
ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

# *Word Sequence Prediction Model for Tigrigna Language*

SENAIT KIROS BERHE

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

ADDIS ABABA, ETHIOPIA

June 2020

# Addis Ababa University

# College of Natural and Computational Sciences

**SENAIT KIROS BERHE**

**ADVISOR: YAREGAL ASSABIE (PhD)**

This is to certify that the thesis prepared by *Senait Kiros Berhe, titled: word sequence prediction model for Tigrigna language* and submitted in partial fulfilment 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:** | | |
| **Examiner:** | | |
| **Examiner**: | | |

# ABSTRACT

Data entry is an important aspect of human computer interaction. It can be performed through the use of a keyboard, or other means. Writing text for work, study or communicating is frequent and time-consuming activity for most computer users. Better data entry performance can be obtained using Word prediction systems. Word Prediction is the task of forecasting words that are expected to follow a given fragment of text. Word prediction software is mainly used to minimize keystrokes for different users especially for people with disabilities, for people having limited language proficiency, for people with frequent spelling errors and for non-native users. Huge volume of Tigrigna documents are being written and made available on the Internet. In this study we designed and developed a word sequence prediction model for Tigrigna language. This is done using n-gram statistical models based on two Markov language models, one for tag, the other for words which are developed using manually tagged corpus, and grammatical rules of the language.

The designed model is evaluated based on a precision evaluation metric that is used to evaluate performance of the system. According to our evaluation, On the average 85 % performance of correctly predicted words are obtained using Sequence of two tags and 81.5 % performance of correctly predicted words are obtained using Sequence of Three tags. According to our result, Word prediction using Sequence of two tags provides better performance than Sequence of Three tag.



**Keywords**: Word prediction, Natural Language Processing, Statistical language modelling, POS tagging, Precision

## Dedication

I am dedicating this thesis to:

**Dad:** Even if it has been a year since you died. I didn't forget your unconditional love and concern to all of your children especially to me. I am grateful for being your daughter.

**MOM and Daughter:** Birhan Bishu and Eliana Dereje, my mom you are my super hero to me and my daughter you are my blessing. I thank you God for giving me those two precious gifts of my life.

# Acknowledgment

All the praise goes to Almighty God and his mother Saint Mary with Saint Gabriel for giving me the strength that I need to accomplish my duty despite of many obstacles I face before.

Firstly, I am sincerely grateful for my advisor Dr. Yaregal Assabie for his concern, supervision, and encouragement and for providing me with valuable comments starting from the title selection to this end. Beside my advisor, I would like to thank all my instructors especially Dr. Solomon Atnafu for giving me moral support and encouragement.

Secondly, I would like to thank Addis Ababa University, department of Computer Science for giving me an opportunity to complete this work. Especially Haymanot Yirga, secretary of the department, for her kind approach and encouragement starting from the beginning towards my thesis.

Thirdly, I would like to thank Wollo University for helping me with financial support, facilities, when it is needed; making the environment suitable to work with colleagues, especially the former leader of the campus Dr. Ahmedin Mohammed.

Fourthly, Special Thanks to my friends and colleague Abrha G/kiros and Kibrom Haftu who were involved in my thesis work in reviewing each chapter and comprehensive guidance in the progress of my thesis. In addition to those mentioned friends, I am sincerely grateful for my friend Shashu Birhane, she helps me by providing her e-video in order to use internet service and suggesting some comments and gives me moral support and Hagerie tulu for helps me with reviewing each chapter and providing valuable comments.

Last but not List, I am grateful to my families especially my mother for her continuous motivation and encouragement throughout my life; my husband Dereje Girma, my daughter Eliana, my sisters, Mihret, Merhawit and Hiwot and, my brothers Leul and Temesgen, who have rendered me all their care, love and encouragement. Without their sustainable support, I couldn't have reached this level of my achievement. In addition to that, Mihret and Temesgen were involved in my thesis work in editing my errors.

.

## Table of Contents

# List of Figures

# List of Tables

# List of Algorithms

## Acronyms and Abbreviations

| | |
|---|---|
| AAC | Augmentative and Alternative Communication |
| CFW | Correctly flagged words |
| KE | Effective Number of Keystroke s |
| KSS | Keystroke Saving |
| KT | Total Number of Keystroke s |
| KUC | Keystroke Until Completion |
| NER | Named Entity Recognizer |
| NLP | Natural Language Processing |
| POS | Parts-of- Speech |
| PDA | Personal Digital Assistance |
| SMS | Short Message Service |
| SOV | Subject-Object-Verb |
| SVO | Subject-Verb-Object |
| TW | Tagged Word |
| WFW | Wrongly Flagged words |
| WPS | Word Sequence Prediction |

# CHAPTER ONE: INTRODUCTION

## 1.1 Background

Natural Language Processing (NLP) is a field of Computer Science that investigates interactions between computers and human languages [1].It is used for both generating human readable information from computer systems and converting human language into more formal structures that a computer can understand [1].Therefore, it is important for scientific, economic, social, and cultural reasons. As its theories and methods are used in a variety of new language technologies it counts rapid growth. Due to this, it is essential to have NLP as a working knowledge for many peoples .Within industry, this includes people in human-computer interaction, business information analysis, and web software development. Within academia, it includes people in areas from humanities computing and corpus linguistics through to computer science and artificial intelligence [4]. Morphological analysis, part of speech tagging, word sense disambiguation, and machine translation are common problems of NLP [1].

Data entry is an important aspect of human computer interaction. It can be performed through the use of a keyboard, or other means. Writing text for work, study or communicating is the most frequent and time-consuming activity for most computer users. Better data entry performance can be obtained using Word prediction systems that help to improve the writing methods of users [2, 3].

Word Prediction is the task of forecasting words that are expected to follow a given fragment of text. Word prediction software is writing support software: which suggests a list of meaningful predictions at each keystroke. From those lists the user can possibly identify the word he/she is willing to type and then, the software will automatically complete the word being written, thus saving keystrokes [12].

Word prediction software is mainly used to minimize keystrokes for different users especially for people with disabilities, for peoples having limited language proficiency, for people with frequent spelling errors and people learning a second language or non-native users. In addition to this in computer aided education, it helps children by suggesting appropriate words of the language when they need to learn using computers to develop their writing skills [3].

NLP contains processing noisy data and checking errors rather than predicting the next word or text in a sentence for semantic understanding solely. For example, noisy data can be produced in speech or handwriting recognition, as the computer may not properly recognize words due to unclear speech or handwriting that differs significantly from the computer's model. Additionally, NLP could be applicable as spell checking in order to catch errors in which no word is misspelled but the user has accidentally typed a word that she or he did not intend. In the sentence "I picked up the phone to answer her fall," for instance, "*fall*" may have been the intended word, but it is more likely that "*call*" was simply mistyped. A spell checker cannot catch this error because both "*fall*" and "*call*" are English words. An NLP algorithm that could catch this error would thus need to look beyond what letters from words and instead attempt to determine what word is most probable in a given sentence [5].

Morpho-syntactic information is obtained by combining sequentially the two basic models of prediction that can either be based on text statistics or linguistic rules. The two basic Markov models can be included: one for word classes (POS tag unigrams, bigrams and trigrams) and one for words (word unigrams and bigrams). Those two models are helpful in force prediction accuracy of the system [14].

## 1.2 Motivation

Tigrigna is the official language of Tigray Region of Ethiopia and it is also one of the two official languages in Eritrea. It is a Semitic language spoken by about 7 million people around the world [15].With the advent of computers, huge volumes of Tigrigna documents are being written, produced and made available on the Internet and the World Wide Web which are accessible for users. However, writing Tigrigna text is difficult due to the need of pressing up to three keys to write a single character**.** We believe that word sequence prediction models for Tigrigna language would help computer users to write Tigrigna text efficiently. This has motivated us to develop a word sequence prediction model for Tigrigna language in order to help the users of the language on their text input method.

## 1.3 Statement of the Problem

A number of word prediction and word sequence researches have been attempted for different languages. Amharic [11, 42], Afaan-Oromo[52] are categories under Word sequence prediction . English [17], Hebrew [8], Basque [7], Italian [9, 11], Swedish

[48] and Urdu [60] are some of the languages for which researches on Word prediction have been conducted by different scholars in the past years. Those Word prediction and Word sequence prediction researches have been developed depending on various characteristics of the language such as morphological complexity and sentence structure of the language. These attempts to create good communication between a human and computer by reducing the time and effort to write a word for slow typists by reducing keystrokes, or people who are not able to use a conventional keyboard in addition to common computer users.

Word prediction is a challenging task especially for inflected languages. Inflected languages are languages that are morphologically rich and have massive word forms. i.e. one word can have different forms. As Tigrigna language is a highly inflected language and morphologically rich it shares this problem. So, storing all forms in a dictionary won't solve the problem as in English and other less inflected languages. This problem makes word sequence prediction models for Tigrigna language much more difficult [22].

Research on Amharic word sequence prediction has also been conducted for Amharic language by Tigist Tensou [11] and word prediction for Amharic online handwriting recognition by Nesredien Suleiman[42] developed. Even if Amharic and Tigrigna have similar linguistic characteristics, they still have basic differences on morphological and syntax processing. To our best knowledge, there is no any research conducted on the topic of word sequence prediction for Tigrigna language. Thus, the purpose of this work is to develop a word sequence prediction model for Tigrigna language with inclusion of context information.

## 1.4 Objectives

### General Objective

The general objective of this research is to design and develop word sequence prediction models for Tigrigna text.

### Specific Objectives

The following specific objectives are identified in order to achieve the specified general objective:

- Review literature for various languages and supplementary researches performed on Tigrigna text
- Collect corpus of Tigrigna text
- Study the grammatical structure of Tigrigna language
- Design Tigrigna word sequence prediction model
- Develop a prototype of the system
- Evaluate performance of the system

## 1.5 Methodology

The following methods will be applied in order to achieve the above specified objective:

**Literature Review**

Different researches and related works that are considered to be relevant for this research were reviewed to take hold of fundamental concepts for the intended study and to propose an appropriate model for the Tigrigna word sequence prediction model. Some of the literature and related works will be reviewed in detail.

**Data Collection**

We use a simple random sampling method to collect the sample data. A corpus has training datasets and testing datasets that contain sentences used to train and test our system performance of predicting a word. For both training and testing purposes 3,000 sentences or around 14,270 words are collected from three sources of areas such as magazines, books, and websites of each 1,000 sentences or manually. To improve the ways of predicting a word, we added for each sentence manually prepared features of the POS tagged and grammar checker.

**Development environment and tools**

In this thesis, we will use tools such as C# programming language, Microsoft Visual Studio, Notepad++, SQL, and SharpNlp toolkit.

**Testing**

Performance evaluation could be conducted manually by comparing the system‘s next word sequence predicting ability with the manually prepared sample as a Testing document. POS tagged test data will be used and the prediction activity is evaluated using the evaluation metric recall, precession, and F-score measure.

## 1.6 Scope and Limitation

This research aims to develop a word sequence prediction system for Tigrigna text. Obviously, Word Sequence Prediction is a very complex and difficult task which needs understanding of natural language techniques and requires a number of Natural language processing tools such as Sentences parser, Part of Speech (POS) tagger, Grammar Checker, Stemmer, and Named Entity Recognizer (NER) and so on. Even though some of the NLP tools have been developed by some researchers, they are not freely available for integrating with our system. Having these limitations, our scope is limited to predict the next word after analysing sequences of words in a given simple Tigrigna sentence by checking the grammatical arrangement which is “single word” than phrases or other sentences in a standalone platform.

## 1.7 Application of Results

The main contribution of this research is on finding an efficient method of word prediction that will benefit common computer users of the Tigrigna language. This supports the user in helping to determine and choose the correct spelling of a word. Therefore, the proposed model will be helpful to increase the typing rate and reduce spelling errors. In addition to the common computer users of the language, it will benefit people with severe motor and oral disabilities, on handwriting recognition, mobile phone/PDA texting etc. Thus, the output of this research work will be an important contribution to the overall development of Tigrigna language technology.

## 1.8 Thesis Organization

The remaining parts of this thesis are organized as follows. In Chapter Two, the literature review part briefly states fundamental concepts of word prediction, methods of word prediction, writing system of Tigrigna language and its grammatical rules. Chapter Three, presents research conducted by different scholars on the topic of word sequence prediction, their approach, and findings. In Chapter Four, the proposed word sequence prediction model, approach, algorithm and architecture are explained.

Implementation and experimentation of the proposed word sequence prediction model is presented in Chapter Five. Finally, the conclusion and recommendation of the thesis are stated in Chapter Six.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Introduction

This chapter focuses on the fundamental concept of word sequence prediction and its ideas associated with the Tigrigna language. The writing system of Tigrigna language including parts of speech, morphological characteristics, and grammatical structure of the language are discussed in respective sections of this chapter. Finally, different approaches used in Word sequence prediction like statistical, knowledge-based, and heuristics are presented in order to understand the concepts of this study.

