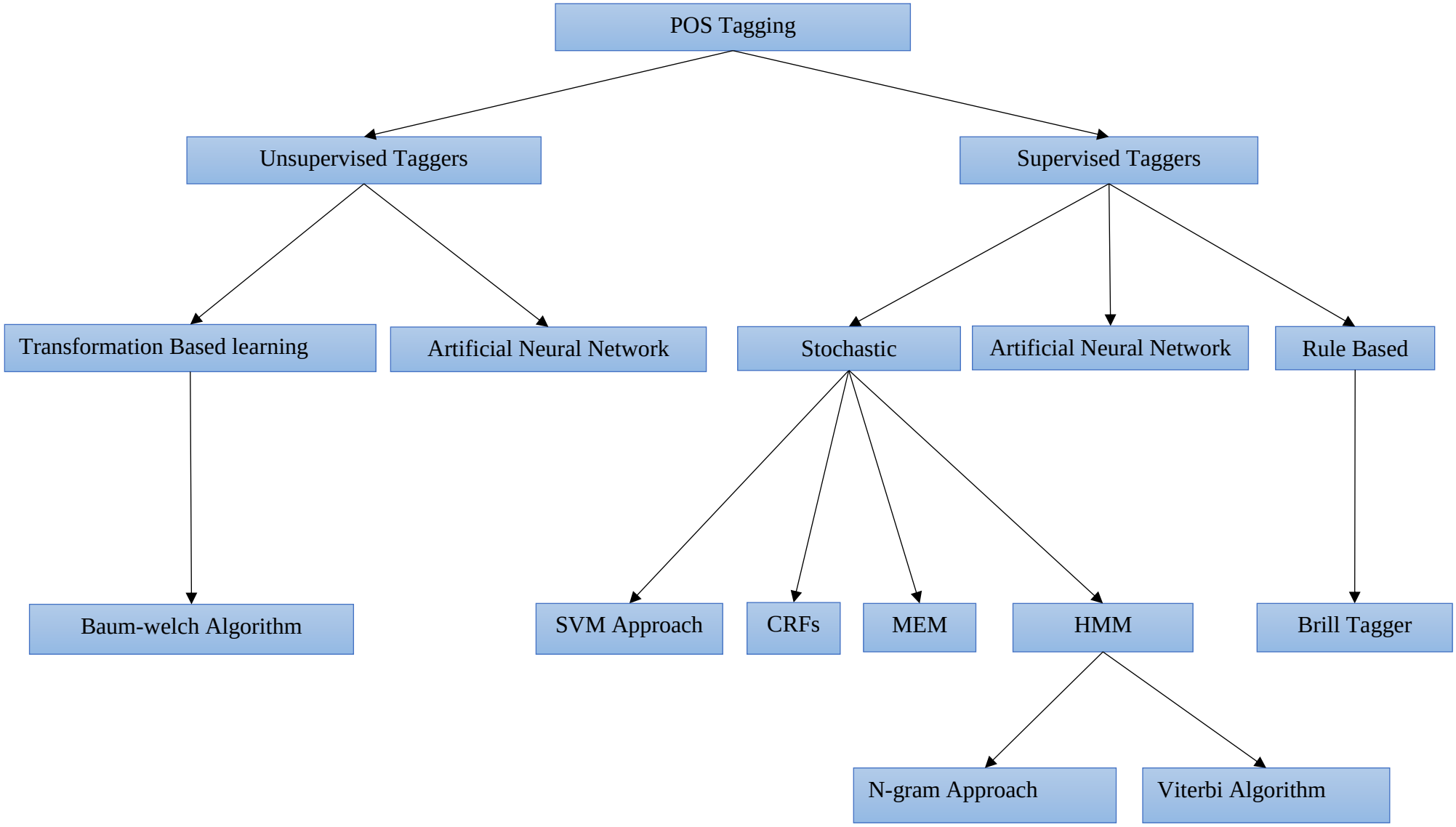

Figure 1.1 Classification of POS tagging models

2.3.1 Supervised POS Tagging

The supervised POS tagging models require a pre-tagged corpus which is used for training to learn information about the tag-set, word-tag frequencies, rule sets etc. The performance of the models generally increases with the increase in size of the pre-tagged corpora [33].

Stochastic: The stochastic approach finds out the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the unannotated text. A stochastic approach required a sufficient large sized corpus and calculates frequency, probability or statistics of every word in the corpus. This model is based on various models such as hidden Markov model (HMM), maximum likelihood estimation, N-grams, maximum entropy, support vector machines and conditional random fields [32].

HMM: A Hidden Markov Model (HMM) is a statistical model in which the system modeled is thought to be a Markov process with the unknown (hidden) parameters.

The basic idea of HMM is to compute or determine the most likely tag sequences. After collecting statistical data of the tagged corpus from tag analyzer, the tagger is activated on the test set which is already tokenized by the tokenizer [34]. The tagger employs a sentence based approach rather than a word-based approach, i.e., first all the possible tags for the words and the word sequences in the sentence are determined, and then the combination of the tags with the highest probability for the whole sentence is selected. The best tag can be determined by it for a word by finding out the probability that it occurs with the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes. These models are termed as unigram, bigram and trigram.

A HMM lets us handle both
Observed events (like the words in a sentence) and
Hidden events (like POS tags).
HMM represented as a set {Q, A, O, B, q} [35]. Where
Q = q1, q2 . . ., qn: a set of N states
A = a11, a12 . . ., an1 . . . ann: a transition probability matrix A, representing the probability of moving from state i to state j, such that $\Sigma_{j=1}^{n} a_{ij} = 1 \; \forall i$
O= o1, o2 . . . oT: a sequence of T observations, each one drawn from a vocabulary V = v1, v2, . . . vT

B = bi(ot): A sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation ot being generated from a state i

q0, qF: a special start state and final state that are not associated with observations, together with transition probabilities a01a02 . . . a0n out of the start state and a1F, a2F, . . ., anF into the final state.

The basic equation of HMM Tagging

Probability is the basic principle behind HMM. The intuition behind all stochastic taggers is simple generalization of the "pick the most-likely tag for this word". For a given sentence or a word sequence, HMM tagger chooses the tag sequence that maximizes: P (word | tag) * P (tag | previous n tags). The goal of HMM decoding is to choose the tag sequence that is most probable given the observation sequence of n words w_1^n:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \qquad (1)$$

Where hat (^) means "our estimate of the best probable tag sequence", argmax f(x) means "the x such that f(x) is maximized" it maximizes our estimate of the best tag sequence.

By using Bayes' rule to instead compute:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \frac{P(w_1^n | t_1^n) P(t_1^n)}{P(w_1^n)} \qquad (2)$$

Furthermore, we simplify Eq. 2 by dropping the denominator $P(w_1^n)$:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n | t_1^n) P(t_1^n) \qquad (3)$$

We can drop the denominator: it does not change for each tag sequence; we are looking for the best tag sequence for the same observation, for the same fixed set of words.

HMM taggers make two further simplifying assumptions. The first is that the probability of a word appearing depends only on its own tag and is independent of neighboring words and tags:

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^{n} P(w_i | t_i) \qquad (4)$$

The second assumption, the bigram assumption, is that the probability of a tag is dependent only on the previous tag, rather than the entire tag sequence;

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1}) \qquad (5)$$

Plugging the simplifying assumptions from Eq. 5and Eq. 4 into Eq. 3results in the following equation for the most probable tag sequence from a bigram tagger.

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}} \qquad (6)$$

Estimating probabilities

In HMM tagging, rather than using the full power of HMM estimating probabilities (EM) learning, the probabilities are estimated just by counting on a tagged training corpus. The tag transition probabilities P ($t_i|t_{i-1}$) represent the probability of a tag given the previous tag. The maximum likelihood estimate of a transition probability is computed by counting, out of the times we see the first tagin a labeled corpus, how often the first tag is followed by the second.

$$P(t_i|t_{i-1}) = C(t_i - 1, t_i) \,/\, C(t_i - 1) \qquad (7)$$

The emission probabilities, $P(w_i|t_i)$, represent the probability, given a tag, that it will be associated with a given word. The maximum likelihood estimation (MLE) of the EM is

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)} \qquad (8)$$

The computation of this formula is very expensive as all possible tag sequences are required to be checked in order to find the sequence that maximizes the probability. So, a dynamic programming approach known as the Viterbi Algorithm is used to find the optimal tag sequence [36].

N-Gram: N-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram“; size 3 is a "trigram".

The Unigram tagger is a simple statistical tagging algorithm. For each token, it assigns the tag that is most likely for that token. For example, it will assign the tag ‘adj’ to any occurrence of the word ‘frequent’, since ‘frequent’ is used as an adjective (e.g., a frequent word) more often than it is used as a verb (e.g., I frequent this cafe).

The Bigram tagger works in exactly the same way as the Unigram Tagger, the only difference is that it considers the context when assigning a tag to the current word. When training, it creates a frequency distribution describing the frequencies with which, each word is tagged in different

contexts. The context consists of the word to be tagged and the tag of the previous word. When tagging, the tagger uses the frequency distribution to tag words by assigning each word, the tag with the maximum frequency given the context.

For describing trigram model for POS tagger, to perform POS tagging to determine the most likely tag for a word, given the previous two tags. So, if t_1, t_2 …t_n are tag sequence and w_1, w_2…w_n are corresponding word sequence then the following equation explains this fact

$$P\left(\frac{ti}{wi}\right) = P\left(\frac{wi}{ti}\right).P\left(\frac{ti}{ti-2, ti-1}\right) \qquad (9)$$

Where $\mathbf{t_i}$ denotes tag sequence and $\mathbf{w_i}$ denote word sequence. $\mathbf{P\ (w_i/t_i)}$ is the probability of current word given current tag. Here, $P\left(\frac{ti}{ti-2, ti-1}\right)$ is the probability of a current tag given the previous two tags. This provides the transition between the tags and helps capture the context of the sentence. These probabilities are computed by the following equation.

$$p\left(\frac{ti}{ti-2, ti-1}\right) = \frac{freq(ti-2, ti-1, ti)}{freq(ti-2, ti-1)} \qquad (10)$$

Each tag transition probability is computed by calculating the frequency count of two tags which come together in the corpus divided by the frequency count of the previous two tags coming in the corpus.

TnT: TnT tagger is proposed by Thorsten Brants, written as very efficient statistical POS tagger and trainable on different languages [37]. TnT is a stochastic HMM tagger based on trigram analysis, which uses a suffix analysis technique based on properties of words like, suffices in the training corpora, to estimate lexical probabilities for unknown words that have the same suffices. Linear interpolation is the main paradigm used for smoothing and the weights are determined by deleted interpolation.

$$arg \max_{t_1 \cdots t_T} \left[\prod_{i=1}^{T} P(t_i | t_{i-1}, t_{i-2}) P(wi | t_i) \right] P(t_{T+1} | t_T) \qquad (11)$$

The set of possible tags for a given word is derived from the training data. It is the set of all tags that exact word has been assigned.

TnT is an implementation of the Viterbi algorithm and it uses second order Markov models for POS tagging. The states of the model represent tags; outputs represent the words. Transition

probabilities depend on the states, thus pairs of tags. Output probabilities only depend on the most recent category. TnT is trained with different smoothing methods and suffix analysis. The parameter generation component trains on tagged corpora. The system uses several techniques for smoothing and handling of unknown words. The tagger is implemented using Viterbi algorithm for second order Markov models. TnT can be used for any language. Adapting the tagger to a new language, new domain or new tag set is very easy.

To handle the unknown words, suffix trie and successive abstraction are used. There are two types of file formats used in TnT, untagged input and the tagged input for tagger. The advantage of this tagger is, first its speed, which is important both for fast tuning cycle and dealing with large corpora. Second its suffix guessing algorithm that is triggered by unseen words. From the training set TnT builds a trie from the endings of words appearing less than n times in the corpus, memorizes the tag distribution for each matrix. The third advantage of this approach is the probabilistic weighting of each label, however, under default settings the algorithm proposes a lot more possible tags than a morphological analyzer would.

The architecture of the TnT tagger for this thesis work is given in Figure 2.2. As it is seen in the Figure 2.2, the tagger consists of three main parts; TnT trainer, TnT tagger and TnT tester. TnT trainer accept annotated corpus as an input. Then it goes through a tokenization process.