## 2.2 Writing System of Tigrigna Language

Tigrigna is the official language of the Tigray Region of Ethiopia and it is also one of the two official languages in Eritrea. It is a Semitic language spoken by about 7 million people around the world. It uses the Geez script nowadays named Ethiopic script which is originally developed for Geez language [15]. A single alphabet in Tigrigna language represents vowel and consonant combination that indicates the language is syllabic. [29, 30].

### 2.2.1 Part of Speech of Tigrigna Language

POS tagging focuses on assigning an appropriate word class for each word. POS tagging and POS n gram modelling are mostly used in researches related to word sequence prediction to give synthetic information and better prediction [7, 8].

Tigrigna language has eight parts of speech. These are grouped as nouns, adjectives, verbs, adverbs, Conjunction, prepositions, pronouns, and Interjection (exclamation) to describe in detail a particular class of a word in a given sentence [28].

**Nouns**

Tigrigna nouns are words that are used for labelling things, used to name or identify a class of things, peoples, places or ideas. Such as a real thing (for example, ድመ/Demu (Cat), an imaginary thing (for example, መንፈስ/Menfes/Ghost/, an idea (for example, ፍቅሪ/Fikri(Love), person name ((for example,ሃና “Hana”)). They are typically used as an arguments, subjects and objects of transitive verbs and complements of prepositions.

**Pronouns**

Tigrigna pronouns are words or morphemes that can be used in place of nouns. That is limited in number and categorized in different sub categories. Those categories can include personal pronouns, possessive pronouns, interrogative pronouns and demonstrative pronouns.

**Personal pronoun** represents the speaker, listener and third party in any speech. This is illustrated in Table 2.1 to indicate the personal pronoun.

**Table 2.1**: Personal Pronouns in Tigrigna Language

| **Personal Pronouns** | | | |
|---|---|---|---|
| **Tigrigna** | **English** | **Number** | **Gender** |
| አነ /ane | I | Singular | Neutral |
| ንስካ/nsʿka | You | Singular | Masculine |
| ንስኪ/ nsʿki | You | Singular | Feminine |
| ንሱ/nʿsu | He | Singular | Masculine |
| ንሳ/nsʿa | She | Singular | Feminine |
| ንሕና/nhena | We | Plural | Neutral |
| ንስካትኩም/nskatkum | You | Plural | Feminine |
| ንስካትክን/ nskatkʿn | You | Plural | Masculine |
| ንሳቶም/nʿsatom | They | Plural | Feminine |
| ንሳተን/nʿsaten | They | Plural | Masculine |

**Possessive pronoun** is a pronoun that is used for indicating possession, for example, ናተይ/natey (*mine).*

**Interrogative pronouns** are pronouns used in questions.

**Demonstrative pronouns** indicate objects in reference to the place it is found. The indicated object can be found near or far from a person indicating the object or for the observant. Therefore these kinds of pronouns are classified based on their distance as well as based on the indicated objects gender [37]. Table 2.2 shows examples of demonstrative pronouns.

**Table 2.2:** Demonstrative Pronouns in Tigrigna

| **Number** | **Gender** | **Near** | **Far** |
|---|---|---|---|
| **Singular** | Masculine | እዚ/ezi/ This/1st thing | እቲ/eti/ That/2nd thing |
| | Feminine | እዚኣ/eziaʹ/1st thing | እቲኣ/etiaʹ/2nd thing |
| **Plural** | Masculine | እዚኦም/ eziomʾ/These | እቲኦም/etiomʹ/ Those |
| | Feminine | እዚአን/ezʹn/እዚኣተን/ezaʹten | እቲኤን/etiaʹn/እቲኣተን/etiaʹtn |

**Verb**

Tigrigna verbs can be described as a word used to show that an action is taking place, a word to indicate the existence of a state or condition, or part of speech to which such a word belongs.

Tigrigna verbs carry inflections of aspect and mood and hence are morphologically the most complex POS in Tigrigna. A lot of words with other POS are derived primarily from verbs. There are two major approaches to identify verbs from other word categories: syntactical and morphological approach. In the former case, verbs function as predicates in a simple sentence and they are found at the end of a sentence. In the latter case, they reflect grammatical categories such as aspect, mood and agreement [35].

Verbs in Tigrigna could be taken as thorough working out of a method in which a root, consisting usually of three radicals, is expressed in patterns which carry different meanings and applications of the verbal idea that belongs to the root. These patterns form subordinate systems with a relatively consistent emphasis of their own, such as “passive”, “reflexive “ “intensive “and so forth. The inflections of these systems give tense and specify the person and number of subjects and objects [33].The simplest type of a verb is generally 3rd person masculine, singular, and simple perfect tense, e.g. ሰበረ

he broke. This form is called a" root form" of a verb and it is used as the name of the verb.

**Adjectives**

Tigrigna adjective is a word that describes or qualifies a noun or pronoun it modifies. Objects are differentiated from one another by different attributes like shape, behaviour, colour etc….and this difference is described using adjective word class. Adjectives are inflected for gender, number and case in a similar fashion to nouns [37, 38].

The adjective precedes the noun or pronoun which it modifies and agrees with it in gender and number. Most languages appear to identify two open classes namely nouns and verbs. However Tigrigna has got an additional open class, the Adjective class. Those adjectives are too many in number and increase from time to time. [32].

Adjectives in Tigrigna usually precede the nouns that they modify or describe. Here is a simple example. **ነዊሕ ወዲ ፈትየ።** /newiH wedi fetye. (I loved a tall boy.)

In this example, the adjective **ነዊሕ**/newiH (tall) precedes the **ወዲ**/wedi(Boy) which it modifies. But this does not mean that a word is an adjective just because it precedes a noun. For instance, in the sentence **እቲ ወዲ ቆንጆ እዩ።**/^ti wedi qonjo ^yu (The boy is handsome.), the word እቲ/^ti (the) precedes the noun **ወዲ** /wedi (boy).Although the word **እቲ**/^ti (the) functionally shares the feature of an adjective, modifier, it is a demonstrative pronoun.

**Adverb**

An adverb is a word that modifies a verb, adjective, sentences or clauses and other adverbs. In many languages adverbs are classified as open classes; however, Tigrigna's adverbs are classified as closed classes [32].Modifiers of verbs or verb phrases usually express time, place, manner etc. Modifiers of adjectives and adverbs commonly express degree while adverbs functioning as sentence modifiers usually express the speaker's attitude regarding the event spoken.

Example: **ሎሚ ቅነ**/lomi qene /Very recently, soon

**ሎሚ ዘበን**/lomi zeben /This year

**ፅቡ ቕ ሰሪሑ**/tsebuk serihu /He did well

In the sentence, ”**ሎሚ ክዳውንቲ ሓጺበ**/”lomi kedawnti haTsibe”/” I washed clothes today”, the word “**ሎሚ**”/”lomi”/” today” is an adverb that modifies the main verb “**ሓጺበ**”/”haTsibe”/” washed”. It tells more about when I washed clothes, which is today.

There are adverbs which may be used in the gerundive as finite verbs, but which would naturally be translated with adverbs.

**አፀቢቆም ይፅሕፉ።።**/atsebikom yitsehfu/They are writing neatly.

**ቀልጢፎም ገይሮም።።**/qeltifom geyrom/They did it quickly

**Conjunction**

Conjunctions are words that link words, phrases and clauses to create larger grammatical units. They are limited in number and can be used with verbs, nouns and adjectives. Some of the conjunctions of Tigrigna language are **ን**, **ኮይኑግና**, **ስለዝኮነ** e.g., **ሰብአይንሰበይቲን/** seb^ay-**n** sebeyti-**n** (husband and wife)

**Preposition**

Prepositions are a small set of words, which will have meanings only when they are used with other words such as nouns, verbs, pronouns and adjectives. They can express relationships between person, thing, or event etc and another.

Common Tigrigna Prepositions: - **ምስ**/m’s, **ኣብ**/a’b ,**ናብ**/na’b, **ካብ**/ka’b ,**ክም**/ka’m….They are not inflected for gender, person, number etc[32, 39].

Tigrigna prepositions have the following central property:

- They take nouns or adjectives as their complements: e.g. **ብሃንደበት**/bhandebet (Suddenly)
- They can stand alone as a separate word: e.g. **ናብ ቤት ትምህርቲ** /nab

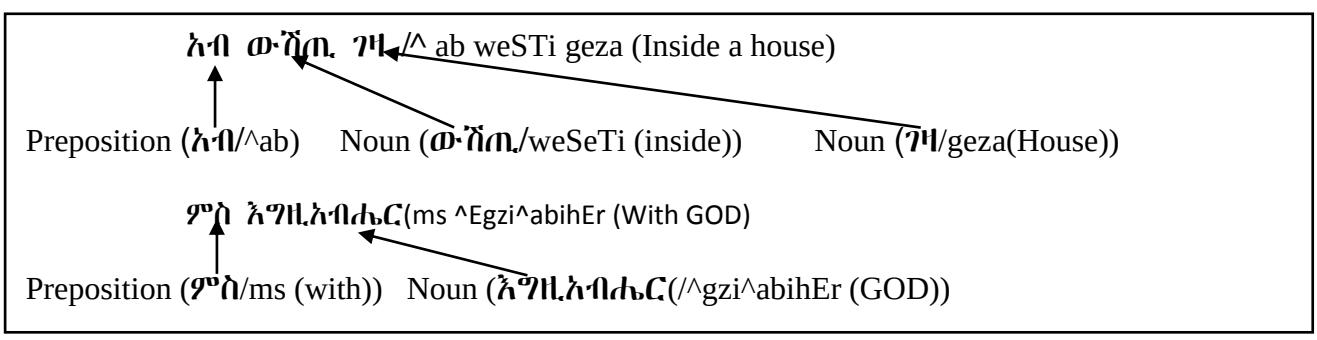


**Figure 2.1:** *Prepositions as a separate word*

Most of the items that function as prepositions in Tigrigna can also function as conjunctions. Moreover the coordinating conjunctions such as - ን/n (and), ወይ/wey (or) can form structural units with the nominal and prepositional phrases they precede or follow [35].

### Punctuation

Tigrigna punctuation marks are word separator marks (:) is used in the old literature to separate one word from other words. In the current literature, it is rarely used. As a result a single space is used to separate words instead of punctuation marks. Most of the Tigrigna language punctuation marks are listed in Table 2.3.

**Table 2.3:** List of Tigrigna punctuation marks

| **Punctuations** | **Meaning** |
|---|---|
| ፡ | Word separator(Ethiopic word Space) |
| ። | End of sentence(Ethiopic full stop) |
| ፤ | Sentence connector(Ethiopic Semicolon) |
| ፥ | Beginning of the list mark(Ethiopic preface colon) |
| ? | End of question |
| ! | End of an emphatic declaration, or command. |
| ” | Quote some words or sentences taken from other |

## 2.2.2 Morphology of Tigrigna Language

The Tigrigna language is morphologically rich both inflectional and derivational. Dictionaries define morphology as the structure of words in a language including patterns of inflections and derivation. Morpheme is the minimal unit of morphology which includes root/stem form and other meaningful parts of a word [37, 38, 40].

To define morphological structure of a Tigrigna word, it is needed to identify seven vowel sounds which are usually called ‘orders’ and within five-ordered alphabets which are variants of some of the basic 35 consonants. Altogether 275 symbols constitute the Ethiopic alphabet chart known as ‘Fidel’.

Due to its morphological richness, Tigrigna exhibits the root and stem pattern morphological phenomenon. Root is a verb that indicates a third person singular masculine, such as **በልዐ፣ሰተየ፣ሰርሐ**,etc. and the noun that expresses a singular noun whereas a stem is a verb in which its ending letter is ‘sads’ (6th order) or a noun that indicates a singular number. The morphological variation is the result of adding suffixes to the root verbs or nouns to indicate number, gender, tense, possession. etc. For this reason; if the word is a verb, a stem may or may not have a meaning [28, 31].

Tigrigna verbs are inflected for gender, person, number, case, tense, aspect, mood, etc. Tense-aspect-mood is expressed by affixes that are prefixed, infixed or suffixed to the root word. The verbs have a ‘root-template’ pattern which predominantly consists of trilateral consonants.

Subject-verb agreement is enforced by alteration of verb suffixes and/or prefixes. Moreover, negating Tigrigna verbs, nouns or adjectives also require circum-fixing the morpheme **ኣይ...ን**/ayI...nI/. /. Grammatical clitics that are attached to words further add to the complexity of word morphology. These clitics are mostly pro-clitics and enclitics of prepositions, conjunctions, possessives and object pronouns [36].

One of the reasons that Tigrigna morphology becomes complex is; sometimes the language follows its own grammatical structure, sometimes it follows from geez and borrowed Arabic, Italian and other languages. For example, **ከልቢ** /kelbi/Dog/ / in a singular form becomes both **አካልብ**/akalb/dogs from Geez and and **አክላባት**/aklabat in plural form.[28].

**Inflectional Morphology of Tigrigna Language**

Inflection is a morphological variation which changes the grammatical function of the sentence without affecting its parts of speech and general meaning. Since Tigrigna language is a highly inflectional language, a given root of a Tigrigna word can be found in different forms.

Nouns, verbs, and adjectives can be marked for person, gender, number, case, definiteness, and time. Gender, number and case marker suffixes are used in these inflections of nouns. Verbs are inflected for person, gender, number, and time with the basic verb form being third person masculine singular. The perfect tense normally expresses past tense. Prefixes are used for first, second, and third person future forms and suffixes are used to indicate masculine and feminine subjects, respectively. Adjectives are inflected for gender, number, and case in a similar fashion to nouns [37, 38].

Affixing is used to derive nouns by adding prefixes, infixes or suffixes to basic nouns, adjectives, verbs, stems and roots. In Tigrigna morphemes can be free or bound; where free morphemes as their name indicates can give complete meaning by themselves whereas to give a meaningful sentence bounded morphemes connected with free morphemes.

**Inflection of Verbs**

Tigrigna verbs are found in perfective, imperfective, gerundive, jussive and imperative by employing affixes. The morphological variations of a perfective verb are formed by adding suffixes that mark (indicate) person, gender and number to the perfect verb stem. For example, **ከድኩ፣ከድካ፣ከድኪ፣ከድኩም፣ከደ፣ከደት፡፣ከድክን፣ከዱ፣ከዳ** are perfective verbs formed from the stem ከድ which means 'to go'. The other verb type is imperfective verb and is formed by affixing gender, person and number markers to the imperfective verb stem. For example, **እኽድ፣ት፡ኽድ፣ት፡ኽዲ፣ ይኽድ፣ንኽድ፣ት፡ኽዱ፣ት፡ኽዳ፣ይኽዱ** and **ይኽዳ** are imperfective verbs that illustrate how the affixes are used to inflect the imperfective verb stem. The gerundive form is inflected by adding suffixes at the end of the gerundive verb to indicate person, gender and number.

For example,

**ሰሪሐ፣ሰሪሕኻ፣ሰሪሕኺ፣ሰሪሑ፣ሰሪሓ፣ሰሪሕና፣ሰሪሕኹም፣ሰሪሕኽን፣ሰሪሖም፣ሰሪሐን** can show how the gerundive verb ሰሪሕ morphologically varies. Jussive and imperative verbs are sometimes called mood. Jussive verbs are used to express a command for first and third persons whereas imperative verbs are used to express second person in the singular and plural form. In addition to indicating person, gender, number and time, Tigrigna verbs express two tenses, namely, perfect and imperfect. Perfect tense

indicates completed actions whereas imperfect tense expresses uncompleted actions [28, 34].

Tigrinya verbs can be classified in "families" according to the nature of the radicals that they contain. Some of the families are subdivided into “two” according to whether the second type is “geminated” or “doubled”. Such doubling affects individual forms of a verb are derived from its basic consonant root. The presence of certain letters or sounds, not ably the laryngeals, also affects the production of forms. Verbs in a given class are changed in regular patterns to produce the tenses, aspects, persons, and so forth, that the verb forms can express. To conjugate a verb means to derive and list in order all the verb forms that belong to that verb. In English, the verbs themselves change very little. Conjugating is a matter of supplying auxiliary verbs and pronouns to a very few forms [33].

**Inflection of Perfect**

**Tense**

Perfect tense expresses a past tense and consists of a stem, pronoun suffix and subject marker. The subject marker shows the subject’s number, person and gender [12, 13]. This is illustrated in Table 2.4 to indicate the subject marker and the pronoun.

**Table 2.4**: Inflections of perfect tense

| **Verb Variations** | **Person** | **Gender** | **Number** | **Pronoun** |
|---|---|---|---|---|
| **ሰሪቹ** | Third | Male | Singular | He-**ንሱ** |
| **ሰሪቻም** | Third | Male | Plural | They-**ንሳቶም** |
| **ሰሪቸን** | Third | Female | Plural | They-**ንሳተን** |
| **ሰሪችኩም** | Second | Male | Plural | You-**ንስኻትኩም** |
| **ሰሪችኪ** | Second | Female | Singular | You-**ንስኺ** |
| **ሰሪችካ** | Second | Male | Singular | You-**ንስኻ** |
| **ሰሪችና** | First | Male and Female | Plural | We-**ንሕና** |
| **ሰሪቾ** | First | Male and Female | Singular | I-**ኣነ** |
| **ሰሪችክን** | Second | Female | Plural | You-**ንስኻትክን** |
| **ሰሪቻ** | Third | Female | Singular | She-**ንሳ** |

As it can be seen on Table 2.3, the suffixes attached are **ኡ**/u/, **ኦም**/om/, **አን**/en/, **ኩም**/kum/, **ኪ**/ki/,**ካ**/ka/, **ና**/na/, **አ**/ae/, **ክን**/kn/, ኣ/a/ and indicate person, gender and number of the subject and the pronoun that indicates the person[31].

**Inflection of Imperfect Tense**

This tense includes indicative, subjective, jussive and imperative forms and are inflected by adding affixes to indicate gender, person and number to the imperfective verb stem. We can consider the following table to see the affixation on the root verb "ሰርሐ". The Inflection of Imperfect Tense is shown in Table 2.5.

**Table 2.5:** Inflections of imperfect tense

| **Person** | **Gender** | **Singular** | **Plural** |
|---|---|---|---|
| Third | Female | **ትሰርሕ**/t'serh | **ይሰርሓ**/y'serha |
| Third | Male | **ይሰርሕ**/y'serh | **ይሰርሑ**/ y'serhu |
| First | Male ,Female | **እሰርሕ**/a'serh | **ንሰርሕ**/ n'serh |
| Second | Male | **ትሰርሕ**/tserh | **ትሰርሑ**/ t'serhu |
| Second | Female | **ትሰርሒ**/t'srhi | **ትሰርሓ**/ t'serha |

In Table 2.4, **ት**(t), **ይ**(y), **ኢ**(i), and **ን**(n) are prefixes and **እ**(e), **ኡ**(u), and **አ**(a) are suffixes. It is also possible to find the negative form of the above mentioned root by using **አይ**/ay/, **አይት**/ayt/, **አይን**/ayn/ as prefixes and **ን**/n/, **አን**/an,/**ኡን**/un/ as suffixes.

**Inflection of Nouns**

Like Tigrigna verbs, Tigrigna nouns inflect to show gender, person and number by adding affixes to the noun stem. Grammatically, Tigrigna language specifies two types of genders: Feminine and masculine. So, Tigrigna nouns are either male or female by nature [28]. Therefore, nouns are used affixes such as **-ኣ**/a/, **ታት**/-tat/, **ኣት**/-at/, **ኣን**/an/, **ውቲ**/-wti/, **ቲ**/-ti/, **ዊ**/wi/, **ና**/na/, etc…in order to express possession, pluralism, tribe and gender. Table 2.6 shows how these affixes are used to inflect Tigrigna noun.

**Table 2.6:** Infection of Nours

| Noun Stem | Affixed Used | After Affixation | Remark |
|---|---|---|---|
| ላሕሚ | ኣ | ኣላሕም | Pluralism |
| ፊደል | ኣት፡ | ፊደላት፡ | Pluralism |
| ቐዕሊ | ታት፡ | ቐዕልታት፡ | Pluralism |
| መምህር | ኣን | መምህራን | Pluralism |
| ገዛ | ው፡ቲ፡ | ገዛው፡ቲ፡ | Pluralism |
| ክልቢ | ና | ክልቢና | Possession |
| ስራሕ | ፡ቲ፡ | ስራሕ፡ቲ፡ | Pluralism |
| ኢትዮጵያ | ዊ | ኢትዮጵያዊ | Tribe |

**Inflection of Adjectives**

Tigrigna adjectives agree with their noun in number and gender. They inflect to indicate number and gender. This is to mean that Tigrigna adjectives can have singular masculine, singular feminine and the same adjective to express both masculine and feminine genders in the plural form [28]. **ት፡**/t/, **ቲ**/ti/, **ኣዊት፡** /awit/, **ኣዊ**/awi/, **ኣት፡**/at/, **ኣዊያን**/awyan/, etc. are used to inflect adjectives and are illustrated how they are used in the Table 2.7.

**Table 2.7**: Inflection of Adjectives

| Masculine | Feminine | Pluralisation |
|---|---|---|
| ዕቡ፡ቐ | ዕብ፡ቐ፡ቲ | ዕቡ፡ቓት፡ |
| ሰራሒ | ሰራሒት፡ | ሰራሕ፡ቲ |
| መዐረይ | መዐረይት፡ | መዐረይት፡ |
| ቀታሊ | ቀታሊት፡ | ቀተልቲ |
| ሰፋይ | ሰፋይት፡ | ሰፈይቲ |

**Derivational Morphology of Tigrigna Language**

Nouns can be derived by adding prefixes, infixes, or suffixes to basic nouns, adjectives, verbs, stems, and roots. Adjectives are derived from verbs, nouns, verbal roots, and stems by adding suffixes. Infixing is used when deriving adjectives from verbal roots and unlike other word categories, the derivation of verbs from other POS is not common [38]. Nouns, verbs, and adjectives can be marked for person, gender, number, case, definiteness, and time [37].

Derivational morphology deals with adding suffixes to words to bring a change in meaning and category, whereas inflectional morphology deals with adding suffixes to words so that the general meaning and category of the original word (stem) will not be changed.

In Tigrigna morphemes like /**ኣዊ** /Awi/, **ኣም/** am/, **ን/** n/ can change nouns of the language in to verbs and adjectives etc. For example, ***መርዓ/***" merea" /marriage / becomes ***መርዓዊ/***" mereawi " /groom/, **ሰንኪ/**"senki" /cause**/**, **ሰንካም/**"senkam" and **ሰበረ**/sebere/he breaks it/, **ኣይ-ሰበረ-ን**/"ay sebere n" /he didn't break/ are some examples of derivational morphologies.

## 2.2.3 Grammar of Tigrigna Language

Grammar is a set of structural rules governing the composition of sentences, clause, phrases, and words in a given natural language. And also grammar is the whole system and structure of a language or of languages in general, usually taken as consisting of syntax and morphology.

In Tigrigna's sentence, like other languages, segments and features can be divided into two parts namely subject and predicate [32]. Let us look at the following examples. The examples are written in Tigrigna alphabets followed by the transliterated form of the alphabets to their corresponding Latin characters [23, 32, 34]. **ሃውካ መፂኡ** /Hawka me^Tsi^u. Your Brother came (has come).

**ሃውካ ትማሊ ካብ ሸረ መፂኡ።** /Hawka tmali kab shire me^Tsi^u.

Your brother came (has come) from shire yesterday.

**እቲ መፅሃፍ ኣብ ልዕሊ ኣራት ግበሮ።** /xeti me^THaf xb l`li `arat gbero. Put the book on the bed.

In the first two sentences **ሃውካ**/Hawka (your brother) functions as a subject, while the rest parts of the sentences are used as a predicate **መጺኡ**/me^Si^u (came (has come)) and **ትማሊ ካብ ሽረ መጺኡ**/tmali kab shire me^Tsi^u (came (has come) from shire yesterday). The subject can be thought as a noun phrase that is used to express some objects or persons and the predicate is used to say something true or false about the subject [32, 34]. These two elements of sentences are known as noun phrases (NP) and verb phrases (VP) which are headed as noun and verb respectively. In the third sentence, the subject is established with the verb **ግበሮ**/gibero which is you third-person singular that acts as the noun phrase (NP) and the sentence **እቲ መፅሃፍ አብ ልዕሊ አራት ግበሮ**"ti me^THaf ^ab l`li `arat gbero" acts as a verb phrase (VP) which in turn is divided into a noun phrase and verb phrase.

**ሃውካ**/Hawka **መጺኡ**/me^Si^u (your brother) (came (has come)) **ሃውካ**/Hawka

**ካብሽረመጺኡ**/tmali kab Shire me^Si^u(came (has come) from shire yesterday)
**ንስካ**/nsKa (You – masculine) **እቲ መፅሃፍ**/^ti me^TSHaf (The book) **አብ ልዕሊ አራት ግበሮ** /^ab l`li `arat gbero (Put on the bed).

When people are writing, they make mistakes, mostly the syntactic rules of a language, such as feature agreement, order, and choice of constituents in a phrase or sentence, thus concerning a wider context than a single word. Those grammatically error writing are also happen in Tigrigna language users. Grammar error may occur due to the subject's insufficient knowledge of the language rules or the written language norms that deviate from the already acquired (spoken) grammatical knowledge have to be learned and even can make grammar errors when writing on a computer due to rewriting or rearranging text [32].