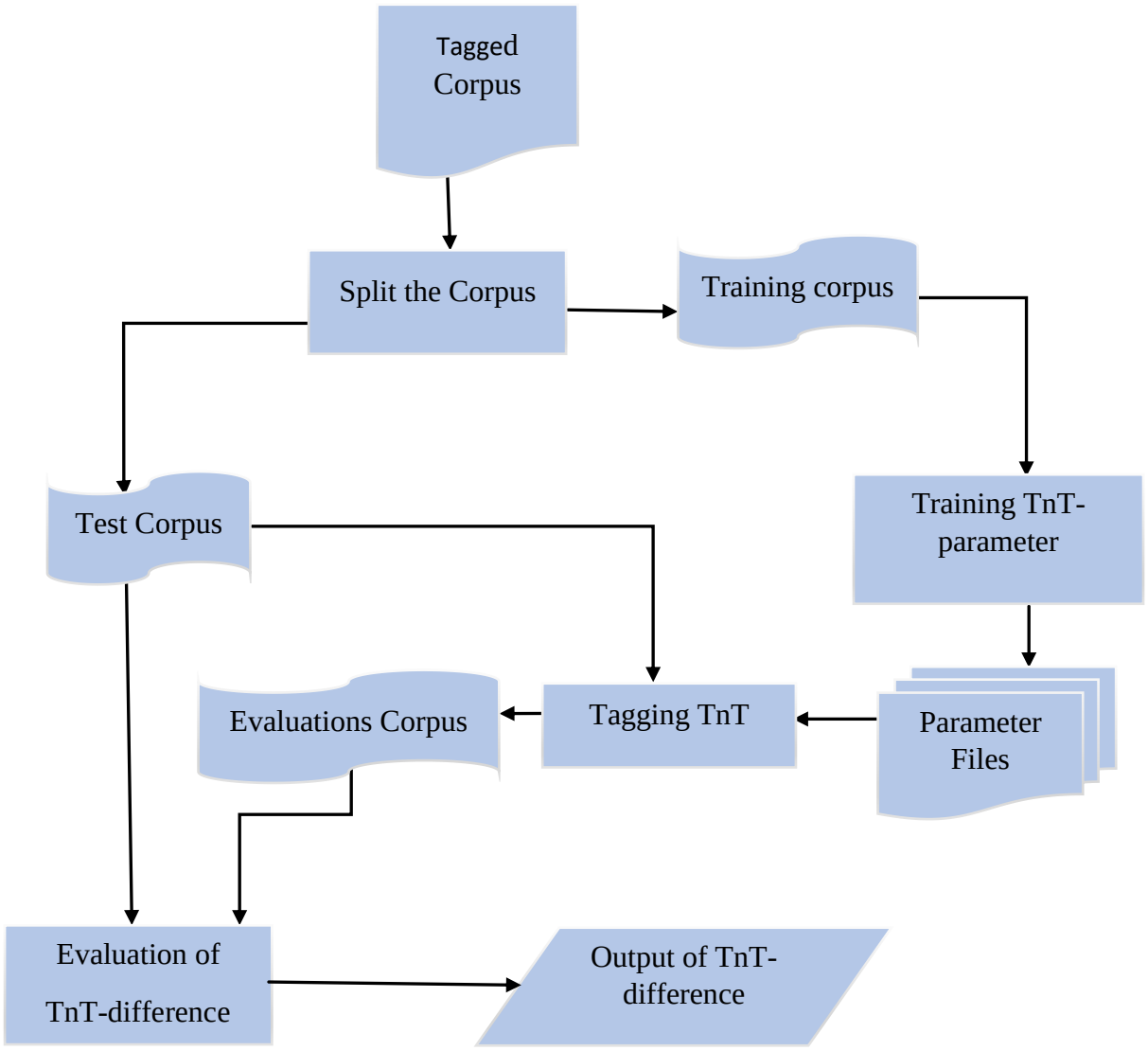

Figure 1.2 TnT Tagger Architecture

Maximum Entropy Model: The Maximum Entropy Model (MEM) is based on the principle of Maximum Entropy, which states that when choosing between several different probabilistic models for a set of data, the most valid model is the one which makes fewest arbitrary assumptions about the nature of the data [38]. The probability model for MEM is defined over (H, X, T), where H is the set of possible word and tag contexts or "histories", and T is the set of allowable tags. Given a sequence of words $\{w_1 \ldots w_n\}$ and tags $\{t_1 \ldots t_n\}$ as training data, hi is defined as the history

available when predicting t_i. The parameters $\{a_1 \ldots a_k\}$ are then chosen to maximize the likelihood of the training data [39].

Conditional Random Fields: Conditional Random Fields (CRF) are conditional probability distributions that take the form of exponential models [40]. A conditional model specifies the probabilities of possible label sequences given an observation sequence. The conditional probability of the label sequence can depend on arbitrary, non-independent features of the observation sequence. The probability of a transition between labels may depend not only on the current observation, but also on past and future observations [39]. The CRF model calculates the probability based on some features, which might include the suffix of the current word, the tags of previous and next words, the actual previous and next words etc. [41].

Rule Based Approaches: Rule-based POS tagging is the oldest approach that uses hand-written rules for tagging. Rule based tagger depends on dictionary or lexicon to get possible tags for each word to be tagged. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. Disambiguation is done by analyzing the linguistic features of the word, its preceding word, its following word and other aspects. For example, if the preceding word is article then the word in question must be noun. This information is coded in the form of rules. The rules may be context-pattern rules or as regular expressions compiled into finite-state automata that are intersected with lexically ambiguous sentence representations [42].

Brill tagger: The is based on rules. It was described and invented by Eric Brill [18] in his 1993 PhD thesis. It can be called as an error-driven transformation-based tagger. It is a form of supervised learning, which aims to minimize error; and is a transformation-based process, in the sense that a tag is assigned to each word and changed using a set of predefined rules. In the transformation process, if the word is known, it first assigns the most frequent tag, or if the word is unknown, it naively assigns the tag "noun" to it. Applying over and over these rules, changing the incorrect tags, a quite high accuracy is achieved. This approach ensures that valuable information such as the morphosyntactic construction of words is employed in an automatic tagging process.

Artificial Neural Network: In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are

continually tweaked. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined.

2.3.2 Unsupervised POS Tagging

The unsupervised POS tagging models is not like supervised models as they do not require pre-tagged corpora. Rather, they use advanced computational methods such as the Baum-Welch algorithm so as to automatically induce tag sets, transformation rules etc. There are basically two classes in which most of the tagging algorithms fall: rule-based taggers and stochastic taggers. The supervised approaches cannot be practically done easily to make them work in applicative settings, but they reach the best performance in many NLP tasks [34]. Not only this, the supervised systems should be trained on a large amount of annotations which are manually provided.

Transformation Based Learning (TBL): Brill described a system which learns a set of correction rules which helps to avoid linguistic rules that are manual. A set of rules is obtained by instantiating every rule template which has data from the corpus, with the help of predetermined rule template. This is done after the initialization process. The words that are tagged incorrectly are applied with each rule temporarily and hence the rule which reduces the maximum number of errors is identified and considered to be the best. Now this rule is added to the propped rules and on the new corpus formed this process iterates by taking the recently added rule, because with the help of remaining rules, the reduction of error rate less than a predetermined threshold cannot be possible [34].

Artificial Neural Network: Artificial neural networks (ANN) is a biologically influenced concept commonly used within the field of AI [43]. The idea for this concept originates from a highly abstract view of the brain, where information is transmitted through the network (brain). This is done by firing neurons that send their electrical charge through their many axons. The information is then processed by the receiving neuron through their many dendrites. If the cumulative information collected from all the dendrites exceeds the neurons activation threshold, the receiving neuron will fire and propagate information itself. An ANN works in a similar way, where a sending unit, a perceptron, is sending information (output), to the perceptron that it is connected to. The difference is that the modern perceptron lacks an activation threshold and will

always output information, even though that information might sometimes be zero (not to be confused with null).

An ANN is composed three layers [43] these are input, hidden and output layer.

Input layer represent the original information that is fed into the network and it is connected to the hidden layer.

Hidden layer is the one that is connected with the output layer. Its activity determined by the activity of the input layer and the weight on the connection between the input and hidden layer.

Output layer represent the outcome of the learning process from input and hidden layer. Its' behavior depend on activity of hidden units and the weight on the connection between the hidden and the output layer.

When ANN approach is taken in to POS tagger developments task, according to [12] before working on the actual ANN based tagger, it requires a pre-processing activity. The output of the pre-processing activity's taken as input for the input layer of the network. From which, the network learns by adapting the weights of the connection between layers until the correct POS is produced.

This approach is the other type of ANN training which is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption.

2.3.3 Hybrid

As its name implies, this approach combines features of both the rule-based approach and statistical approach, the rule based approach and the Artificial Neural Network or other different two approaches. Like rule-based systems, they use rules to specify tags. Like stochastic systems, they use machine-learning to induce rules from a tagged training corpus automatically. The transformation based tagger or Brill tagger is an example of the hybrid approach. Most work on POS tagging have got better results than the corresponding uncombined approaches.

2.4 Summary

This Chapter mainly deals with the most common approaches to POS tagging. Accordingly, the approaches used so far are described as advantage and disadvantage in summarized format in the Table 2.7.

Table 1.7 Advantage and disadvantage of different POS tagging approaches

Approach	Advantage	Disadvantage
Stochastic	Researchers may not need language specialists, expertise Coverage depends on the training data	Required a sufficient large sized corpus Inability to deal with unknown words Not easy to work with ill-formed input
CRF	ability to relax strong independence assumptions made in those models. Avoid a bias towards states with few successor states. Flexible enough in terms of feature selection	High computational complexity of the training stage of the algorithm. Does not work with unknown words, i.e., with words that were not present in training data sample.
TBL	Transformation rules can be created/ edited manually Sequences of transformation rules have a declarative, logical semantics Simple to implement Can be extremely fast	Does not provide tag probabilities Training time is often intolerably long,
Supervised	Achieved useful accuracies Suitable for most applications	Pre-tagged corpora are not readily available for the many languages and genres which one might wish to tag Tagging of training data is a costly and time-consuming process.
Unsupervised	Speedily scale to any language as it does not require many amounts of labeled text or an exhaustive list of hand coded rules. A small amount of labeled data in some form is still used as a bootstrap in many unsupervised approaches. Reducing annotation cost	Difficulty in evaluation, i.e., there is no test corpus represented in the cluster format. Have not achieved useful accuracies Not suitable for most applications
HMM	Attain good accuracy Can be trained from unannotated text.	Needs to be trained on a set of seed sequences and generally requires a larger seed. The algorithms for HMM such as, Viterbi and forward-backward are more expensive
TnT	Its speed is important for both fast tuning cycle and when dealing with large corpora. Its suffix guessing algorithm that is triggered by unseen words is very important for morphological very rich languages.	Under default settings the algorithm proposes a lot more possible tags than a morphological analyzer would.

Rule Based	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	Need 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
MEM	There is a great deal of flexibility in what contextual cues can be used. Powerful to achieve the accuracy	Has label bias problem Biased towards states with few successor states.
ANN	It is suitable for languages having small number of tag set and small amount of training corpus It combines the advantage of HMM and trigram tagger	As the number of tag set increase, the performance of the tagger become worse It has lower processing speed compared to stochastic approach Both selection and treatment of ambiguous words are performed by only considering the corpus

Chapter 2 : Related Work

3.1 Introduction

This Chapter reviews earlier POS tagging works conducted in Semitic languages which are categorized under the same language branch as the Ge'ez language and other Ethiopian and international languages which are not Semitic language families. Besides, this Section also reviews POS tagging works conducted using different POS tagging approaches such as conditional random fields (CRFs), support vector machines (SVMs), TnT, Rule-Based, HMM, Hybrid etc.

3.2 Development of POS Tagger for Ethiopian Languages

Yemane Keleta and Yamamoto Kazuhide [10] presented POS tagging research for Tigrinya from the newly constructed Nagaoka Tigrinya Corpus. The raw text was extracted from a newspaper published in Eritrea in the Tigrinya language. The POS tagged corpus contains 72,080 tokens and 73 tag set. Subsequently, a supervised learning approach based on CRFs and SVMs was applied, trained over contextual features of words and POS tags, morphological patterns, and affixes. For a reduced tag set of 20 tags, an overall accuracy of 90.89% was obtained on a stratified 10-fold cross validation. Enriching contextual features with morphological and affix features improved performance up to 41.01 percentage point, which is significant.

Binyam Gebrekidan [4] developed a POS tagger for Amharic language. The author designed a POS tagger state-of-the-art machine learning algorithm for Amharic language. The author used annotated data available for their experiments which is walta information center (WIC) corpus (≈207k) tokens. In order to increase the performance of the tagger the author used the following three methods: First, the POS tagged corpus (WIC) has been cleaned up to minimize the preexisting tagging errors and inconsistencies. Second, the vowel patterns and the roots, which are characteristics of Semitic languages, have been used to serve as important elements of the feature set. Third, state-of-the-art of machine learning algorithms have been used and parameter tuning has been done whenever necessary and as much as possible. Finally, the accuracies have crossed above the 90% limit.