Frequent agreement errors in Tigrigna are located within the words in a text can disagree subject and verb, verb and object, adjective and noun, adverb and verb, noun and modifier, word sequence and others in gender, number and persons. Other agreement errors appear between the subject and the predicative, which can involve long distance dependencies.

**Subject-Verb Agreement Errors:** Is the most common errors with subject-verb agreement are to do with number, person and gender in Tigrigna language. For instance,

**ኢደይ ክሕፀብ ኢና።**/ edey kheTsb ena/ I will wash our hand. In this example the subject ኣነ /ane/I which is taken from the object **ኢደይ** and the verb is ኢና /ena/we . The singular subject ኣነ /ane is disagreeing with the plural verb **ኢና**, this make the grammar rule is incorrect. Hence, correct subject-verb agreement has same number that means singular subject takes singular verb. The correct grammar of the sentences is **ኢደይ ክሕፀብ ኢየ።**/ edey kheTsb eye/I will wash my hand.

As any other language the grammar of Tigrigna happens the words in a sentence can disagree

according to person, gender, cases, tenses and number. Other agreement errors appear between the subject and the verb, which can involve long distance dependencies

**Object -verb agreement error:** The same as the subject verb agreement error the object and verb agreement of a sentence is expected to agree in number, gender, and person.

For example, Hagos gezeu'serihato/ **ሃጎስ ገዝኡ ሰሪሃቶ**/Hagos build her home

Here there is a disagreement in object and verb and also feature like gender. **ሃጎስ**/ms3/male singular 3rd person, it disagrees with the verb **ሰሪሃቶ**/female singular 3 rd it disagrees OmVf rule pattern. The correct grammatical sequence is **ሃጎስ ገዝኡ ሰሪሕዎ**/Hagos build his home.

**Noun-modifier agreement error:** Tigrigna nouns are modified by adjectives, determiners,

quantifiers, and others. Adjectives in Tigrigna inflect according to the gender, person and

number of the head noun. For instance, in the **ሃፃር ወዲ** /hatsar wodi/ Short boy, text the adjective ሃፃር /hatsr (short) is third person singular feminine; and The noun **ወዲ** / wodi (Boy) is third person singular masculine. Both the Adjective and the noun agree in number and person. Whereas, disagree in gender maker. In order to correct the grammatical of this text either the adjective could be changed to male or the noun must be changed to fame gender. Therefore, the correct grammar is; the **ሃፃር ንልእ**/ hatsar

Gual or ሃፂር ወዲ/ hatsar wodi/ The quantifier in noun phrase disagrees with number agreement.

**Adverb and Verb Agreement:** Tigrigna Adverb usually modifies the first verb that comes next to it in time, place, circumstance etc. The time adverbs describe the time at which an event takes place. These adverbs may show a specific time at which a given action takes place or its duration. Tigrigna verbs indicate the time at which action takes place in relation with the adverbs.

Tigrigna has generally a subject-object-verb (SOV) word order unlike the English language which follows the subject-verb-object (SVO) sequence in the sentence structure [32].

## 2.3 Word Sequence Prediction Systems

Word prediction can be a task which is challenging and ambitious in a research areas, basically with methods coming from Artificial Intelligence, Natural Language Processing and Machine Learning [16]. What is word sequence prediction? We can easily define the term word sequence prediction once we capture the essence of its essential components which are “sequence” and “prediction”. A sequence is a finite or infinite list of terms (or numbers or things) arranged in a definite order, that is, there is a rule by which each term after the first may be found [61]. Prediction is concerned with guessing the short-term evolution of certain phenomena [16]. Forecasting tomorrow’s temperature at a given location or guessing which asset will achieve the best performance over the next month could be examples of prediction problems. One must predict the next element of an unknown sequence given some knowledge about the past elements and possibly other available information. The entities involved in forecasting task are the elements forming the sequence, the criterion used to measure the quality of a forecast, the protocol specifying how the predictor receives feedback about the sequence, and any possible side information provided to the predictor [62]. Therefore, word sequence prediction is forecasting or guessing the next word the user intends to write or to insert based on some previous information [62].

In natural language processing word prediction is used to guess missing letter, word, phrase, or sentence that are likely to follow a given segment of a text. It has different terminologies like text prediction/word prediction and word completion. They have

been used to express similar and related concepts of word prediction. Word Prediction implies both 'Word Completion' and 'Word prediction' to increase the text prediction rate. Word completion deals with suggesting the user a list of words after a letter has been typed, while Word prediction deals with suggesting the user a list of probable words after a word has been typed or selected, based on previous words rather than on the basis of the letter [2]. Although both of the technologies are used to facilitate typing speed, the implementations and effects are quite different. Word completion deals with prediction of which word the user wants to type now. It starts operating as soon as the user has typed the first letter in a word; analyzing the letters, whereas, word prediction deals with predicting a correct word in a sentence. It saves time, keystrokes and also reduces misspelling [46]. Word prediction system is more effective than character prediction system because word prediction system depends on several words of past context whereas character prediction depends on only upon characters in the current word [23].

Word prediction systems are useful to display a lists of most likely letter, words, or phrases for current position of the system being typed by a user [17,18,19]. Those systems support writing and are commonly used in combination with assistive devices such as keyboards, virtual keyboards, touch pads and pointing devices.

In general, the main purpose of word prediction system is to speed up text entry in different kinds of applications by minimizing keystrokes. Improving and enhancing text entry and interaction with computers for disabled users had been investigated for many years and many systems are proposed to facilitate and simplify text input process [24].

The main issues in the development of word prediction systems include prediction methods and user interface issue. Prediction methods include decisions on prediction units (characters, words), information sources and structure (both lexical and statistical), levels of linguistic processing, size and type of corpora and learning methods [25].

## 2.4 Approaches to Word Prediction Systems

There are many approaches developed towards word prediction that are used to model the natural language. Well known approaches of word prediction and text input methods are statistical and linguistic rules. Those approaches have been studied for different languages. These approaches can be classified into three groups as statistical,

knowledge based and heuristic (adaptive) modeling. Most of existing methods employ statistical language models using word n grams and POS tags [26].

Word frequency and word sequence frequency are commonly used methods in word prediction. In the former method the system use uni gram word model with a fixed lexicon that involves sorting the complete lexicon into their frequency order, and offers the few at the top of the list to user as predictions and it predicts the same suggestion for a particular sequence of letters. However prediction will be better if context is taken into account. The early prediction systems use frequency information of each word independently to complete a word in the current position of a sentence being typed by the user without considering previous context. [26].

In the past various studies are conducted to develop systems that consider previous history of words based on first order Marcov model (bigram) or second order Marcov model (trigram) [26].

### 2.4.1 Statistical Modelling

Statistical word prediction is made based on Markov assumption in which only the last n-1 word of the history affects succeeding word and it is named n-gram Markov model. It is based on learning parameters from large corpora; it is widely used in word prediction. This statistical modeling can be described by prediction using frequencies and prediction using word probability tables.

**Prediction using frequencies**

In prediction using frequencies, the statistical modelling is described by building a dictionary containing words and their relative frequency of occurrence is the simplest word prediction method. It provides *n* most frequent words beginning by this string in the same way they are stored in the system. This method may need some correction by a user in order to adjust its concordance when applied to inflected words since context information are not considered. In other words this method uses unigram model with a fixed lexicon and it came up with the same suggestion for similar sequences of letters. To improve accuracy of word prediction result, indication about recency redundancy of each word may be included in the lexicon. In this way, the prediction system is able to offer most recently used words among most probable words. Adaptation of each word to a user's vocabulary is possible by updating frequency and recency of each word used [26, 27]. Most probable words beginning with the same characters are offered when a

user has written the beginning of a word. If the required word is not available among options offered by the system, a user may continue writing the word, else the required word is accepted from the given list and it may automatically adapt to user's lexicon by simply updating frequencies of words used and by assigning an initial frequency for new words added to the system. In order to enhance the results of this approach, recency field is stored in dictionary with each word along with its frequency. Results obtained with recency and frequency based methods are better than the ones based on frequency alone. However, this method requires storage of more information and increases computational complexity [21, 27]. The advantage of this approach is it is simple when it is applied for non-inflected languages.

**Prediction using word probability tables**

In prediction using word probability tables the statistical model uses word probability tables that consider probability of appearance of each word after the one previously composed. This method builds a two dimensional table, where conditional probability of word Wj after word Wi is stored. Therefore, if the system has N words, there are $N^2$ entries in this table, where most of them are zero or nearly zero probabilities. By using this strategy, the system may offer predictions before a user starts writing the initial character of a word and these results may be improved by integrating recency. The advantage of this method is it offers a word before a user starts typing the first character of a word. The main problem with this method is the difficulty of adaptation to user's vocabulary when dimensions of the table are fixed [21, 27].

### 2.4.2 Knowledge Based Modelling

This knowledge based modelling or linguistic rules focuses on the sequence of syntactic categories of a given input. In detail this can be described as follow:

**Syntactic Prediction**

Considering part of speech tags, and phrase structures, syntactic prediction is to ensure that the system tries to suggest grammatically appropriate words to the user. Almost all human discourse languages are defined and structured. If one follows the grammar rules, he can be able to predict with some degree of accuracy at least the type of words what will come next. Primarily, syntactic prediction needs to know the detail grammar structure of the sentences being created in order to make choices about the types of words it can offer [26].

Statistical syntax and rule-based grammar are two general syntactic prediction methods, where statistical syntax uses the sequence of syntactic categories and POS tags for prediction. Therefore a probability would be assigned to each candidate word by estimating the probability of having this word with its tag in the current position and using most probable tags for previous words. In rule-based grammar, syntactic prediction is made using grammatical rules of the language. A parser will parse current sentence according to the grammar of the language to reach its categories [26].

**Syntactic Prediction Using Probability Tables**

Syntactic prediction using probability table takes syntactic information intrinsic to natural languages into account. This approach makes use of probability of appearance of each word and relative probability of appearance of every syntactic category after each syntactic category.

These systems offer words with most probable syntactic categories at the current position of sentence and results are usually better than the ones obtained using purely frequency based word prediction methods. Probability of appearance of the categories after each category is stored in two dimensional table stores. This table is much smaller than the one presented in frequency based approach and the number of probabilities which are nearly zero is also lower. The probabilities of table and frequencies in lexicon can be updated for adaptation of these systems [21, 26].

**Syntactic Prediction Using Grammars**

In this approach sentences are analysed using grammars either top-down or bottom-up and natural language processing techniques are applied in order to obtain the categories which have the highest probability of appearance. Each natural language has a set of syntactic rules which usually have right to left structure. The sequence that occurs in the right category helps to decompose categories in the left part of the rule. All categories are defined in the system if at least one category has to happen on the right side of the arrow.

Among categories on the right side of a rule, it is possible to define a number of morphological agreement constraints. So that, proposals offered by the predictor are in the appropriate morphological characteristics. The dictionary requires inclusion of morphological information in order to enforce morphological agreement. These systems have a higher computational complexity than the previous ones, mainly due to the fact

that they take the entire beginning of a sentence into account (while previous systems take, at most, last entirely composed words). Word probabilities and weights of syntactic rules can be updated to adapt these types of systems [21, 26, 27].

**Semantic Prediction**

Semantic prediction is to semantically analyse sentences as they are being composed, where each word has an associated semantic category or a set of semantic categories. The working method, complexity, dictionary structure, adaptations, etc. are very similar to syntactic approaches using grammars. It provides comparable results to syntactic approaches though it has much higher complexity, and due to this these methods are not commonly used [21, 27].

In semantic word prediction, Lexical source and Lexical chain are two methods that are used. The first method is a lexical source, like WordNet in English, which measures the probability of words to get certain that predicted words are related in that context. The second method is a lexical chain that assigns highest priority to words which are related semantically in that context by removing unrelated words to that context from the prediction list [21].

Predictions can be correct syntactically or semantically but wrong according to discourse. Pragmatics affects capability of the predictor and taking this knowledge while training the system enhances accuracy of predictions [21].

### 2.4.3 Heuristic Modelling

To make predictions more appropriate for a specific user, the adaptation methods are used. This approach tries to adapt the system to every individual user .There are two general methods that make the system adapted to the users. One of the methods is short-term learning and the other one is long-term learning that will be described in this section.

**Short-term Learning**

In this approach, the system adapts to the user on a current text that is going to be typed by an individual user. Recency promotion, topic guidance, trigger and target, and n-gram cache are the methods that a system could use to adapt itself to a user in a single text. The methods are commonly used in prediction systems.

**Long-term Learning**

In this method, the system gets adapted to the user by considering not only the current text, but previous texts that are produced by the user. As a result, gradually by using the system more, it adapts to the user heuristically [3]. Some of the methods for heuristics adaptations that are language specific are adding new words, automatic capitalization, providing inflected form of words, and compounding.