Martha [44] developed a POS tagger for Amharic language for factored language modeling. For the POS tagger development, the author used a POS tag set developed within "The Annotation of Amharic News Documents" project at the Ethiopian Language Research Center. The tag set has 11 basic classes. Some of these basic classes are further subdivided and a total of 30 POS tags have

been identified. Although the tag set contains a tag for nouns with preposition, with conjunction and with both preposition and conjunction, it does not have a separate tag for proper and plural nouns. It consists of 210,000 manually annotated tokens of Amharic news documents. The author describes a series of POS tagging experiments aimed at providing a factored language model with an additional information source. Two software tools, TnT and SVM tool, have been applied to train different taggers. As SVM-based taggers outperformed the probabilistic ones, they decided to use them to tag the text for their factored language modeling experiment. The overall accuracy of the best performing TnT-based and SVM-based taggers is 82.99% and 85.50%, respectively. Generally, with respect to accuracy SVM-based taggers perform better than TnT based taggers although TnT-based taggers are more efficient regarding speed and memory requirement. They have developed factored language models (with two and four parents) for which the estimation of the probability for each word depends on the previous one or two words and their POS. These language models have been used in an Amharic speech recognition task in a lattice rescoring framework and a significant improvement in word recognition accuracy has been observed.

From the above related thesis works, Tigrigna [10] and Amharic [4,44], we learn two points. First, they share some common tag sets because they are from the same language family, Ethio-Semetic language family and they characterized by rich inflectional and derivational morphology [4,10,44]. As a result, Ge'ez also share some tag sets from the aforementioned languages. The other point is by using stochastic approach, they achieved good accuracy result. It confirms that for such morphologically rich languages it is advisable to using either stochastic only or the combination of stochastic approaches with others to get better performance.

Zelalem Mekuria and Yaregal Assabie [12] developed a POS tagger for Kafi-Noonoo using a hybrid approach. For training and testing purpose, 354 untagged Kafi-Noonoo sentences are collected from two genres and annotated using an incremental corpus preparation approach. For tagging purpose, 34 POS tags were identified. After assigning word class information on each word within the sentences, both HMM and rule-based taggers are trained on 90% of the tagged sentences to generate probabilities i.e., lexical and transitional probability for the statistical component of the hybrid tagger and set of transformation rules for the rule-based component of the hybrid tagger. Based on these probabilities and transformation rules, the hybrid tagger assigns the most suitable word class information for the given untagged Kafi-Noonoo texts. The performance of the prototypes, i.e., HMM, rule-based and hybrid taggers were tested using different experiments. As a result, HMM and rule-based tagger with unigram

initial state tagger show 77.19% and 61.88% accuracy respectively whereas, the hybrid tagger improves the accuracy to 80.47%. Even though there is no one way of choosing the size of training/testing set, this thesis applies heuristics such as 10% testing and 90% training corpus. But, doing so can bias the classification results and the results may not be generalizable.

Getachew Mamo and Million Meshesha [13] presented POS tagger for Afaan-Oromo using HMM approach. For training and testing purpose, the authors collected 159 sentences (with a total of 1621 words) from different sources to make the corpus balanced and they used 17 tag sets. In the tagging process, the tagger assigns word classes to a given Afaan-Oromo text with two main phases. In the first phase, the tagger trains on the training data in order to compute and store both lexical and transitional probability of training data. In the second phase, the tagger accepts untagged Afaan-Oromo text and tokenized into words. Then, the tagger assigns the correct POS tag for each token. This is achieved by using unigram and bigram model of the Viterbi algorithm by taking the stored information during the first phase. The authors have tested the performance of the tagger using tenfold cross validation mechanism. As a result, they have got 87.58% and 91.97% accuracy for unigram and bigram model respectively.

3.3 Development of POS Tagger for Non-Ethiopian languages

Eric Brill [18] developed a simple rule-based tagger for English language with very few rules performs on equivalence with stochastic taggers. The author ran two experiments where all words were known by the system. First, the Brown Corpus was divided into a training corpus of about one million words, a patch corpus of about 65,000 words and a test corpus of about 65,000 words. When tested on the test corpus, with lexical information derived solely from the training corpus, the error rate was 5%. Next, the same patches were used, but lexical information was gathered from the entire Brown Corpus. This reduced the error rate to 4.1%. Finally, the same experiment was run with lexical information gathered solely from the test corpus. This resulted in a 3.5% error rate. Note that the patches used in the two experiments with no unknown words were not the optimal patches for these tests, since they were derived from a corpus that contained unknown words.

Hadni Meryeme *et al.* [9] proposed POS tagging technique for Arabic language using hybrid approach. The developed tagger employed an approach that combines rule-based method with HMMs based on the Arabic sentence structure. The proposed technique uses different contextual

information of the words with a variety of the features which are helpful to predict the various POS classes. To evaluate its accuracy, the proposed method has been trained and tested with two corpora: The Holy Quran corpus and Kalimat corpus for discretized Classical Arabic language. Parts of it were used to train and to test the tagger. The experiment results demonstrate the efficiency of the method for Arabic POS tagging. In fact, the obtained accuracies rates are 97.6%, 96.8% and 94.4% for their Hybrid Tagger, HMM Tagger and for the Rule-Based Tagger with Holy Quran corpus respectively. For Kalimat corpus they obtained 94.60%, 97.40% and 98% for rule-based tagger, HMM tagger and their hybrid tagger respectively. In fact, the accuracy was slightly increased with the increasing of the number of words in the training corpus. However, their tagger cannot handle unknown words or tagging accuracy of unknown words was very low. Additionally, their tagger cannot handle extraction of multi-word terms.

3.4 Summary

From the related work, we have reviewed researches done for both Semitic and non-Semitic language families that were conducted by different approaches. From those reviewed, we understand those solutions were used different techniques such as rule based, TnT, CRF, SVMs, HMM, and hybrid. However, those all techniques are language dependent and the way they applied depends on the characteristics of the language. As a result, those POS taggers cannot be applied directly to Ge'ez language since it is morphologically complex and follow free grammar which can follow subject-object-verb, object-subject-verb or subject-verb-object order without change the meaning of the sentence. To our best knowledge, there is no research attempt on Ge'ez POS tagger. Therefore, the purpose of our proposed research is to fill-in this research gap.

Chapter 3 : Design of Ge'ez POS Tagger

4.1 Introduction

In this Chapter, tag set selection and preparation for Geez language will be presented. Additionally, a detail description of design issues and techniques of the Ge'ez POS tagger will be discussed. Besides, the design of combination of TnT, human annotated rule, and unknown word guesser using morphological pattern analysis will be presented.

4.2 System Architecture

POS tagging involves many difficult problems, such as insufficient amounts of training data, inherent POS ambiguities, and most seriously, many types of unknown words which are pervasive in any application and cause major tagging failures in many cases.

Several approaches have been proposed to annotate words automatically with their POS tags. Among these, the hybrid of TnT and rule-based approach is assumed to perform better than the TnT and rule-based taggers when they are taken alone. For this thesis, a hybrid approach, which is a combination of TnT, human annotated rule, regular expression and unknown word guesser is designed for Ge'ez language. The hybrid tagger of Ge'ez consists of three main components these are preliminary tagger which combines TnT, regex rule-based tagger and unknown word guesser tool based on prefix pattern analysis. The overall architecture of the system including the connection between the components is shown in Figure 4.1.

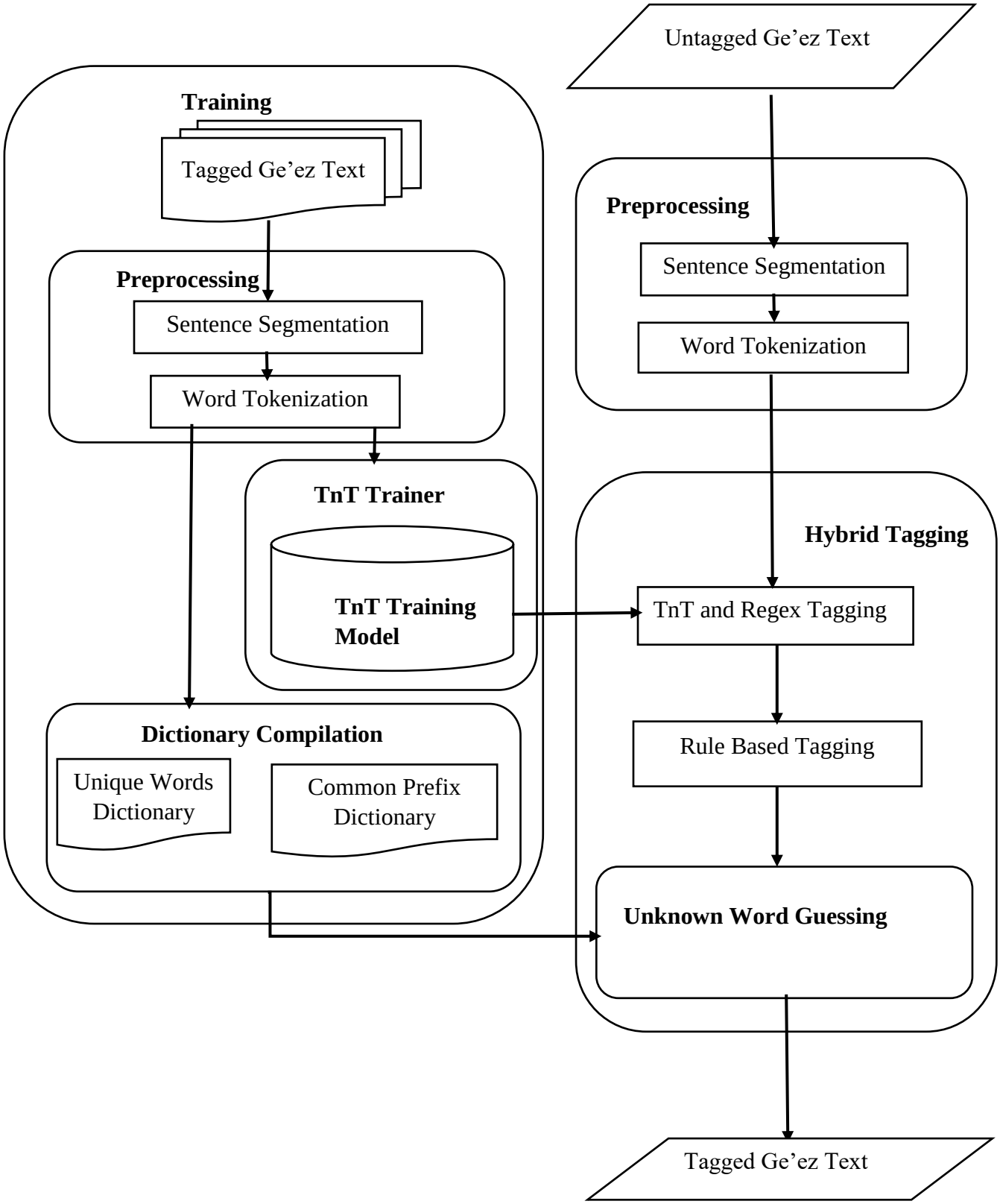

Figure 3.1 Hybrid Tagger Architecture

4.3 Training

Ge'ez POS Tagger uses a tagged training corpus to determine which POS tag is most likely for each context. For preliminary tagging purpose, we use TnT trainer. We train it by specifying tagged sentence data as a parameter when we initialize the tagger. The training process involves inspecting the tag of each word and storing the most likely tag for any word in a dictionary that is stored inside the tagger. The training component holds Preprocessing phase which performs sentence segmentation and word tokenization processes. Also, it contains Dictionary Compilation Phase for Unique word extraction and Common Prefix selection.

4.3.1 Tagged Geez Text

The training phase of Geez tagger accepts tagged corpus which is used for training to learn information about the tag-set, word-tag frequencies, etc. As far as the researchers' knowledge is concerned, there is no publicly available tag set for Ge'ez language. In order to identify and develop tag set for this thesis, first, the researcher has used convenience sample to select three Ge'ez language professionals namly, Memher Zer-Adawit Adhana, Ato Petros Zemaryam and Ato Negasi Gidey. Memher Zer-Adawit Adehana is Ge'ez language teacher in Holy Trinity college, Ato Petros is teacher of Ge'ez language in Miskaye Hizunan secondary school also he was Kine (Wax and Gold) teacher in Ethiopian orthodox church and Ato Negasi Gidey has deep knowledge in Ge'ez language as a result of that he wrote a fiction in Ge'ez language. Next, the researcher gives Untagged Ge'ez Text to those language experts. Then, they tagged the given Untagged Ge'ez Text by annotating equivalent POS tag of the words in the text. In order to select unique Ge'ez POS tag sets, the researcher made continuous discussion with those Ge'ez language professionals. The discussion was open end discussions. Finally, we agree to select 26 Ge'ez tag set. In addition, we adapt some tag sets from the work of [4].

According to our discussion with language experts about Ge'ez tag sets, we classified tag sets as basic classes and sub-classes of the basic categories of POS such as noun, verb, adjective, pronoun, adverb and preposition are considered. In the other hand, conjunction, interjections, cardinal numbers and punctuations are also included as basic classes in the process of identifying the tag sets. The hierarchical structure of the tag sets is identified in Figure 4.2.

Noun and its sub-classes: In nouns, because of tag set complexity problem, we did not include the entire noun sub classes. For the purpose of tag set preparation, we identify noun as a general

class and noun with conjunction, conjunction and preposition with noun, conjunction and preposition with noun possession, noun possession, conjunction with noun possession, conjunction with noun, preposition with noun, preposition with noun possession, and noun possession as sub-classes. Noun class and its sub-classes are explained in the following examples.

- Nouns that have not any prefix or suffix is tagged by N. Example አብርሃም/Abrham tagged as N.
- Noun prefixed with conjunction and when the conjunction cannot be separated from the noun is considered to be the conjunction noun subclass and tagged as CN. Example: ወምድር/Wemidir/and earth.
- Noun suffixed with conjunction and when the conjunction cannot be separated from the noun is considered to be the noun conjunction subclass and tagged as NC. Example: ሣዕረኒ/Saereni/the grass.
- Noun prefixed with conjunction and preposition and cannot have separated each other, it considered to be conjunction and preposition noun subclass and tagged as CPREN. Example: ወለወልደ/we-le-welde /and for the son.
- Noun prefixed with preposition and cannot have separated each other, it considered to be preposition noun subclass and tagged as PREN. Example: ለአብርሃም/le-Abriham / for Abriham.

Verb and its sub-classes: We identified one general tag set and two subclass tag set of verbs. Verb prefixed with conjunction and when the conjunction cannot be separated from the verb is considered to be the conjunction verb subclass and tagged as CV. Example: ወሆረ/WeHore/and he goes. Verb prefixed with preposition and when the preposition cannot be separated from the verb is considered to be the preposition verb subclass and tagged as PREV. Example: ለርእይ/LeReey/to view. The reaming every part of verb can be tagged as V. Example: ደይ/Dey/put.

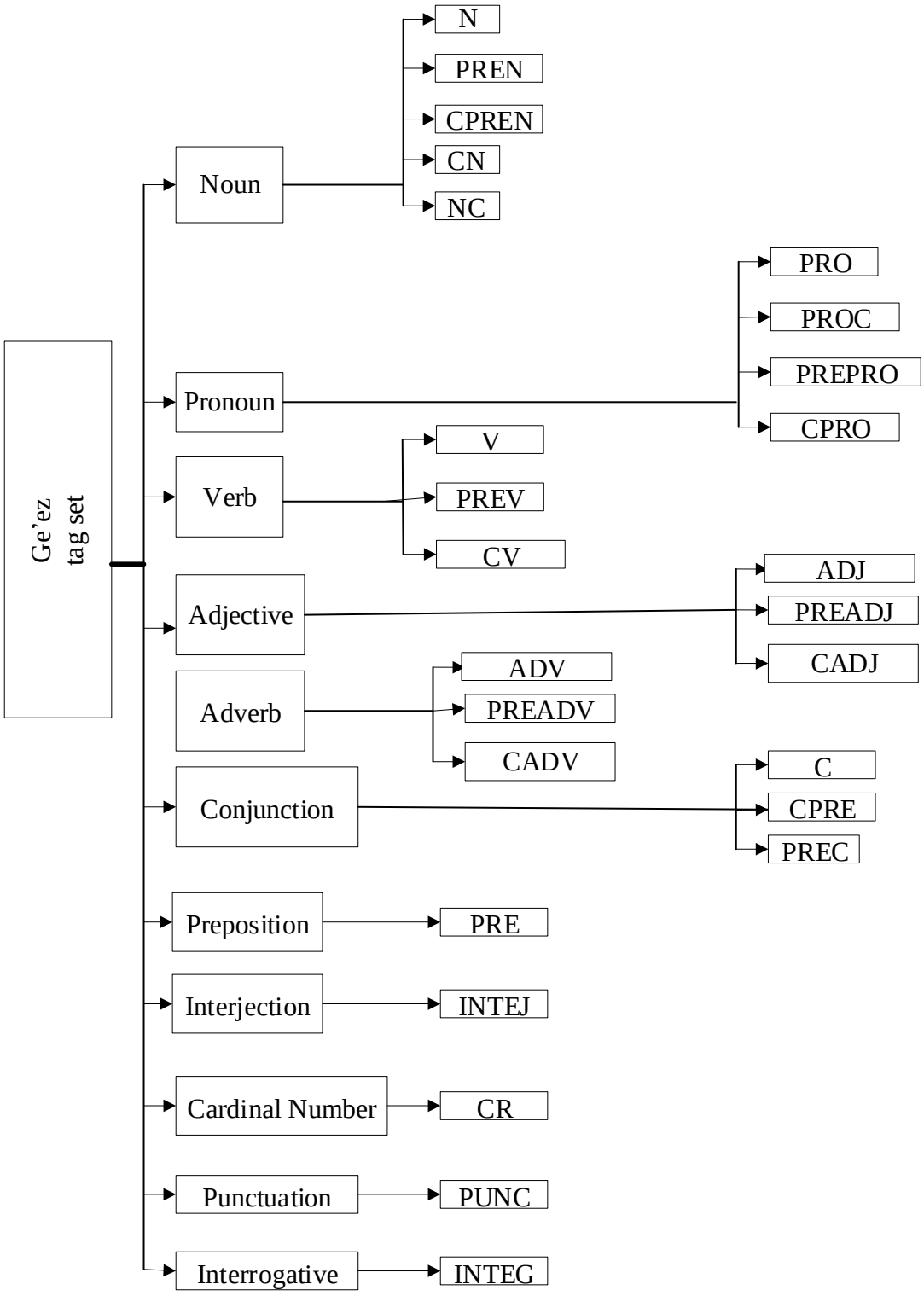

Figure 3.2 Ge'ez Tag Sets Hierarchy

Adjective and its sub-classes: In this main class, we identify one general class and two sub-classes. This class and its subclasses are explained in the following example.

- Adjectives attached with conjunction are tagged by CADJ. For example: ወኩሎ/wekulo/and all
- Adjectives attached with preposition are tagged as PREADJ. For example: በኩሉ/Bekulu/in all
- All other forms of adjectives that cannot be classified under the above classification are tagged by ADJ.

Pronoun and its sub-classes: In this main class, we identify one general class and three sub-classes. This class and its subclasses are explained in the following example.

- Pronouns suffixed with conjunction classified under pronoun conjunction and tagged as PROC. For example: አንተኒ/AnteNi/but you
- Pronouns prefixed with conjunction classified under conjunction pronoun and tagged as CPRO. For example: ወአንተ/weAnte/and you
- Pronouns prefixed with preposition classified under preposition pronoun and tagged as PREPRO.
- All pronouns that cannot be classified under the above sub-classes are tagged with PRO.

Conjunction and its sub-classes: In Ge'ez language, conjunctions are highly morphologically affixed. To handle this difficulty, we decided to have one main class and three subclasses. This class and its sub-class are
explained using the following example.

- Conjunction suffixed with preposition classified under conjunction preposition and tagged as CPRE. For example: ወውስተ/we-wiste/and inside
- Conjunction prefixed with preposition classified under preposition conjunction tagged as PREC. For example: በከመ/bekeme/ as like
- All Conjunction that cannot be classified under the above sub-classes are tagged with PRO.

Numerals: Ge'ez numerals, cardinal or ordinal are tagged as CR and ADJ respectively. For example: ፩/1 as tagged as CR.

Interjection: Interjections are words used to express strong feeling or sad emotion. All words that show this type of characteristics are tagged as INTEJ. Example: ሐዊሳ /hawisa/'sign of joy.'

Interrogative: In Ge'ez like as English grammar, an interrogative is a word that introduces a question which cannot be simply answered with yes or no. *All* interrogative words are tagged as INTERG. Example: መኑ/Menu/what or who.

Preposition:Preposition is a word used to link nouns, pronouns, or phrases to other words within a sentence. If prepositions are exist separated they tagged as PRE. Example. ላዕለ/Laele/above

Adverb*:* In this thesis, all separated adverb words are tagged as ADV. Example ጥቀ/Tiqe/small

In this main class, we identify one general class and two sub-classes. This class and its subclasses are explained in the following example.

- Adverbs attached with conjunction are tagged by CADV. For example: ወቀዳሚ/weqedami/and the first.
- Adverbs attached with preposition are tagged as PREADV. For example: በቀዳሚ /Beqedami/by the first.
- All other forms of adverbs that cannot be classified under the above classification are tagged by ADV.

Punctuation: refers to the specific markings, signs and symbols that are used in and around sent ences to give themstructure and to allow for correct understanding and comprehension [45].All Ge'ez punctuation marks such as ※ section mark,፡ word separator, ። full stop (period), ፣ comma, ፤ colon, ፥ semicolon, ፦ preface colon, ፧ question mark and ፨ paragraph separator etc. are tagged by PUNC. The summarized version of Ge'ez tag sets which are used to tag untagged Ge'ez texts are shown in Table 4.1.

Table 3.1 Ge'ez tag set

NO	Basic Category	Derived Tag	Description	Example
1	Noun	CN	Conjunction + Noun	ወምድር/Wemidir
2		CPREN	Conjunction + Preposition +Noun	ወለወልደ/we-le-welde
3		N	Noun	አብርሃም/Abriham
4		NC	Noun + Conjunction	ሣዕረኒ/Saereni
5		PREN	Preposition + Noun	ለአብርሃም/Le-Abriham
6	Verb	CV	Conjunction + Verb	ወልህቀ/Welihiqe
7		PREV	Preposition + Verb	ለርእይ/LeReey
8		V	Verb	ደይ/Dey
9	Adjective	ADJ	Adjective	ኩሉ/Kulu
10		CADJ	Conjunction + Adjective	ወዘንተ/Wezente
11		PREADJ	Preposition + Adjective	በኩሉ/Bekulu
12	Pronoun	CPRO	Conjunction + Pronoun	ወኪያሃ/Weki-yaha
13		PRO	Pronoun	አነ/Ane
14		PREPRO	Preposition + Pronoun	ለእማንቱ/LeMantu
15		PROC	Pronoun + Conjunction	አንተኒ/AnteNi
16	Conjunction	C	Conjunction	ከመ/Keme
17		CPRE	Conjunction + Preposition	እምኀበ/Emhabe
18		PREC	Preposition + Conjunction	ለእመኬ/Le-emeke
19	Cardinal Number	CR	Cardinal Number	፩/1
20	Interjection	INTEJ	Interjection	ናሁ/Nahu
21	Interrogative	INTEG	Interrogative	መኑ/Menu
22	Preposition	PRE	Preposition	ላዕለ/Laele
23	Adverb	CADV	Conjunction + Adverb	ወቀዳሚ/weqedami/
24		PREADV	Preposition + Adverb	በቀዳሚ/Beqedami/
25		ADV	Adverb	ጥቀ/Tiqe
26	Punctuation	PUNC	Punctuation	።

4.3.2 Preprocessing

In this sub section we discuss text segmentation which is the process of dividing written text into meaningful units, such as words, sentences, or topics. In Ge'ez POS tagger we implement text segmentation in two steps, sentence segmentation and word tokenization.

Sentence Segmentation: is the process of separating a string of written language into its component sentences. Due to the nature of this tagger, it works best when trained over sentence delimited input. Often NLP tools require their input to be divided into sentences for several reasons. However, sentence boundary identification is challenging because punctuation marks are often ambiguous. For example, Ge'ez language apply sentence delimiter such as ። or ፤ which is unlike other languages such as English. As well, question marks and exclamation marks may appear in embedded quotations, emoticons etc. Therefore, input for training is expected to be a list of sentences where each sentence is a list of (word, tag) tuples with sentence delimiter such as ። or ፤ which are sentence delimiters of Ge'ez language. This tagger also accepts single sentence as an input for tag function and the same format will have applied for an output.