# CHAPTER THREE: RELATED WORK

## 3.1 Introduction

This chapter provides different word or text prediction researches conducted to Ethiopian languages such as Amharic and Afaan-Oromo, and Non-Ethiopian languages such as Hebrew, Italy, Basque, Swedish, Persian and Urdu. Word prediction has been investigated by many researchers and a number of researches has been proposed and implemented through the years especially for non-inflected and less inflected language. For highly inflected languages some works are done. Here we can present the researcher's work by categorizing as Word Prediction for Ethiopian Languages and Word Prediction for non -Ethiopian Languages because it is difficult to categorize the research work based on approaches they used. In order to choose appropriate approaches for our work, the approaches along with their prediction methods used and the result obtained by the researchers are overviewed and presented.

## 3.2 Word Prediction for Ethiopian Languages

Tigist Tensou [11] developed word prediction for Amharic language using statistical methods and linguistic rules. Constructing Language Model and Generation of Predicted Words are the two major parts. Morphological Analysis of User Input, Word Sequence Prediction, and Morphological Generation are key components of the Generation of Predicted Words part. Here, a large set of collected news texts are morphologically analysed to their component morpheme. Statistical models are constructed for root/stem, and morphological properties of words like aspect, voice, tense, and affixes are modelled using the training corpus. In addition to this, preferred morpho-syntactic features like gender, number, and person are captured from a user's text to ensure grammatical agreements among words. This work is based on Bigram, Tri-gram and Mixed models and it is evaluated using keystroke savings (KSS).Using the test data, 20.5%, 17.4% and 13.1% keystroke savings is obtained in hybrid, tri-gram and bi-gram models respectively. In their experimental situations word sequence prediction using a hybrid model offers better performance than other models for keystroke savings. The researchers tried to consider context information with some morphological features and linguistic rules. However, the researchers' work did not consider the two most important features of WSP which are speed and search space.

Nesredien Suleiman [42] developed word prediction for Amharic online handwriting recognition. The aim of this work is to propose a word prediction model for Amharic online handwriting recognition using statistical information based on frequency of occurrence of words. As the researchers state, this study is motivated by the fact that speed of data entry can be enhanced with integration of online handwriting recognition and word prediction mainly for handheld devices. In this research a corpus of 131,399 Amharic words and 17, 137 names of persons and places are prepared. The prepared corpus is used to extract statistical information like to determine value of n in the n-gram model, the average word length of Amharic language, and the most frequently used Amharic word length. Hence, n is set to be 2 based on statistical information and by hindsight of this, the research is done using bi-gram model, where the intended word is predicted by looking at the first two characters. Finally, a prototype is developed to evaluate performance of the proposed model and 81.39% prediction accuracy is obtained according to the experiment. However, the researchers used the dictionary approach and this study doesn't consider context information while developing the word sequence prediction model.

Ashenafi Bekele [52] developed word sequence prediction model for Afaan Oromo language using bi-gram and tri-gram word statistics, and the bi-gram, and tri-gram POS tag statistics of the language. Initially, the training corpus and user inputs are tokenized and then morphologically analyzed. Subsequently, word statistics model is built for root or stem word and POS tag statistics model is built for root or stem with tag like noun, verb, adjective, pronoun, adverb and conjunction by using training corpus. After that, the most likely probable root or stem words are suggested. Finally, lexical words are synthesized based on the proposed root or stem words. In this research a corpus that consists of 23,400 sentences and a total of 312,208 words are generated in order to filter 49,143 unique words. The designed model is evaluated based on the developed prototype. Depending on the evaluation metrics, the primary word-based statistical system achieved 20.5% KSS, and the second system that used syntactic categories with word-statistics achieved 22.5% KSS. According to the researchers result, statistical and linguistic rules have good potential on word sequence prediction for Afaan Oromo. It doesn't address the problem of speed and search time.

## 3.3 Word Prediction for Non-Ethiopian Languages

Yael Netzer *et al.* [7, 25] developed word prediction for Hebrew languages AAC users. Hebrew Language is morphologically rich, with a high level of ambiguity in which several words combine into a single token in both agglutinative and fusion ways. Nouns, adjectives, verbs, prepositions, and adverbs can be combined with a pronominal pronoun suffix, indicating functions, such as possessive, accusative and nominative (with person/gender/number inflections). These properties are applied to most content-carrying words. The researcher's trained the language model on various training sets of 1M, 10M and 27M words, which were found to achieve good results. However, when they tested prediction performance using morpho-syntactic and syntagmatic information, the performance decreased. The use of morpho-syntactic information such as part of speech tags didn't increase the prediction results contrary to what they expected before the experiment. Furthermore, it decreases the prediction results. The best results were obtained using statistical data on the Hebrew language with rich morphology. The result shows keystroke savings up to 29% with nine word proposals, 34% for seven word proposals and 54% for a single proposal. We believe that an increase in the number of proposals affects search time.

Vitoria and G. Abascal [27] carried out a research on word prediction for inflected language specifically Basque language based on three approaches. This study has briefed various word prediction techniques and their difficulties to apply to an inflected language. Basque language is mainly inflected using suffixes even though there is a possibility of infixes and prefixes. The first approach needs two dictionaries, one for lemmas and other for suffixes since it predicts lemmas and suffixes separately. Where, the first dictionary stores lemmas of the language alphabetically ordered with their frequencies and some morphologic information in order to know which possible declensions are possible for a word. The second dictionary stores suffixes and their frequencies. The system starts prediction by providing a lemma of the next word and when the lemma is accepted, the system offers the suffixes that are correct for this lemma ordered by frequencies. As the acceptable suffixes for a noun can be about 62 only the most probable n suffixes are offered. Possibilities of recursively composed suffixes are some of the challenges in this approach even though hopeful results are obtained. In the second approach syntactic information is added to the dictionary of lemmas and some weighted grammatical rules on the system. The main idea is to parse

a sentence while it is being composed and to propose most appropriate lemmas and suffixes, where parsing allows storing and extracting information that has influenced in forming verb. The third approach treats, beginning of a sentence using statistical information, while advancing in composition of a sentence, and uses this information to offer the most probable word including both lemma and suffix. Three tables are used, one with probabilities of syntactic categories of the lemmas to appear at the start of a sentence, probability of basic suffixes to appear after those words and probabilities of basic suffixes to appear after another basic suffix. Adaptation of the system would be made updating the first table and while suffixes would be added to a word, the other two tables will also be updated. As the researchers state, to predict whole words it is necessary to determine syntactic role of the next word in a sentence, which can be done by means of a syntactic analysis. However the results are not good enough compared with results obtained in non-inflected languages.

Agarwal and Arora [16] developed a Context Based Word Prediction system for SMS messaging in which context is used to predict the most appropriate word. The development of wireless technology has made available different ways of communications like short message service (SMS) and with its tremendous increase of use there comes a need to efficient text input methods. Various scholars came up with frequency based text prediction methods to attempt this problem. However, using only frequency based word prediction may not grant correct result most of the time. For example: considering a sentence *"give me a box of chocolate"* and *"give of a box of chocolate"*, the appropriate word after the word "give" is the word *"me"*, however the system proposes the word *"of"* since it has higher frequency than word *"me",* similarly the appropriate word after the word "*box"* is *"of"* than *"me"* and here frequency based is acceptable. Therefore incorporating context information is helpful to offer suitable word and this work models first order Marcov dependency between POS of consecutive words. A machine learning algorithm is used to predict the most probable word and POS pair, given its code and previous word's POS. Considering the fact that short emails resemble SMS messages closely, the algorithm is trained on 19,000 emails and testing is done on 1,900 emails which are collected from Enron email corpus. The results show 31% improvement compared to the traditional frequency based word estimation.

Aliprandi *et al.* [43, 44] focuses on designing letter and word prediction systems called FastType for Italy language. Italy has a large dictionary of word forms, which go with a number of morphological features, produced from a root or lemma and a set of inflection rules. Statistical and lexical methods with robust open-domain language resources which have been refined to improve keystroke saving are used. The user interface, predictive engine and Linguistic resource, are main components of the system. The predictive engine is the kernel of the predictive module since it manages communication with the user interface keeping trace of prediction status and of the words already typed. The morpho-syntactic agreement and lexicon coverage, efficiently accessing linguistic resources as language models and very large lexical resources are core functionalities of the predictive module. In addition, to improve morphological information available for prediction engine, POS n-grams and Tagged word (TW) n-grams are used. The prediction algorithm for Italian language is presented by an extended combination of POS trigrams and simple word bigrams model. A large corpus prepared from newspapers, magazines, documents, commercial letters and emails are used to train Italian POS n-grams, approximated to n = 2 (bi-grams) and n = 3 (trigrams) and tagged word n-grams, approximated to n = 1 (uni-grams) and n = 2 (bigrams).Keystroke saving (KS), Keystroke until completion (KUC) and Word Type Saving (WTS) are three parameters used to evaluate the system on a test set of 40 texts disjoint from the training set. According to test benchmarks, the result shows a relevant improvement in keystroke saving, which reached 51%, comparable to what was achieved by word prediction methods for non-inflected languages. Moreover, on average 29 % WTS, meaning at standard speed without any cognitive load saving in time and 2.5 KUC is observed.

Hunnicutt .S *et al*. [48] developed a statistical based word prediction system by constructing a database for Swedish language called Profet; it extends the available word prediction system which uses word frequency lexicon, word pair lexicon, and subject lexicon. Profet has been used for a number of years as a writing aid by persons with motoric disabilities and up to 34% in letters when only one word is typed.

Qaiser Abbas [60] conducted a research text prediction of an inflected and under-resourced language Urdu. The interface developed is not limited to a T9 (Text on 9 keys) application used in embedded devices, which can only predict a word after typing initial characters. It is capable of predicting a word like T9 and also a sequence of word

after word in a continuous manner for fast document typing. It is based on the N-gram language model. This stochastic interface deals with three N-gram levels from unary to ternary independently. The unigram mode is being in use for applications like T9, while the bigram and tri-gram modes are being in use for sentence prediction. The measures include a percentage of keystrokes saved, keystrokes until completion and a percentage of time saved during the typing. Two different corpora are merged to build a sufficient amount of data. The test data is divided into a test and a held out data equally for an experimental purpose. According to the researchers view this whole exercise enables the QASKU system outperforms the FastType with almost 15% more saved keystrokes.

## 3.4 Summary

In this chapter we reviewed previous studies both for Ethiopian and non-Ethiopian languages, including the specific approach they applied, and results obtained from their experiment. All the above reviewed papers provided information that is used during development of the word sequence prediction model; this helped us to choose which approach and method of procedure to follow for Tigrigna word sequence prediction. From the reviewed works, we also learnt that considering only the frequency of words is not enough for inflected languages. Incorporating context information increases the effectiveness of prediction output for some languages and to other languages it decreases the result. For instance, Hebrew language. Considering the nature of Tigrigna language and based on literature review we decided to use statistical approach and linguistic rules.

# CHAPTER FOUR: DESIGN OF TIGRIGNA WORD SEQUENCE PREDICTION SYSTEM

## 4.1 Introduction

The aim of this thesis work is to develop a Tigrigna word sequence prediction system that provides or retrieves list of next words to the user. Here, whether the user writes the first word or not the system suggest the list of words. This chapter presents the system architecture, descriptions of each component of its modules with their algorithm.

## 4.2 Architecture of the System

The architecture has three major modules: **Candidate Tag Prediction, Language modelling,** and **Candidate Word Selection.** Here in candidate tag prediction module the user input is tokenized and then the list of token is assigned to its appropriate tag using Tagging component in order to find its appropriate next word based on its tag type and sequence in the tagger dictionary. Then, the Tag sequence prediction component extracts tag sequence and candidate tag list with their tag type and probability by comparing from Tag sequence Model if it is available. After it gets the candidate tag with its tag sequence, the Tag sequence prediction component generates the candidate tag with its tag sequence to Grammar Agreement Checking component. Then, the Grammar Agreement Checking component checks whether the candidate tag is in a grammar rule or not. Subsequently, after checking the grammatical arrangement of candidate tag, the Grammar agreement checking component produces the grammar result and their probability. Language modelling like Word Sequence Model and Tag Sequence Model are built based on the tagged training corpus. The candidate word prediction component uses the Word sequence model to propose the appropriate next word and combine the tag sequence with the word sequence. The candidate word Prediction component produces list of candidate words based on their grammar result to word ranking component. Finally, the word ranking component generates top five lists of words to the word list. The detailed description of each module, components and their algorithms within module is given in the subsequent sections.

The architecture of Tigrigna word sequence prediction model is described in Figure 4.1.