Word Tokenization: Apart from sentence segmentation, word tokenization also important to the tagger. Word tokenization is splitting a sentence into individual words which contains pair of word/token and it's POS tag. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word divider (word delimiter). In Ge'ez writing systems however, words are explicitly delimited with a non-whitespace character. It uses A punctuation mark which is a colon (:). In this thesis work, based on input sentence, the tagger uses either colon (:) or white space as a delimiter. If the input sentence is from training component which is Tagged Ge'ez Text, it will use white space delimiter. On the other hand, if the input sentence is from testing component which is Untagged Ge'ez Text, it will use colon delimiter. Afterward, the output of the sentence segmenting component is given to the word tokenizer 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 Ge'ez word and punctuation marks. Finally, the corpus will be nested list which is list of words in side sentence lists. Algorithm 4.1 shows how text tokenization processed.

```
Input: Text
 FOR each Sentence in Text.split("sentence delimiter")
    IF Sentence is from Training Component
       Worddelimiter = whiteSapace
    ELSE
       Worddelimiter = colon(:)
    FOR word/token in Sentence.split("Worddelimiter")
       Get list of word/token
    GET list of Sentence
```

Algorithm 3-1 Text Tokenization

4.3.3 TnT Trainer

TnT is trainable on languages that separate words by white space [46]. TnT has two training options. If the tagger needs already trained input, the parameter must pass in Trained=True. Otherwise, it will call unk.train(data) with the same data you pass into the train()method.

During training phase TnT create the parameter files, parameter.lex (lexicon) and parameter.123 (trigram). Parameter.lex is lexicon information which contains frequency of words and their tags as they occurred in the training corpus. These frequencies are used during tagging to determine lexical probabilities. The Tri-gram file is, like the lexicon file, created during the parameter generation step. It contains the contextual frequencies for Uni, Bi, and Trigrams.

The TnT tagger maintains a number of internal frequency distribution and conditional frequency distribution instances based on the training data. These frequency distributions count unigrams, bigrams, and trigrams. Then, during tagging, the frequencies are used to calculate the probabilities of possible tags for each word. So, instead of constructing a backoff chain of Ngram tagger subclasses, the TnT tagger uses all the Ngram models together to choose the best tag. It also tries to guess the tags for the whole sentence at once by choosing the most likely model for the entire sentence, based on the probabilities of each possible tag.

4.3.4 Dictionary Compilation

During training phase dictionary compilation component compile two separate dictionary files which are unique words dictionary and common prefix dictionary. Those two files are used for the hybrid tagger in unknown word guessing component.

Common prefix dictionary file contained morphological patterns that are extract the common prefixes from the training tagged texts in each of POS categories. There, we considered prefix as

a common prefix if only that prefixes occurs in more than four distinct words. In addition to common prefixes patterns, common prefix dictionary file contained the POS tag information of categorized common prefixes.

In training phase after word tokenization unique words dictionary are extracted for the purpose of unknown word guesser component. Those unique words are extracted from the training corpus and saved with their POS tag information in dictionary file.

4.4 Hybrid Tagging

Obtaining a good balance between performance and accuracy is an important step for many reachability analysis problems. One way to address the trade-off between accuracy and coverage is to use the more accurate algorithms when we can, but to fall back on algorithms with wider coverage when necessary. For the case of controlling the trade-off between performance and accuracy of the tagger, we highly concentrated on the design of the tagger. We are using hybrid approach to get the advantage of individual components of the hybrid tagger.

Hybrid tagging process come after Preprocessing step which we discussed in Section 4.3.2. Hybrid tagger is combination of different class of taggers. In this thesis work, we use different type of taggers such as TnT, Regex, human written rule based taggers and unknown word guessing tool. Among those diverse statistical taggers, we adapt TnT tagger to hybrid tagger. The first reason is that it enables us to perform prefix analyzer, so it is best choice for morphological rich languages such as Ge'ez. The second reason is it maintains several internal frequency distribution and conditional frequency distribution instances based on the training data. These frequency distributions count unigrams, bigrams, and trigrams. Then, during tagging, the frequencies are used to calculate the probabilities of possible tags for each word. So, instead of constructing a back off chain of Ngram tagger subclasses, it uses all the Ngram models together to choose the best tag. It also tries to guess the tags for the whole sentence at once by choosing the most likely model for the entire sentence based on the probabilities of each possible tag. In addition to this, it is fastest for both training and tagging, and largely used to assign the correct label sequence to sequential data or assess the probability of a given label and data sequence. Even though TnT is preferred tagger among those statistical taggers for Ge'ez language, still it has shortcoming. TnT does not deal with prefix pattern of word and it tag UNK for unknown words. Regex can solve slightly the drawback of TnT tagger. To get acceptable accuracy of Ge'ez POS tagger using only the

aforementioned techniques as hybrid tagger is not enough, because Ge'ez language is the most morphologically complex language and free grammar which is there is no agreement on subject-object-verb order. Furthermore, human written rules and unknown word guessing are combined to the hybrid tagger.

4.4.1 TnT and Regex Tagging

For automatic POS tagger application, a tagger with the highest possible accuracy is required. The debate about which approach solves the POS tagging problem best is not finished. As languages are varying in grammar, morphological derivation and morphological inflection, one best approach for one language may not work for other languages. In addition, using the same corpora may vary the result if it uses different approaches. Recent comparisons of approaches that can be trained on corpora [47] have shown that in most cases statistical approaches outperformed finite-state, rule-based, or memory-based taggers outperformed finite-state, rule-based, or memory-based taggers. They are only surpassed by combinations of different systems forming a "voting tagger".

Furthermore, a tagger may be selected based on its features like how it handles unknown word, word which was not found during training, performance and accuracy. For this research work, we think TnT will fit to solve the problem of POS tagging for Ge'ez language among statistical taggers.

Because it uses linear interpolation for smoothing, by deleting interpolation, the respective weights will be determined. In addition, it is stochastic tagger based on trigram analysis, which uses a suffix analysis technique based on properties of words like, suffices in the training corpora, to estimate lexical probabilities for unknown words that have the same suffices. This feature is very important for language that are morphologically complex such as Ge'ez. Besides, it is optimized for training on larger variety of corpora and for speed. Furthermore, it used linear interpolation for smoothing. By deleting interpolation, the respective weights will be determined which is adopted from Peter *et al* [48].

The set of possible tags for a given word is derived from the training data. It is the set of all tags that exact word has been assigned. The probability of a tag for a given word is the linear interpolation of three Markov models; a zero order, a first order, and a second order model. A beam search is used to limit the memory usage of the algorithm. The degree of the beam can be

changed using N in the initialization. N represents the maximum number of possible solutions to maintain while tagging.

TnT tagger tag a single sentence at a time by producing a list of tags. Then it associates the sequence of returned tags with the correct words in the input sequence. In TnT tester part, first it untagged the golden standard sentences. Then it gets a list of untagged and tagged sentences for testing. Next, it scores the accuracy of the tagger against the gold standard. In other words, it strips the tags from the gold standard text, retag it using the tagger, and then compute the accuracy score.

Regular Expression: In this thesis work, the regular expression **(Regex)** tagger is used as back off tagger for TnT tagger in order to improve the performance of the tagger. Regular expression is class of sequential based tagger [49]. It assigns tags to tokens on the basis of matching patterns. For instance, we might guess that any word contains ten digits of numbers or match numbers with \d is a cardinal number, and is tagged as CR. It follows sequential order, and the first one that matches are applied. This means that if you have two expressions that could match, the tag of the first one will always be returned, and the second expression won't even be tried. Basically, language is naturally inexact and there are always exceptions to the rule so be careful of over-specifying is need. However, it handles most known word patterns, such as the word ወለደ(welede)/getting birth, with its sufficed. The final regular expression (r'.*', 'N'), is a catch-all that tags everything as a noun. In addition to tag the most know words this tagger is a best way to tag date patterns, money patterns, location patterns and so on.

4.4.2 Rule based Tagging

We apply human written rules in hybrid tagger to solve ambiguous words like **ውእቱ(wuetu)** and **ይእቲ (yiEti)**, which have more than one POS tag. In Ge'ez language, the word ውእቱ(wuetu) and ይእቲ (yiEti) are assigned for verb, adverb, pronoun, etc. To understand ambiguity of the word "ውእቱ(wuetu)", we can see the following cases:

- Case 1: ወፈለግ/CN ራብዕ/ADJ ***ውእቱ/V*** አፍራጥስ/N ።/PUNC when it tagged as Verb.
- Case 2: ጐየ/V ሎጥ/N ***ውእቱ/PRO*** ወከልኤ/CADJ አዋልዲሁ/N ምስሌሁ/PRE ።/PUNC when it tagged as Pronoun.
- Case 3: በላዕከ/V እምነ/PRE ***ውእቱ/ADJ*** ዕፅ/N ዘአዘዝኩከ/CV ከመ/C ኢትብላዕ/V ።/PUNC when it tagged as Adjective.

From the cases, we have observed more than one POS tag for a single word ውእቱ (wuetu). In Ge’ez language, the ይእቲ (yiEti) words also share the same property as ውእቱ (wuetu). The algorithm of human annotated rule is shows in Algorithm 4.2.

This rule is manually written by Ge’ez professional and used to handle some vague words. It uses one upto three preceding and succeeding words or tags of the of the ambiguous word in sentence. Finally, according to the rule, the word assigned a POS tag that fulfill the rule.

```
Input: Sentence
FOR each word 'ውእቱ' in Sentence
   IF word is 'ውእቱ'
      FOR wordno, (newword, post) in enumerate(sent)
        IF newword==word
         IF sent[wordno-1][1]in("PRO,ADJ,C,ADV,INTEJ,CADJ")
           Return V
         ELSE IF sent[wordno-1][1]=='CV'and sent[wordno+1][1]== 'N'
           Return ADJ
         ELSE IF sent[wordno+1][1]=='N'and sent[wordno+2][1]in (N,C,ADJ,PRE)
           Return ADJ
         ELSE IF sent[wordno+2][1]==PUNC orsent[wordno+1][1]in(CV,ADJ,C,PREN)
           Return V
         ELSE IF sent[wordno+1][1]== 'N'
           Return ADJ
         ELSE
           Return PRO
         END IF
       END IF
      END FOR
     END FOR
   ELSE IF word starts with ውእቱ
    Return PRO
   ELSE IF word starts with 'ለ' or 'በ'
    Return PREADJ
   ELSE IF word starts with 'ወ' or 'ዘ'
    Return CADJ
   END IF
END FOR
```

Algorithm 3-2 Human Annotated Rules

4.4.3 Unknown Word Guessing

The main problem in the TnT phase was guessing the tag profile for unknown words. Therefore, in order to develop a high accuracy tagger, a good quality unknown word guesser is essential to integrate with the existing hybrid tagger of TnT with Regex and rule based tagger. Most unknown word guessing modules use morphological/compound analysis or ending analysis, or a combination of both. The difference between morphological analysis and ending analysis is that the former bases its analysis on morphologically related words already known to the lexicon, whereas the latter bases its analysis solely on a word's ending. This can be explained by the fact that morphologically related words share the same stem (the common part shared by all word forms) as the given unknown word. In this thesis work we will use prefix analysis of morphological pattern analysis.

In this thesis work, we are not going to deal with the detail of morphological analyzer of Ge'ez language for the case of complexity of the language; however, we use unknown word guessing mechanism [50]. In unknown word guessing, the POS tag of an unknown word is predicted using affix of the unknown word, morphological patterns and substrings methods. We use probability method to guess the POS tag of unknown word.

POS tagging using statistical methods can be done without directly using the morphology of the language by using large size of the training corpus and this may achieve reasonably better results. However, for morphologically complex and poor resource languages like Ge'ez, a significant performance improvement can be achieved by integrating the essential morphological elements in the features that are learned by stochastic methods. Next, we will explore briefly Ge'ez morphology with the view to find vowel patterns that will be used in improving POS tagging accuracies.

Affix is a morpheme that is combined to a stem/root of a word to form a new word. It has two major parts namely prefixes and suffixes and it comes in any language. Prefix is an element that is placed at the beginning of a root word. Suffix is an element that is placed at the end of a root word. In this thesis work, we concentrate on prefix only because those suffixes of words were handled on the previous tagger, TnT tagger. Finally, it extracts common prefixes in each of the categories. There, we considered a prefix as common prefixes if only that prefix occurs in more than three distinct words.

In prefix analysis we consider two cases; Unknown word guessing using POS tag as prefix words and Unknown word guessing using unknown prefix words. The detail descriptions of those cases are shown in Algorithm 4.3 and Algorithm 4.4 respectively.

```
Input: Sentence
FOR each UnknownWord in Sentence
    IF UnknownWord start with any Ge'ez POS tag
        FOR each Prefix in UnknownWord
            Split prefix from UnknownWord
            Add prefix to List
        END FOR
        IF Stem in UniqueWordList
            Tag UnknownWord
        ELSE
            Apply Default Tagger
        END IF
    ELSE
        Go to Next Algorithm 4.4
    END IF
 END FOR
```

Algorithm 3-3 Unknown word guessing using other POS tag as prefix

```
Input: Sentence
FOR each UnknownWord in Sentence
    FOR i = 1 to Prefix Size
    IF Prefix[i] in CommonPrefixDictionary
       GET Most Frequent POS tag of Prefix
       BREAK
    END IF

END FOR
```

Algorithm 3-4 Unknown word guessing Using Prefix

Chapter 4 : Experiment

5.1 Introduction

In this Chapter, detail implementation of the proposed system is presented. Additionally, corpus preparation, the tag set, the annotated corpus and the issues surrounding the application of the tagging methods on the corpus are explained. Furthermore, the issues include cleaning the corpus, tokenization, feature extraction, training and testing procedures are discussed.