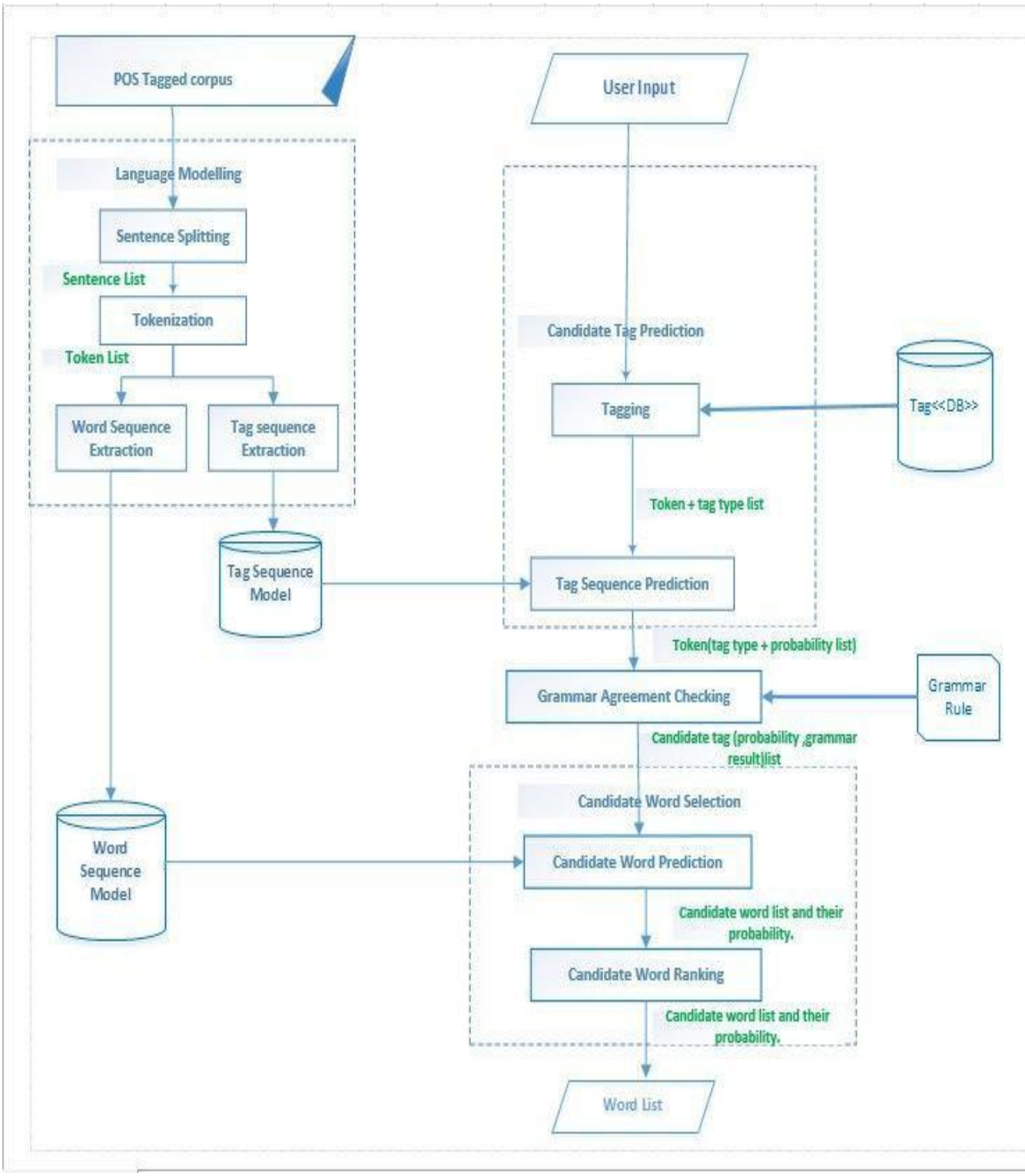


**Figure 4. 1:** Architecture of Tigrigna word sequence prediction model

## 4.3 Language Modelling

This module describes how the POS tagged corpus of Tigrigna word sequence prediction system works in order to produce the prediction model. This corpus received a tagged Text inputs which is treats as a group of sentences. This module has main components namely Sentence Splitting, Tokenization, Tag sequence extraction and word sequence extraction. And finally produces a word and tag language model. This model is used to store statistical information which serves as a knowledge base when predicting suitable tags and words. A sample of Tigrigna sentence annotated with morphology and tag information is shown in Table 4.1.

**Table 4.1:** Sample Tigrigna sentence annotated with morphology and tag

| | |
|---|---|
| <s n="13"><br><w type="N">ምልክታ፡</w><br><w type="ADJ">ኣእምሮኣዊ</w><br><w type="N">ስንክልና</w><br><w type="ADV">ከመይ</w><br><w type="V_REL">ዝበሉ፡</w><br><w type="V_AUX">ኢዮም</w><br><c type="PUN">?</c><br></s> | <s n="14"><w type="N_p">ጸገማት፡</w><w type="N">ስነ-ኣእምሮ</w><br><w type="N_p">ምልክቶም</w><w type="V">ድንጉይ</w><w type="V_IMF">ክብልን</w><br><w type="ADV">ብቐረባ</w><w type="ADJ_p">ዝከታተሉ</w><w type="N">ሰብ</w><br><w type="CON">እንተ</w><w type="V_PRF">ዘይረኽቡ</w><w type="CON">ድማ</w><br><w type="PRE">ክሳብ</w><w type="PRE">ናብ</w><w type="ADJ">ዝለዓለ</w><br><w type="N">ደረጃኡ፡</w><w type="V_REL">ዝበጽሕ</w><br><w type="V_PRF">ከይተለለየ</w><w type="V_IMF">ክጸንሕን</w><br><w type="V_AUX">ይኽእል</w><c type="PUN">። </c></s> |

### 4.3.1 Sentence Splitting

The task of Sentence Splitter component is to split the text input of the corpus to sentences by recognizing the end of a sentence in Tigrigna corpus like full stop(::), question mark (?), and Exclamation (!).

<s n="36">

<w type="ADJ">ጥዕይ</w><w type="N">ኣእምሮ</w><c type="PUN">። </c></s> <s n="37">

<w type="ADJ">ጥዕይ<</w>

<w type="N">ሕብረተሰብ</w>

<c type="PUN">። </c>

</s>

<s n="38">

<w type="ADJ">ጥዕይቲ</w>

<w type="N">ሃገር</w>

<c type="PUN">። </c></s>

### 4.3.2 Tokenization

Tokenization component is responsible to break the sentence into smaller parts called tokens (Words and punctuation).Here the tokenization component split the sentence (which is the output of Sentence Splitting component after splitting the Text) to list of Tokens by identifying space (‘ ’) and single colon (‘’) delimiters. Finally this component returns a list of tokens. The task of tokenization component is shown in Algorithm 4.1.

```
BEGIN

 INPUT list of sentence
For each sentence in SentenceList

Tokenlist[]=Sentence.Split('|', ' '); //tokenize the
    sentences using the white space character and word
and tag concatination '|"

     TagList.add(<start>);
     For i=0; i<TokenList.Count-1; i++

          Wordlist.add(Tokenlist[i],Tokenlist[i+1]);//
          add the word and its tag type from the
          tokenlist

         TagList.add(TokenList[i+1]);// add the tag type
 of the word

          i+=2; // go to the next word token

     end for
     TagList.add(<end>);

 End for each

 OUTPUT list of token
  END
```

**Algorithm 4. 1**: *Tokenization*

### 4.3.3 Tag Sequence Extraction

The Tag sequence extraction is responsible to fill the Tag sequence model Bigram and Trigram tables with the token tag sequence and their probability list by performing probability calculation. This component is responsible to calculate the probability of each unique tag sequence list that comes from the Tokenization component by using the N-gram statistical language model to calculate the probability of the tag sequence. Specifically, N is given as N=2 and N=3 applied to calculate the probability of 2-tag sequence and 3-tag sequence respectively. If the bigram is found, a subset of all 3-grams starting with that 2-gram is generated. The probability of the resulting trigram options is calculated by dividing the count of the trigram by the count of the bigram.

Finally, after it is calculated the sequence probability of tags, the token its tag type, tag sequence, and their probability list is stored on the Tag Sequence Model. The task of the Tag Sequence Extraction component is depicted in Algorithm 4.2.

```
BEGIN
INPUT list of token  Probability=
For each tag in TagList// unique tag extraction
If tag is not in TagModel.Unigram
       Probabiity=count (TaginTagList)/TagList.Count();
       TagModel.Unigram.Add(tag, Probability);
  End If
For i=0; i<TagList.count;i++ // two tag sequence extraction
 Tag1=TagList[i], Tag2=TagList[i+1]; Tag3=TagList[i+2];  sequence
If Tag2 is not <start> and Tag1 is not <end>// extract two tag
   If Tag1,Tag2 is not in TagModel.Bigram
       Probabiity=count(Tag1,Tag2 in  TagList)/TagList.Count();
       Probabiity=count(Tag in TagList)/TagList.Count();
       TagModel.Bigram.Add(Tag1,Tag2,Probability);
  End If
  End If
If Tag2 and Tag3 are not <start> and Tag1 and Tag2 are not
<end>//extract Three tag
sequence
       If Tag1, Tag2, Tag3 are not in TagModel.Trigram
Probability=count(Tag1, Tag2, and Tag3 inTagList)/TagList.count();

 TagModel.Trigram.Add(Tag1,Tag2,Tag3, Probability);
           End IF
 End If
End For each
```

**Algorithm 4. 2:** *Tag Sequence Extraction*

### 4.3.4 Word Sequence Extraction

The Word sequence extraction component is responsible to fill the Word sequence model Bigram and Trigram tables with the token sequence and their probability list by performing probability calculation. This component is responsible to calculate the probability of each unique word sequence list that comes from the Tokenization component by using the N-gram statistical language model in order to calculate the probability of the word sequence. Specifically, N is given as N=2 and N=3 applied to calculate the probability of Bi-gram and Tri-gram sequence respectively. If the bigram is found, a subset of all 3-grams starting with that 2-gram are generated. The probability of the resulting trigram options is calculated by dividing the count of the trigram by the count of the bigram. Finally, after it is calculated the sequence probability of words, the token its word list and their probability list is stored on the Word Sequence Model. The task of Word Sequence Extraction component is depicted in the Algorithm 4.3.

```
BEGIN
INPUT list of token
 Probability =0;
For each word  in wordlist
 If word is not in WordModel
Probability=Count (word in Wordlist)/Wordlist.Count;
    WordModel.Add(wordlist.Word, wordlist.tagtype,
Probability);
    End IF
 OUTPUT Bigram and Trigram table for
END
```

**Algorithm 4.3:** *Word Sequence Extraction*

## 4.4 Candidate Tag Prediction Module

This module is responsible to process the extraction sequence of words in tag, and attaches a part of speech tagger to each word. It also performs the tag sequence prediction of candidate tags from Tag Sequence Model to return a list of tag candidates

based on the probability of tag sequences. This module has the following components Tagging, Tag Sequence Prediction and Grammar Agreement checking.

### 4.4.1 Tagging

The Tagging component is used to tag tokens with their appropriate POS tags from the tagger dictionary called "Tag <<DB>>". This is performed by accepting token list from the tokenization component as input and assigns each token with a tag by the help of tagger dictionary. Finally the tagging component returns token and Tag list. The tag has information of tokens in word class (POS) and its inflection information (morphology) of each word class. The task of Tagging component is depicted in Algorithm 4.4.

```
BEGIN
Read Tag, UserInput, WordModel, TagModel;
Taglist[];
Tokenlist[]=UserInput.split(' ', ':'); //Split text into tokens
If(Tokenlist.Count>0)
Taglist.add(<start>);//sentence start maker
ForeachTi in Tokenlist //Where Ti,i=1,2,3,…,n n is number of
tokens
     If Ti is in Tag
Taglist.Add(Tagtypetag of Ti) If Tagtypetag of Ti is sentence
end punctuation
     Taglist.Add(<start>) End if
End if
Else Taglist.Add("Unknown")
End foreach
End if Else
Taglist.Empty(); OUTPUT Token and tag list
END
```

**Algorithm 4.4 Token Tagging**

### 4.4.2 Tag Sequence Prediction

The Word sequence extraction component is responsible to fill the Word sequence model Bigram and Trigram tables with the token sequence and their probability list by performing probability calculation. This component is responsible to calculate the probability of each unique word sequence list that comes from the Tokenization component by using the N-gram statistical language model in order to calculate the probability of the word sequence. Specifically, N is given as N=2 and N=3 applied to calculate the probability of Bi-gram and Tri-gram sequence respectively. If the bigram is found, a subset of all 3-grams starting with that 2-gram is generated. The probability of the resulting trigram options is calculated by dividing the count of the trigram by the count of the bigram. Finally, after it is calculated the sequence probability of words, the token its word list and their probability list is stored on the Word Sequence Model. The task of the Word Sequence Extraction component is depicted in Algorithm 4.5.

```
BEGIN
INPUT candidate tag list
Candidatetag[][]=Empty;// initialize candidate as none
If N>1 // Predict next tag from TagModel.Trigram
FirstTag=Taglist[N-2],SecondTag=Taglist[N-1];
    For i=0,i<TagModel.Trigram.count;i++
          If FirstTag is TagModel.Trigram[i].Tag1 and secondTag is in
TagModel.Trigram[i].Tag2 //
Candidatetag.Add(TagModel.Trigram[i].Tag3,TagModel.Trigram[i].Probability
     End if
   End for
Else if N=1 //Predict Candidate tag from TagModel.Bigram
     Firsttag= Taglist[0];
     For i=0,i<TagModel.Bigram.count;i++
          Tag1=If FirstTag is TagModel.Bigram[i].Tag1
     Candidatetag.Add(TagModel.Bigram[i].Tag2,TagModel.Bigram[i].Probabilit
y
          End if
     End for
End Else If
Else// if there is no user input Predict top ten candidate tag
Candidatetag.Add(TopTen(TagModel.Unigram.Tag1),TopTen(TagModel.Unigram.Prob
ability));
          LM.Add(candidate,probability);
End else
OUTPUT token tag type probability list
END
```

**Algorithm 4.5:** *Tag Sequence Prediction*

### 4.4.3 Grammar Agreement Checking

The Grammar Agreement Checking component is responsible to get candidate tag, if the candidate tag is available by checking the existence in the list whether it exist or not using grammar rule. Here the system compares the candidate tag which retrieves from Tag sequence prediction component with the manually prepared rules like SUBJECT-VERB, OBJECT-VERB, MODIFIER-NOUN, and ADVERB-VERB grammar agreement rules with the appropriate Morphological feature. So that if the candidate is available then the rule filters and deletes the candidate from candidate list. The task of Grammar Agreement Checking component is depicted in the Algorithm 4.6.

```
BEGIN
INPUT token, tag type and probability
 If candidate is in GrammarRule// grammatically incorrect candidate
Candidatetag.remove(candidate)// delete the candidate
End if
  End foreach
Return Candidatetag;
OUTPUTcandidate tag with grammar result END
```

**Algorithm 4.6:** ***Grammar Agreement Checking***

## 4.5 Candidate Word Selection Module

The task of this module is to present the list of candidate next words to the user. To perform this task it uses Candidate Word Prediction and Candidate Word Ranking components**.**

### 4.5.1 Candidate Word Prediction

The main task of Candidate word prediction component is to retrieve the list of candidate words using the Word Sequence Model, by interacting with candidate tags and their grammar result list that comes from Grammar Agreement Checking component.

```
UT Read Candidatetag, WordModel;
CandidateWord[][].empty;
TopCandidateWord[];
If Candidatetag.Count>0// Predict top words based on the
Candidatetag
   For each tag in Candidatetag
     For i=0;i<WordModel.count;i++
          If tag is in WordModel[i].tagtype

     CandidateWord.add(WordModel[i].word,WordModel[i].Proba
bility);
          End If
      End for
End For each
```

**Algorithm 4.7***: Candidate Word Prediction*

### 4.5.2 Candidate Word Ranker

The function of Candidate word ranker component is to extract the candidate tag and generate its appropriate list of next word based on the result that comes from candidate word prediction component. Word sequence model is a language model used for retrieving the candidate word from the Word Sequence Extraction by comparing the probability of candidate word prediction with the unigram table. The unigram table is used simply to present the candidate word lists and their probability. The candidate word ranking component selects the most top 5 appropriate next words based on highest

probability result and lists them on the *word list* by prioritizing the lists and provide to users in order to choose their intended next word. The task of candidate word ranker is shown Algorithm 4.8.

```
 BEGIN
INPUT candidate word
For i=0;i<5;i++
ProbIndex= i
For J=i+1; J<CandidateWord.count,J++
   If
CandidateWord[ProbIndex].Probability<CandidateWord[J].Pr
obability
              ProbIndex=J;
End if
      End for
        TopCandidateWord.add(CandidateWord[ProbIndex].word);
    End For
End If
Else //there is no candidate tag
  TopCandidateWord.add("Select top 5 word from WordModel");
End Else //
    Display( TopCandidateWord)
 OUTPUT Top candidate word
END
```

**Algorithm 4.8:** *Candidate Word Ranking*

# CHAPTER FIVE: EXPERIMENT

## 5.1 Introduction

This chapter focuses on the corpus collection and experiment of Tigrigna word sequence prediction prototype. The corpus preparation part includes the Text Corpus preparation, Tagged Text Preparation, Grammar Rule Preparation, Dictionary Table Preparation and Sample Tigrigna next word prediction input output. In the Experiment part we will discuss the experiment parameters and the evaluation result.

## 5.2 Corpus Collection

To implement the prototype of Tigrigna next word prediction we use a corpus of 3,000 sentences, a tagger consists of 10,000 unique words, a grammar rule consists of 114 grammar agreement rule patterns, and tools of C# programming language, and SQL Database. The detail of the prototype implementation and tools used is presented in this section.

**Notepad++:** is a free tool that is used in this thesis to prepare the annotated Training corpus and agreement grammar rules that are used for grammar agreement checking component in an XML structure.

**Microsoft Visual Studio:** is one of the windows application development platforms, which runs under windows operating system, contains the windows SDK manager, class libraries and SQL database. We use this SDK to create a Graphical User Interface for input of texts from a user and to display list of predicted words to the user.

**C# programming language:** We write a C# code that helps to train the word prediction algorithm and to acquire experimental results from the database.

**SharpNlp:** is an open source toolkit that contains open source C# modules such as sentence splitter, tokenizer, POS, etc.

SharpNLP is a collection of natural language processing tools written in C#.

- it is an open source toolkit that contains open source C# modules, currently it provides the following NLP tools: a sentence splitter, a tokenizer, a part-of-speech tagger, a chunker (used to "find non-recursive syntactic annotations such as noun phrase chunks"),a parser, a name finder and an interface to the WordNet lexical database

- Developed for research development in NLP fields.

**SQL database tool:** is used to store the tokens within tag type in the tagger dictionary database, and the trained data in the unique tag sequence and probability database that used to determine agreement error.

### 5.2.1 Text Corpus Preparation

The Corpus contains 3,000 sentences collected from three sources of areas such as magazine [47], books [62, 63], and websites [34] of each 1,000 sentences manually. From thus; we use 2,800 sentences for training the system and a unique of 1,200 words from the rest 200 sentence for the purpose of testing the system. The grammatical correctness of the corpus is approved by Tigrigna language professionals in order to test the performance of our system. Appendix A shows the corpus that we are used in our system.

### 5.2.2 Tagged Text Preparation

The tagger assigns linguistic meanings to each word. Each affix (prefix, infix, and suffix) or morpheme in word refers to different linguistic meaning such as number, person, gender, definiteness and the like. The tagger uses manually prepared dictionary.

The dictionary contains 10,000 unique words extracted from the [28]. It contains seven different category of word with their labels. Noun, Pronoun, adjective, Number, adverb and verb contain seven labels individually and we add linguistic meaning of the words manually. Finally the dictionary has 32 unique labels, which are used to assign tag type of input word and the corpus.

### 5.2.3 Grammar Rule Preparation

Grammar is used to find grammar errors from the statistically predicted words. Thus, we use Tigrigna agreement grammar rules in XML structure [32]. The system uses the grammar rule file in the xml file to check errors in a sentence, if a pattern declared in the rule matches the input sentence, then error is shown to the user. The grammar rule has 114 unique incorrect grammar agreement patterns. Out of the 114 patterns, around 43 patterns are MNA, 27 patterns are SVA, 27 patterns are OVA, and the rest 17 patterns are AVA. The grammar rules are shown in Figure 5.1.

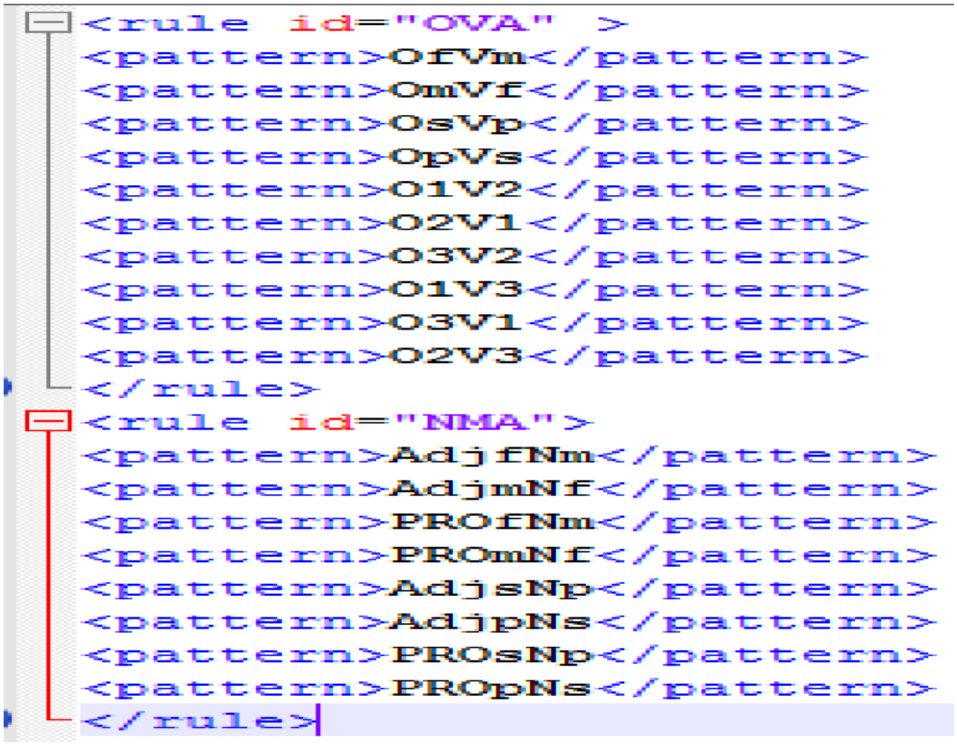


**Figure 5. 1:** Sample of Prepared grammar rules

### 5.2.4 Dictionary table Preparation

**SQL Database Model**

The database model is a storage that holds the tagger dictionary and the language model data's such as unigram, bigram, and Trigram.

- **Unigram Table:** is prepared in a three columns; the first column is the unique word, the second column is their tag type and the third their frequency in the corpus. It is stored in SQL database table as shown in Figure 5.2.

| Word | Tag | Frequency |
|---|---|---|
| ሐበርኩዋ | V_PRF_Ssm1_Osf3 | 0.04089219 |
| ሐበሮ | V_PRF_Ssm3_Osm3 | 0.007434944 |
| ሃበቕናይ | N_s | 0.007434944 |
| ሃበቶ | V_PRF_Ssf3_Osm3 | 0.007434944 |
| ሐበን | N_s | 0.01115242 |
| ሃቡ | V_PRF_Spm3 | 0.007434944 |
| ሃቡቕናይ | N_s | 0.01115242 |
| ሃቡኔ | V_IMV_Spm3_Osm1 | 0.04089219 |
| ሃቡና | V_IMV_Spm3_Op1 | 0.01115242 |
| ሐቢረን | V_GER_Spf3 | 0.01115242 |
| ሐቢረየም | V_GER_Spm3 | 0.01115242 |
| ሐቢና | V GER Ssm3 | 0.01115242 |

**Figure 5.2**: Sample data stores in the unigram Table

- **Bigram Table:** is prepared in a three columns; the first column is the first tag, the second column is their next tag and the third their frequency in a sequence in the corpus. It is stored in SQL database table as shown in Figure 5.3.

| First_Tag | Next_Tag | Frequency |
|---|---|---|
| ADJ | N | 0.01486989 |
| N | PUN_End | 0.01115242 |
| PRE | N | 0.007434944 |
| N | N_p | 0.01115242 |
| N_p | PUN_dcot | 0.007434944 |
| PUN_dcot | ADJ | 0.01115242 |
| ADJ | N_s | 0.04089219 |
| N_s | N_PRP_s | 0.01115242 |

**Figure 5.3**: Sample data Stores in the Bigram Table

**Trigram Table:** it contains the sequences of three tags and their frequency in the corpus. The trigram table is shown in the Figure 5.4.

| First_Tag | Second_Tag | Next_Tag | Frequency |
|---|---|---|---|
| start | ADJ | N_s | 0.007874016 |
| ADJ | N_s | ADJ_sm | 0.007874016 |
| N_s | ADJ_sm | N_s | 0.007874016 |
| ADJ_sm | N_s | ADJ | 0.007874016 |
| N_s | ADJ | CON | 0.007874016 |
| ADJ | CON | V_IMF | 0.007874016 |
| CON | V_IMF | V_AUX_Ssm3 | 0.007874016 |
| V_IMF | V_AUX_Ssm3 | PUN_End | 0.007874016 |

**Figure 5.4:** Sample data store in the Trigram Table

- **Dictionary Table:** It contains a unique of 10,000 words and their linguistic meaning. The dictionary table is shown in the Figure 5.5.

| Word | Tag_Type |
|---|---|
| ህንጡይነት | N_s |
| ህንጸት | N_s |
| ህንጻ | N_s |
| ህንጻታት | N_p |
| ህንጻታትን | N_PRP_p |
| ህንጻዊ | ADJ_sm |

**Figure 5.5:** Sample words in the Dictionary

## 5.3 Implementation

We used different tools and developing environments in order to implement the algorithms and to do necessary experiment on the system. We write a C# code that helps to train the word prediction, to validate the grammatical correctness of candidate words, to display top five candidate words, and to acquire experimental results from the database.

The User type a text in text box (input) and when space bar or one of delimiters is pressed, the system predicts 5 possible single next words and shows them in list box, with the most likely suggestion in the top of the list. Next, a user clicks his or her preferred word from a given list of word options instead of typing each character and

then click add button. However, if the required word is not listed in a given option, then a user continues typing as usual writing method manually. Figure 5.6 and Figure 5.7 shows User interface of word sequence prediction.

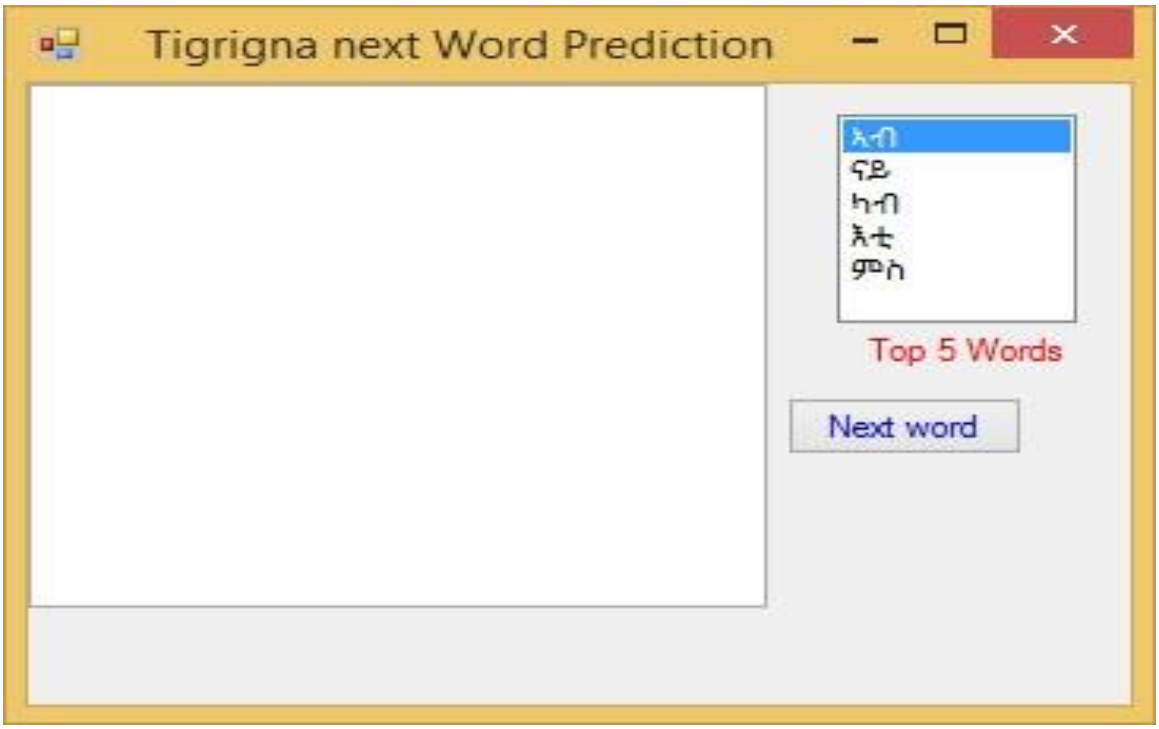


**Figure 5.6**: User interface without user input

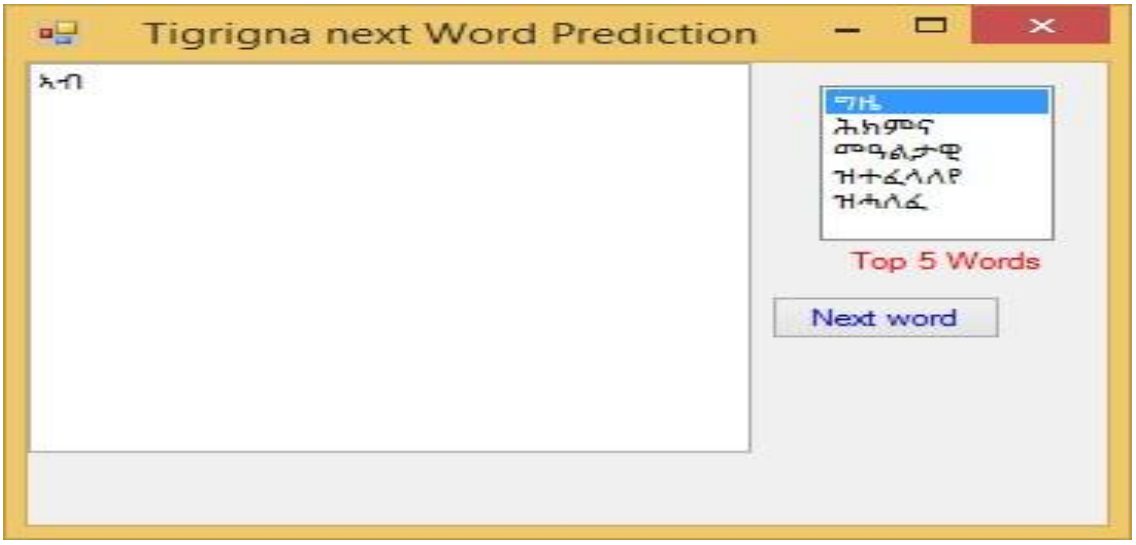


**Figure 5.7**: User interface of word sequence prediction

## 5.4 Test Results

We evaluate Tigrigna word sequence prediction prototype, using the evaluation metric precision manually. In order to evaluate our system performance, we use an evaluation metrics precision manually. Given that FCW be the number of words correctly flagged and FWW be the number of words wrongly flagged, NFW be the number on flagged words, then the precision is given by the following equations.

$$\text{Precision} = \frac{CFW}{CFW+FWW} \quad \text{.....eq(1).}$$

Where, CFW indicates correctly flagged words and WFW are flagged wrong words.

The testing is performed a sample of 1,000 words were taken from the test data set by categorized as sequence of two words and sequence of three words as shown in the Table 5.1.

**Table 5.1**: Test data

| Testing category | Number of wo correctly flagged | Number of words wrongly flagged | Total |
|---|---|---|---|
| Sequence of two words | 428 | 72 | 500 |
| Sequence of three words | 405 | 95 | 500 |
| Total | 833 | 167 | 1000 |

We perform system testing in four different days by labelling as Exp1, Exp2, Exp3 and Exp4 of results in Day1, Day2, Day3, and Day4 respectively. In each experiment 250(Two hundred fifty) sequence of word were distributed and tested daily and accordingly, the result of correctly predicted sequence of words are shown in Table5.2 and Figure 5.3.

**Table 5.2**: Correctly predicted words in each experiment

| | Experiment Results | | | | |
|---|---|---|---|---|---|
| | Exp1 | Exp2 | Exp3 | Exp4 | Average |
| Sequence of two tags | 79% | 83% | 92% | 86% | 85% |
| Sequence of three tags | 81% | 85% | 76% | 84% | 81.5% |
| Average test result | 80% | 84% | 84% | 85% | 83.25% |

The average experiment result for correctly predicted word is shown in Figure 5.3

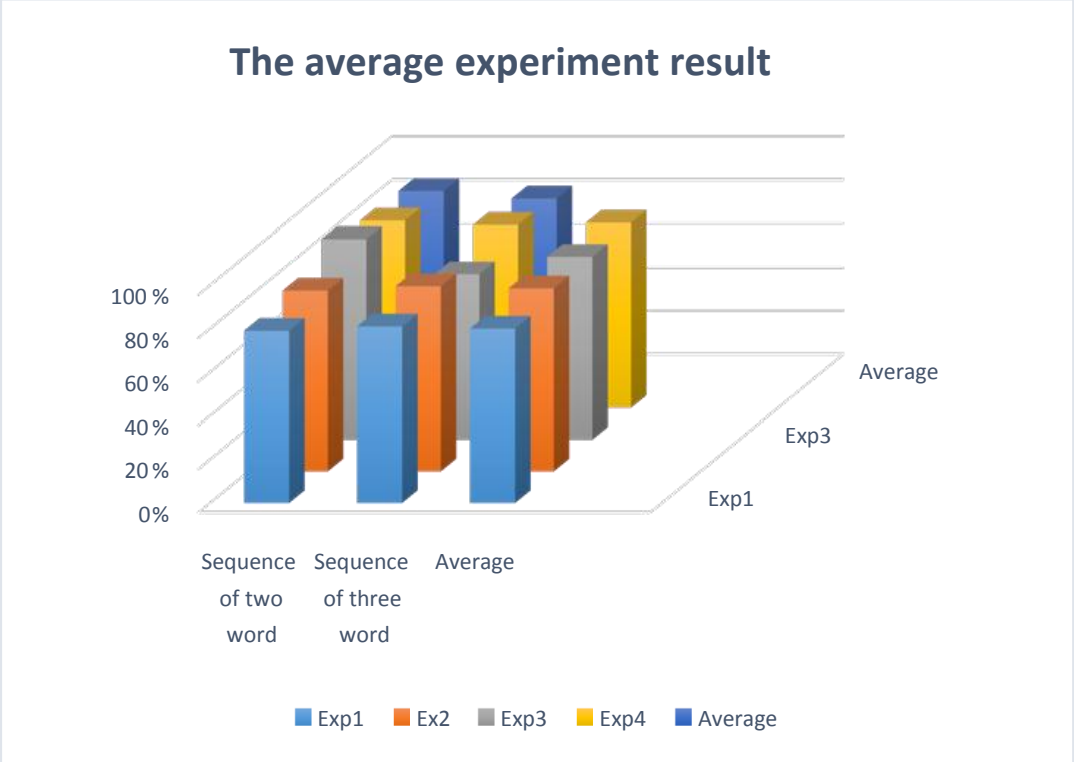


**Figure 5.7**: The average experiment result

## 5.5 Discussion

The result of the experiment is shown on table 5.2. When Testing is done using Sequence of two tags in different days by labelling as Exp1, Exp2, Exp3 and Exp4 we obtain the performance of correctly predicted words 79%, 83%, 92% and 86% respectively. When Testing is done using Sequence of Three tags in different days by labelling as Exp1, Exp2, Exp3 and Exp4 we obtain the performance correctly predicted words 81%, 85%, 76% and 86% respectively. On the average, 85 % performance of correctly predicted words are obtained using Sequence of two tags and 81.5 % performance of correctly predicted words are obtained using Sequence of Three tags. Word prediction using Sequence of two tags provides better performance than Sequence of Three tags.

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

In this study, a word sequence prediction model is developed for Tigrigna language using statistical methods and linguistic rules. Word prediction software is used to minimize keystrokes for different users, especially people with disability by forecasting the next intended word in order to improve their writing methods. We prefer considering Part Of Speech based prediction techniques that could help the predictor to suggest the next word. As we have stated in Chapter 3 a number of researches have been conducted on various languages. Even if there are different researches in Tigrigna, there is no work on the topic of word sequence prediction that considers both syntax and word information.

This study is used to develop a Tigrigna word sequence prediction system that provides or retrieves a list of next words to the user depending on previous history words. This is done using n-gram statistical models based on two Markov language models, one for tag, the other for words which are developed using manually tagged corpus, and grammatical rules of the language.

The designed model is evaluated based on a precision metric used to evaluate systems performance. According to our evaluation, On the average 85 % performance of correctly predicted words are obtained using Sequence of two tags and 81.5 % performance of correctly predicted words are obtained using Sequence of Three tags. Word prediction using Sequence of two tags provides better performance than Sequence of Three tags.

## 6.2 Contribution of the Thesis

The contributions of this thesis work are listed as follows:

- We proposed architecture for Tigrigna word sequence prediction model
- We identified that POS based n-gram provides better prediction when there is a good tagger.
- We developed algorithms for the Tigrigna word sequence prediction model.

## 6.3 Future work

This work can be extended in numerous ways to enhance the task of Tigrigna word sequence prediction. The following are some of the recommended views for future work.

1. Lack of adequate POS tagged corpus makes this works hard to keep morph-syntactic agreement complete. However, Tigrigna word sequence prediction can be improved if good Tigrigna POS tagger is incorporated and if the model is supplemented with POS. 2. Tigrigna is a morphologically complex language as we have discussed in Chapter 2. Moreover, there is no morphological analyser tool for Tigrigna language. In this work, we used manually prepared morphological information embedded with a POS tagger to keep morpho-syntactic information. Hence, we recommend the development of morphological analyser and synthesizer tools for the language.
2. To accomplish the task of word sequence prediction process for Tigrigna language needs an efficient training data set with quality and quantity. We use a corpus of