5.2 Corpus Preparation

Corpus is a collection of text, a large and structured set of texts or speech. Text corpus 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. In POS tagging, corpus with additional linguistic information is called tagged. In fact, the annotated text i.e., the tagged corpus is considering representing all the domains of the language. Based on the source of the corpus, domains can be text of Bible category, scientific category, news category, fiction category, editorial category etc. A corpus with all possible categories is called a balanced corpus. Although it is difficult to prepare a balanced corpus, it is an important element in most natural language processing applications in general and POS tagging in particular in order to maintain the language representation. However, a category specific corpus contains words are faced to performance degradation if the train or test text are from different category.

Development of balanced corpus takes time, effort/skills of language experts and money as it needs data to be collected from different domains. Ge'ez is one of the under-resourced languages both in terms of electronic resources and processing tools. Due to these constraints, a balanced corpus is not developed for this thesis work rather one category, Bible category. To the best of our knowledge, let alone a balanced corpus, there is no category specific corpus developed for Ge'ez language. Therefore, an incremental approach is used for developing a category specific corpus. Ge'ez corpus was collected from web[1] that was written by August Dillmann so called ''The Ethiopic Bible''. Afterwards, these sentences are given to linguistic professionals for manual tagging.

[1] http://www.tau.ac.il/~hacohen/Biblia.html

We select Genesis from the part of Old Testament for the case of two reasons. The first one, is it narrate creations, and it is the history of many social political and economic perspective of the first humans being especially the people of Hebrew and Egypt. Secondly, Old Testament is available in the web in editable soft copy format. Before to tag the corpus, we made some preprocess and ready for tagging that is why manual tagging process is tedious and time-consuming activity. We remove (:) colon that was used as word separator in the corpus, there was spelling error and has been corrected by the language professionals, we also separate each word in sorted manner. Then tagged using Microsoft Excel by the linguistic professionals. These tagged sentences were used for experiments and evaluation as training and test corpus. This process is repeated until the desired size of the corpus for this thesis work is achieved. Finally, we select 34 Chapters and 1087 Verses from the aforementioned resource. A sample corpus is shown in appendix A.

5.3 Implementation

Ge'ez POS tagger is implemented using corpus from a single genre using NLTK and Python programming language as a tool. The reason behind the selection of these tools is their suitability for processing POS tagging. NLTK is an open source toolkit that contains open source python modules, linguistic data and documentation for research and development in NLP field. Python is a general purpose interpreted, versatile, interactive, object-oriented, and high-level popular programming language. It is a simple but powerful programming language with admirable functionality for processing linguistic data. It has efficient and high-level data structure with simple but effective approach to object-oriented programming. It supports many NLP tasks such as tokenizer, stemmer, POS tagger, classifier with distributions for different operating systems for different languages such as English, German etc. Moreover, its syntax and dynamic typing feature with its interpreted nature makes it a powerful language for POS tagging task

Preprocessing component of hybrid tagger that was explained in Chapter Four, have three main modules, sentence segmentation, word tokenizer and sentence shuffler module. The sentence splitter module accepts tagged texts using Ge'ez corpus reader and splits down at sentence level based on Ge'ez sentence end marker characters. The default sentence tokenizer is an instance of NLTK tokenize with '\n' to identify the gaps of sentences. It assumes that each sentence is on a line all by itself, and individual sentences do not have line breaks. We customize this, by passing in our own tokenizer to the function to tokenize Ge'ez language sentences.

Afterwards, the word tokenizer module tokenizes the output of sentence splitter module into word level. In the training phase, the tokenized word comprises two components, word/token and POS tag. The word and its POS tag is separated using forward slash (/) character. This enables the tagger to compute statistical information for both word and POS tag during the training phase.

5.4 Test Results

Several experiments with different training set on three POS tagger have been conducted for Ge'ez POS tagger. The entire corpus shuffled and divided into two main sets: training set and testing set. The training set covers 90% of the entire corpus. The remaining 10% of the corpus is used for testing purpose.

5.4.1 Test Result of TnT Tagger

Ten different experiments are conducted on the TnT tagger using different portions of the training set to see the excellence of the training set based on the observation that can be made on the learning curve. We 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, we kept on adding the training data by 10% until they got a desired performance of the tagger. Table 5.1 shows the different experiments conducted using different portions of the training set with the corresponding performance of the tagger.

Table 4.1 TnT tagger performance

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance (%)	50.65	58.51	62.78	66.08	69.08	70.44	72.57	74.08	75.187	77.87
Difference	50.65	7.86	4.27	3.3	3	1.36	2.13	1.51	1.107	2.683

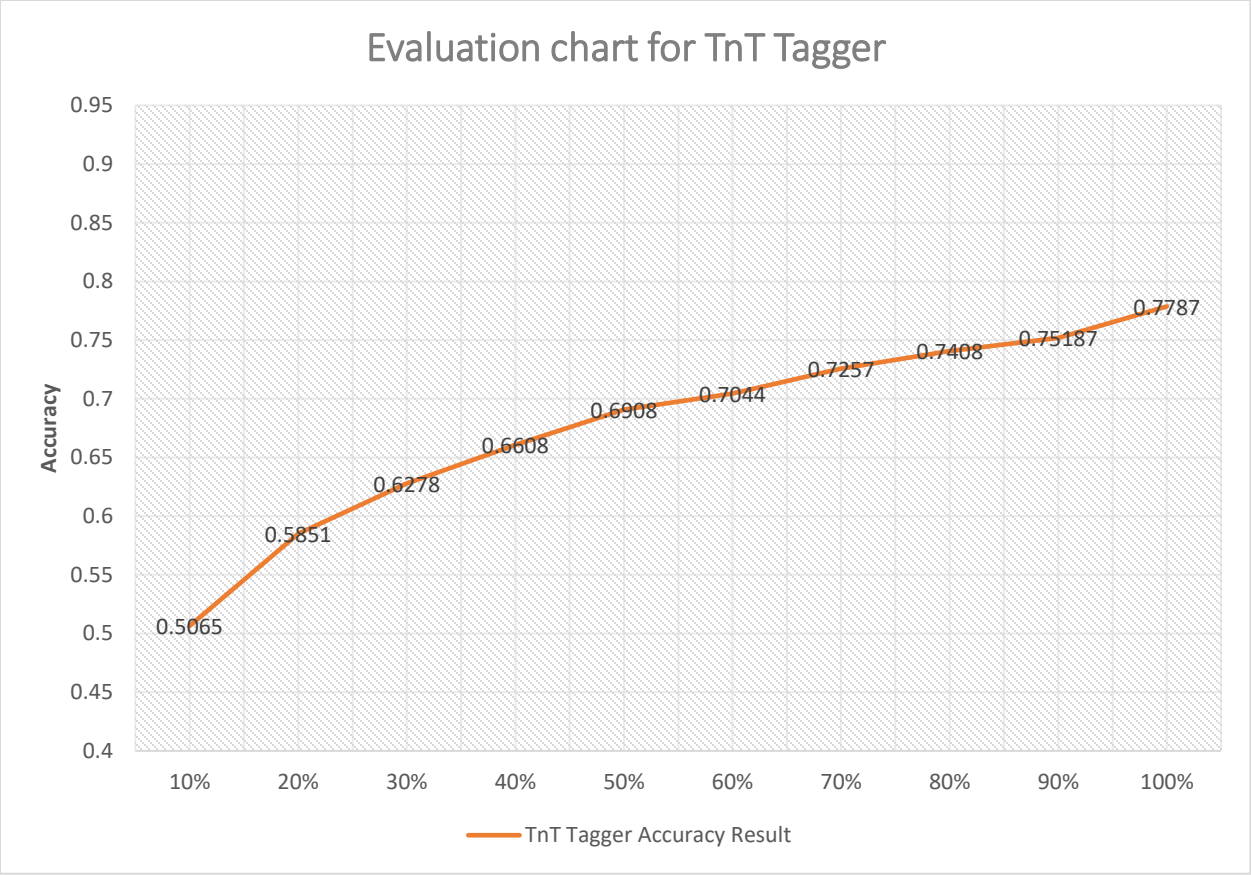

Figure 4.1 TnT Tagger Performance curve

The TnT tagger in Figure 5.1, shows 77.87 % accuracy when all the training data is used (100%). However, this result indicated the worst performance of an annotating Ge'ez corpora. This can be explained by two reasons. The first one is as TnT is a statistical tagger for training purpose it needs large corpus size but we use small corpus size which makes it very difficult for stochastic taggers to create probability distribution to hold transitions between different states. The second reason can be from grammar order in Ge'ez sentences in which free grammar which is no agreement among subject-object-verb order. Due to the aforementioned reasons, for Ge'ez language, TnT tagger score the worst accuracy result comparing with different language using this tagging approach, for example for English using Penn Treebank corpus which contains 50,000 sentences (1.2 million words) scores an accuracy of 96.7 % . In the same manner (additionally) for Amharic language using 1065 news texts (210,000 words) score the overall performance 92%.

5.4.2 Test Result of TnT and Regex Tagger

To test the performance of the TnT and Regex Tagger like that of TnT tagger, ten different experiments are conducted using different portions of the training set. Table 5.2 shows the different experiments conducted using different portions of the training set with the corresponding performance of the TnT with backoff of Regex tagger.

The most difficult task of TnT tagger is tagging of unknown words, words do not appear in training phase [23]. Hence, if the baseline TnT algorithm encounters a word in the testing set which did not appear in the training set, it will simply annotate it as "UNK" (unknown). Rather than failing to annotate in this way, the alternate versions of TnT identify a backoff tagger. Thus, when the algorithm comes upon an unknown word, it will pass off the tagging task to the backoff tagger. Such backoffs can be chained together but there is usually no additional improvement in having more than one or two backoffs. The most common class of lexeme in the corpus is nouns. TNT and Regex performs better than TnT based tagger. By replace "UNK" to "N" get a little bit accuracy change in the tagger. Figure 5.2, shows the curve 82.23 % which is 4.36 % difference comparing with TnT tagger.

Table 4.2 TnT and Regex tagger performance

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance (%)	63.50	68.56	71.42	73.65	75.67	76.77	78.45	79.65	80.04	82.23
Difference	63.50	5.06	2.86	2.23	2.02	1.1	1.68	1.2	0.39	2.19

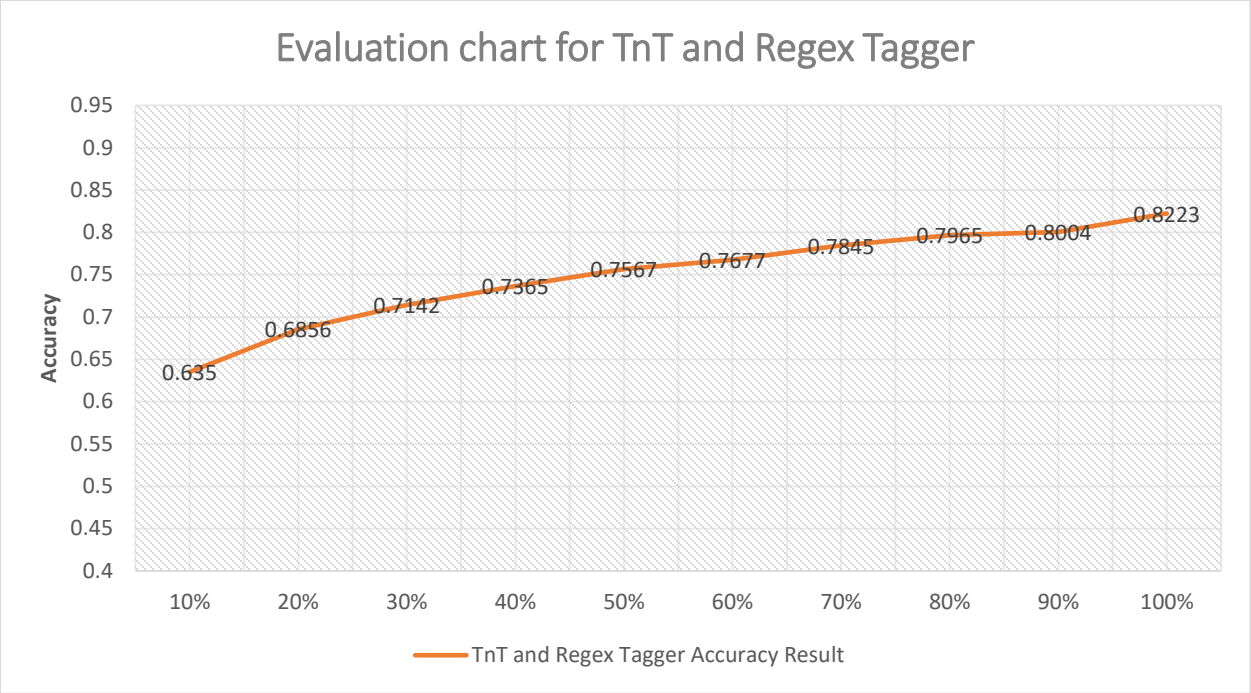

Figure 4.2 TnT with Regex Tagger Performance curve

5.4.3 Test Result of Hybrid Tagger

Hybrid tagger of Ge'ez language is combination of TnT with backoff of Regex tagger and contains unknown word guessing. In order to tag a given text with the hybrid tagger, first the Regex assigns tags to tokens on the basis of matching patterns. For instance, we might guess that any word contains ten digits of numbers or match numbers with \d is a cardinal number, and is tagged as CR. It follows sequential order, and the first one that matches are applied. The final regular expression (r'.*', 'N'), is a catch-all that tags everything as a noun. The reaming task will be done by TnT tagger. Even though the combined tagger, TnT with backoff of Regex tagger is better perform than TnT only, but still the result is acceptable. Consequently, it is important to associate unknown word guessing with the tagger which is making hybrid tagger. In addition to TnT with backoff of Regex tagger, the hybrid tagger work by guessing unknown word using morphological pattern of the word. In unknown word guessing, the POS tag of an unknown word is predicted using affix of the unknown word, morphological patterns and substrings methods. We use probability method to guess the POS tag of unknown word. Finally, by combing all those techniques we got an acceptable performance result.

Table 4.3 Hybrid Tagger performance

Training set	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Performance (%)	87.24	89.46	90.46	91.30	91.97	92.24	92.85	93.40	93.92	94.32
Difference	87.24	2.22	1.00	0.84	0.67	0.28	0.61	0.55	0.52	0.40

5.5 Discussion

After the experiment of different Ge'ez POS taggers ware conducted, we made performance analysis on individual taggers for the case of analyzing the performance of those taggers for the different POS 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 Ge'ez POS taggers. A total of 26 tags are identified in this research work and based on their frequency, they are divided into two groups namely the 13 most frequent tags and the rest as others. The frequency of the tags is given in Table 5.4.

Table 4.4 Frequency of Tags

Tag	Tag frequency within total corpus	Tag frequency with in training se	Testing set	
			Tag frequency	%
N	4214	3803	411	9.75
CV	2178	1940	238	10.93
V	1648	1482	166	10.07
PRE	964	868	96	9.96
PREN	855	760	95	11.11
C	840	756	84	10.00
ADJ	787	713	74	9.40
CN	604	548	56	9.27
PRO	369	327	42	11.38
CADJ	243	223	20	8.23
ADV	209	185	24	11.48
PREADJ	172	156	16	9.30
CPREN	165	152	13	7.88
OTHERS	1906	1727	179	9.39
Total	15154	13640	1514	9.99

The confusion matrix for the TnT Ge'ez POS tagger is given in Table 5.5. The TnT based tagger confusion matrix shows that it assigns 1207 tags correctly and 307 tags incorrectly to the tokens within the testing set. Due to lack of balanced and large corpus for training the system, it confused the tags to other POS tags; for instance, out of 166 verb tokens, 74 are assigned incorrectly to other POS tag. The performance of the TnT based tagger varies from one POS tag to another POS tag. As we can see from Table 5.5, the results in a confusion matrix. Rows represents tags proposed by our tagger, predicted tag. Columns represents tags that was manually tagged, actual tagged. The performance of the TnT tagger is different for the different POS tags. The order of performance of the TnT tagger for the POS tags in descending order is: ADV, PRE, PRO, ADJ, N, PREN, CADJ, CV, V and CN. In this confusion matrix, the worst performance is for CN because the tagger confused during tag with CV and C.

Table 4.5 Confusion matrix for TnT based tagger

		Test												Total	Performance (%)
		N	CV	V	PRE	PREN	C	ADJ	CN	PRO	ADV	CADJ	Others		
Reference	N	340		1			1	1			1		67	411	82.73
	CV		162										76	238	68.07
	V			92									74	166	55.42
	PRE				94								2	96	97.92
	PREN					69							26	95	72.63
	C				1		81						2	84	96.43
	ADJ	1		1				62					10	74	83.78
	CN	1							25				30	56	44.64
	PRO			1						36			5	42	85.71
	ADV										24			24	100
	CADJ											14	6	20	70
	Others												208	208	100

The confusion matrix for the TnT and Regex Ge'ez POS tagger is given in Table 5.6. The TnT based tagger confusion matrix shows that it assigns 1319 tags correctly and 195 tags incorrectly to the tokens within the testing set. This matrix shows slightly performance improvement on some POS tags like nouns and verbs. However still the tagger confused in some POS tags like noun prefixed with conjunction, out of 56 CN tokens, 31 are assigned incorrectly to other POS tag. The performance of the TnT based tagger varies from one POS tag to another POS tag. As we can see from Table 5.6, the results in a confusion matrix. Rows represents tags proposed by our tagger, predicted tag. Columns represents tags that was manually tagged, actual tagged. The performance of this tagger is different for the different POS tags. The order of performance of this tagger for the POS tags in descending order is: V, ADV, N, PRE, C, PRO, ADJ, PREN, CADJ, CV, and CN. As in Table 5.5 for TnT, in this confusion matrix also, the worst performance is for CN because the tagger confused during tag with V.

Table 4.6 Confusion matrix for TnT and Regex based tagger

		Test												Total	Performance (%)
		N	CV	V	PRE	PREN	C	ADJ	CN	PRO	ADV	CADJ	Others		
Reference	N	407		4										411	99.03
	CV		162	76										238	68.07
	V			166										166	100
	PRE			2	94									96	97.92
	PREN			26		69								95	72.63
	C			3			81							84	96.43
	ADJ			12				62						74	83.78
	CN			31					25					56	44.64
	PRO			6						36				42	85.71
	ADV										24			24	100
	CADJ			6								14		20	70
	Others			29									179	208	86.06

The hybrid tagger confusion matrix shows that it assigns 1428 tags correctly and 86 tags incorrectly. Even though, the tagger still confused the tags to other POS tags for different reasons such as morphological complexity of the language, lack of standard corpus and small size of tokens in the prepared corpus, but as we can see from Table 5.7, the confusion made by the hybrid tagger is less than the confusion made by the individual tagger. Like TnT and TnT with Regex, the performance of the hybrid tagger varies from POS tag to another POSs tag. Table 5.7 indicate that the performance of CV and PREN better than TnT and TnT with regex taggers.

Table 4.7 Confusion matrix for Hybrid tagger

		Test												Total	Performance (%)
		N	CV	V	PRE	PREN	C	ADJ	CN	PRO	ADV	CADJ	Others		
Reference	N	407		4										411	99.03
	CV		236										2	238	99.16
	V			166										166	100
	PRE			2	94									96	97.92
	PREN		1	26		94								95	98.95
	C		2			1	81							84	96.43
	ADJ			7				67						74	90.54
	CN		29			2			25					56	44.64
	PRO			6						36				42	85.71
	ADV										24			24	100
	CADJ		6									14		20	70
	Others		13	1		10							184	208	88.46

Chapter 5 : Conclusion and Future Work

6.1 Conclusion

POS tagging is the process of assigning POS like noun, verb, preposition, pronoun, adverb, adjective or other lexical class markers to each word in a sentence or literature. POS tagging is the first step to understanding a natural language. Most other tasks and applications heavily depend on it. POS tagging is considered as one of the basic necessary tools. It is a research area in the field of NLP for different languages. Several techniques have been suggested to tag words automatically with their POS tags. Among these, the hybrid of TnT with human annotated rule, regex and unknown word guessing of Ge'ez language is assumed to perform better than the TnT taggers taken alone.

Corpus is an important component in NLP in general and POS in particular. For this thesis, a corpus with a total of 1305 sentences is collected from one genre. For this thesis, 26 POS tags are identified as a tag set for annotating a raw text. The tag set indicates only word class rather than gender, number, tenses etc. The training set consists 90% of the total corpus (around 1175 sentences) and the testing set consists 10% of the corpus (around 130 sentences).

NLTK and Python3.6.2 are used in the implementation and experiment of the Ge'ez POS tagger. Hence, different experiments are conducted for the three types of taggers namely the TnT tagger, TnT with Regex tagger and Hybrid tagger. Therefore, 77.87%, 82.23% and 94.32% performances are obtained for TnT tagger, TnT with Regex tagger and Hybrid taggers respectively. Therefore, it is possible to conclude that the hybrid tagger performs better than the TnT tagger and TnT with Regex tagger used individually.

6.2 Contribution

The main contributions of this thesis work are listed as:

- Prepare tag sets of Ge'ez language
- Designed new system architecture that used to tag Ge'ez language sentences
- Combined unknown word guesser to statistical tagger i.e. TnT and regular expression tagger
- Identified and Designed rules for the most ambiguous words of Ge'ez language.

6.3 Future Work

There are lots of research areas in NLP that can be done for local languages. The same thing holds true for Ge'ez language. Therefore, to assist researchers, it will be of great paramount if a standard corpus for Ge'ez language is developed that will be available for NLP researchers in Ge'ez language. Among these, POS tagging is a useful form of linguistic analysis. It serves as pre-processing component for many higher levels NLP applications such as spelling checker, grammar checker, question answering, etc. Therefore, the researchers in the area of NLP application can use the design of our model or the implemented system as input or as a preprocessing component within their research.

Finally, this research work suggests the following key points as a future work:

- Preparation of a balanced corpus that contains texts which represent different genres like theological and hymn books such as Synaxarium (the book of the saints of the Ethiopian Orthodox Church), deeds of the martyrs etc. and other books beyond religious scriptures such as fictions, textbook etc.
- Comparative study of three different approaches (CRF, SVM classifiers based and ANN based taggers for Ge'ez Language with more training and testing data)
- Extending this work by training in large corpus and using large tag sets that can identify gender, number, tense etc. with different feature set
- Comparison of two hybrid approaches: the hybrid of ANN and TnT tagger and the hybrid of TnT and CRF for Ge'ez language
- Morphological pattern analysis component of hybrid approach that proposed for Ge'ez POS tagger is based on unknown word guessing mechanism. Therefore, in order to further improve the tagging results, this approach can be extended to use the full feature of Ge'ez morphological analyzer.

References

[1] Allen James, Natural language understanding, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1995.

[2] "Chomsky-Definition," [Online]. Available: https://www.scribd.com/doc/22325162/Chomsky-Definition. [Accessed 02 11 2016].

[3] Liddy Elizabeth, "Natural Language Processing," in *Encyclopedia of Library and Information Science*, 2nd ed., New York, Marcel Decker, 2001.

[4] Binyam Gebrekidan, Part of Speech Tagging for Amharic, UNITED KINGDOM, 2009.

[5] Sisay Fissaha, "Part of Speech tagging for Amharic using Conditional Random Fields," *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages,* no. Association for Computational Linguistics, p. pages 47–54, June 2005.

[6] Jurafky and Martin, "Speech and Language Processing," in *An introduction to natural Language Processing, Computational Linguistics, and speech recognition.*, 2nd ed., New Jersey, Prentice Hall, 2009.

[7] Imad Zeroual, Abdelhak Lakhouaja and Rachid Belahbib, "Towards a standard Part of Speech tagset for the Arabic language," *Journal of King Saud University –,* vol. Volume 29, no. 2, pp. 171-178, 2017.

[8] Dat Quoc Nguyen, Son Bao Pham, Dang Duc Pham and Dai Quoc Nguyen, "Ripple down rules for part-of-speech tagging," in *CICLing'11 Proceedings of the 12th international conference on Computational linguistics and intelligent text processing*, Tokyo, Japan, February 20 - 26, 2011.

[9] Hadni Meryeme, Ouatik Said Alaoui, Lachkar Abdelmonaime and Meknassi Mohammed, "Hybrid Part-Of-Speech Tagger for Non-Vocalized Arabic Text," *International Journal on Natural Language Computing (IJNLC),* vol. Vol. 2, p. No.6, December 2013.

[10] Yemane Keleta, Yamamoto Kazuhide and Marasinghe Ashuboda, "Tigrinya Part-of-Speech Tagging with Morphological Patterns and the New Nagaoka Tigrinya Corpus.," *International Journal of Computer Applications,* vol. 146(14), pp. 33-41, July 2016.

[11] Teklay Gebregzabiher, "PART OF SPEECH TAGGER FOR TIGRIGNA LANGUAGE," Addis Ababa University, Addis Ababa, November, 2010.

[12] Zelalem Mekuria and Yaregal Assabie, "A Hybrid Approach to the Development of Part-of-Speech Tagger for Kafi-noonoo Text," no. Unpuplished, November 2013.

[13] Getachew Mamo and Million Meshesha, "Parts of Speech Tagging for Afaan Oromo," *International Journal of Advanced Computer Science and Applications,* no. 2011.010301, 2011.

[14] Leslau and Wolf, Comparative Dictionary of Ge'ez (Classical Ethiopic), Wiesbaden: Harrassowitz., 1987.

[15] Marvin Lionel Bender, "Language in Ethiopia," London, Oxford University Press, 1976, pp. pages 23-27 ; 99-106.

[16] Mahibre Kidusan ResearchCenter, "Ethiopian church studies,," *Journal of Ethiopian church studies,the Ethiopian Orthodox Tewahido church Sunday schools department,* 2010.

[17] Desta Berihu, *DESIGN AND IMPLEMENTATION OF AUTOMATIC MORPHOLOGICAL ANALYZER FOR GE'EZ VERBS,* Unpublished, November, 2010.

[18] Eric Brill, "A SIMPLE RULE-BASED PART OF SPEECH TAGGER," in *Applied Computational Linguistics (ACL)*, Trento, Italy, 1992.

[19] Roy Bar-Haim, Khalil Sima'an and Yoad Winter, "Part-Of-Speech Tagging of Modern Hebrew Text," *Natural Language Engineering,* vol. 14, no. 2, pp. 223-251, 1998.

[20] M. L. Bender, R. L. Cooper and C. A. Ferguson, "Language in Ethiopia: Implications of a Survey for Sociolinguistic Theory and Method," *Language in Society,* vol. 1, pp. 215-233, 1972.

[21] University of California, General history of Africa, London Berkeley: Heinemann Educational Books University of California Press, 1981.

[22] MERCER and Samuel Alfred Browne, ETHIOPIC GRAMMAR, OXFORD AT THE CLARENDON PRESS , 1920.

[23] Ludolf Hiob, Grammatica Aethiopica, Francofurti ad Moenum 1702, 1702.

[24] Herman Hupfeld, Exercise as an Ethiopian, Vogel, 1825, 1825.

[25] August Dillmann, Ethiopic Grammar, 2 ed., Wipf& Stoc, 2003, p. 581.

[26] ዓለማየሁ ሞገስ, ሰዋስወ ግዕዝ, Addis Ababa: Tesfa, ፲፱፻፶.

[27] "Notes on Ethiopic Localization," The Abyssinia Gateway, 22 07 2013. [Online]. Available: https://web.archive.org/web/20160317063000/http://abyssiniagateway.net/fidel/l10n/. [Accessed 28 03 2017].

[28] ZeraDawit Adhena, መርኖ ሰዋስወ ዘልሳነ ግእዝ, Addis Ababa, 2008.

[29] Anderson and Stephen, "Morphology," Yale University, New Haven, Connecticut, USA, [Online]. Available: https://cowgill.ling.yale.edu/sra/morphology ecs.htm. [Accessed 12 09 2017].

[30] Anand Kumar, "Morphology Based Prototype Statistical Machine Translation System," Amrita School Of Engineering Amrita Vishwa, Tamilnadu, India, 2013.

[31] DestaBerihu, "DESIGN AND IMPLEMENTATION OF AUTOMATIC MORPHOLOGICAL ANALYZER FOR GE'EZ VERBS," Unpublished, Addis Ababa, November, 2010.

[32] Fahim Muhammad, Naushad UzZaman and Mumit Khan, Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla, Bangladesh: Springer Netherlands, 2007, pp. 121-126.

[33] Muhammad Fahim, Khan, MumitUzZaman and Naushad, "Comparison of different POS tagging techniques for some South Asian languages," BRAC University, 2006, 2006.

[34] Deepika Kumawat and Vinesh Jain, "POS Tagging Approaches: A Comparison," *International Journal of Computer Applications (0975 – 8887),* vol. Volume 118 – No. , May 2015.

[35] Rabiner Lawrence, *A Tutorial on Hidden Markov Models and Selected Application inSpeech Recognition,* vol. 77, New Jersey: In: Proceeding of the IEEE, 1989.

[36] Manoj Kumar, "Stochastic Models for POS Tagging," Bombay.

[37] Thorsten Brants, "TnT: a statistical part-of-speech tagger," in *ANLC '00 Proceedings of the sixth conference on Applied natural language processing*, Seattle, Washington, 2000.

[38] Andrew MacKinlay, *The Effects of Part-of-Speech Tagsets on Tagger Performance,* University of Melbourne, 2005.

[39] Karthik Kumar, Sudheer and Avinesh, "Comparative Study of Various Machine Learning Methods For Telugu Part of Speech Tagging," in *In Proceedings of the NLPAI Machine Learning 2006 Competition*, 2006.

[40] Sisay Fissaha, "Part of Speech tagging for Amharic using Conditional Random Fields," *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages,* vol. Association for Computational Linguistics, p. 47–54, 2005.

[41] Guilder Linda, "Automated Part of Speech Tagging: A Brief Overview," Georgetown University, 1995.

[42] Robin, "NATURAL LANGUAGE PROCESSING," 15th December 2009 . [Online]. Available: http://language.worldofcomputing.net/pos-tagging/rule-based-pos-tagging.html. [Accessed 21 Feb 2017].

[43] Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach., New Jersey: Pearson Educational, 1995.

[44] Martha Yifiru and Wolfgang Menzel, "Amharic Part-of-Speech Tagger for Factored Language," in *International Conference RANLP 2009 - Borovets,*, Bulgaria, 2009.

[45] Dictionary and American Heritage, "punctuation," American Heritage® Dictionary, [Online]. Available: http://www.thefreedictionary.com/punctuation. [Accessed 28 5 2017].

[46] Thorsten Brants, "TnT -- Statistical Part-of-Speech Tagging," Universität des Saarlandes , [Online]. Available: http://www.coli.uni-saarland.de/~thorsten/tnt/. [Accessed 21 8 2017].

[47] Van Halteren, Jakub Zavrel and Walter Daelemans, "Improving data driven wordclass tagging by system combination," in *International Conference on Computational Linguistics*, Montreal, 1998.

[48] Peter Brown, Peter deSouza, Robert Mercer, Vincent Della Pietra and Jenifer Lai, "Class-based n-gram models of natural language," *Computational Linguistics,* vol. 18, no. 4, pp. 467-479, 1992.

[49] Nitin Hardeniya, NLTK Essentials, MUMBAI: Packt Publishing, 2015.

[50] Tetsuji Nakagawa, Taku Kudoh and Yuji Matsumoto, "Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines," in *In Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, Tokyo, Japan, 2001.

Appendices

Appendix-1: POS tagged Ge’ez text

ወሶበ/C ርእየት/V ራሔል/N ከመ/C ኢትወልድ/V ለያዕቆብ/PREN ቀንአት/V ላዕለ/PRE ልያ/N እኅታ/N ወትቤሎ/CV ለያዕቆብ/PREN ሀበኒ/V ውሉደ/N ወእማእኮ/CV እመውት/V ።/PUNC

ወለኬሮዖስ/CPREN እለ/C በአድባረ/PREN ሴይር/N እስከ/C ጠረሜስ/N ወፋራን/CN እንተ/C ውስተ/PRE ሐቅል/N ።/PUNC

እስመ/C ሰምዐ/V አብርሃም/N ቃለ/N ዚአየ/PRO ወዐቀበ/CV ትእዛዝየ/N ወኵነኔየ/CN ወሕግየ/CN ።/PUNC

ወይቤሎ/CV እግዚአብሔር/N ጽራሖሙ/N ለሶዶም/PREN ወለኀሞራ/CPREN በዝኀ/ADJ ወኀጣውኢሆሙኒ/CN ዐብየት/V ጥቀ/ADV ።/PUNC

ወፀንሳ/CV ከልኤሆን/ADJ አዋልዲሁ/N ለሎጥ/PREN እምአቡሆን/PREN ።/PUNC

ወተካየዱ/CV ኪዳነ/N በኀበ/PRE ዐዘቅተ/N መሐላ/N ወተንሥኡ/CV አቤሜሌክ/N ወአኮዘት/CN መጋቤ/N መራዕይሁ/N ወፋኮል/CN መልአክ/N ሰርዌሁ/N ወተመይጡ/CV ውስተ/PRE ምድረ/N ፍልስጥኤም/N ።/PUNC

ወርእያ/CV እግዚአብሔር/N ለምድር/PREN ከመ/C ማሰነት/V ወከመ/C አማሰኑ/V ፍኖቶ/N በዲበ/PRE ምድር/N ኵሉ/ADJ ዘሥጋ/CN ።/PUNC

ወአምኖ/CV አብራም/N ለእግዚአብሔር/PREN ወተኰለቆ/CN ሎቱ/PRO ጽድቀ/N ።/PUNC

ለእመ/PREC ይቤለኒ/V ዘሕብር/CN ይኩን/V ዐስብከ/N ወይወልዳ/CV ኵሎን/ADJ አባግዕ/N ዘሕብር/CN ወለእመ/CPRE ይቤለኒ/V ጻዕዳ/ADJ ይኩን/V ዐስብከ/N ይወልዳ/V ኵሎን/ADJ ጸዐዳ/ADJ ።/PUNC

ወእንዘ/C ዘንተ/ADJ ይትናገሩ/V በጽሐት/V ራሔል/N ምስለ/PRE አባግዐ/N አቡሃ/N ።/PUNC

ወይእዜኒ/CADV ወልድየ/N ስምዐኒ/V ቃልየ/N ዘእብለከ/CV ወተንሥእ/CV ሑር/V ማእከለ/PRE አፍላግ/N ኀበ/PRE ላባ/N እኁየ/N ውስተ/PRE ካራን/N ።/PUNC

ወትተከል/CV ዐጸደ/N ወይን/N ወትትጌበሮ/CV ወኢትሰቲ/CV እምነ/PRE ወይኑ/N ወኢትትፌሣሕ/CV እምኔሁ/PRE እስመ/C ዕፄ/N በልዖ/V ።/PUNC

እስመ/C እባርካ/V ወእሁባ/CV ውሉደ/N እምኔከ/C አሕዛብ/N ወነገሥተ/CN አሕዛብ/N ይወጽኡ/V እምኔሃ/C ።/PUNC

ወይቤሎ/CV ይስሐቅ/N ቅረበኒ/V ወእግስስከ/CV ወልድየ/N ለእመ/PREC አንተኑ/PRO ውእቱ/V ዔሳው/N ወለእመ/CPRE ኢኮንከ/V ።/PUNC

ወኮነ/CV ኵሉ/ADJ መዋዕሊሁ/N ለለሜክ/PREN ፸፻ወ፯/CR ዓመተ/N ወሞተ/CV ።/PUNC

ወይቤሎሙ/CV ፈርዖን/N ለኵሉ/PREADJ ሰብኡ/N ቦኑ/V እንከ/ADV ዘተረክበ/CV ከመ/C ዝንቱ/ADJ ብእሲ/N ዘመንፈሰ/CN እግዚአብሔር/N ላዕሌሁ/PRE ።/PUNC

Appendix-2: Ethiopic Alphabet

		a [ə] or [a]	u	i	a	e	ə [ɨ]	o	wa	ya [jə]
Hoy	h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ		
Läwe	l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ	
Ḥäwt	ḥ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ	
May	m	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ	ሸ
Śäwt	ś	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ	
Rəʾs	r	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ	ሯ	ሸ
Sat	s	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ	
Ḳaf	ḳ	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቋ	
Bet	b	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ	
Täwe	t	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ቷ	
Ḫarm	ḫ	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኋ	
Nähas	n	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ	ኗ	
ʾÄlf	ʾ	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ	ኧ	

		a [ə] or [a]	u	i	a	e	ə [ɨ]	o	wa	ya [jə]
Kaf	k	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኳ	
Wäwe	w	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ		
ʿÄyn	ʿ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ		
Zäy	z	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ	ዟ	
Yämän	y	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ		
Dänt	d	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዷ	
Gäml	g	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጓ	
Ṭäyt	ṭ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ጧ	
Ṗäyt	ṗ	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ	ጷ	
Ṣädäy	ṣ	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ	ጿ	
Ṣäppä	ṣ́	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ		
Äf	f	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ	ፏ	ፘ
Psa	p	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ፗ	

Appendix-3: List of Labiovelars in Ge'ez

	ä	I	a	e	ə
kʼʷ	ቈ	ቍ	ቋ	ቌ	ቊ
ḫʷ	ኈ	ኍ	ኋ	ኌ	ኊ
kʷ	ኰ	ኵ	ኳ	ኴ	ኲ
gʷ	ጐ	ጕ	ጓ	ጔ	ጒ

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:

Signature:

Date:

Confirmed by advisor:

Name:

Signature:

Date: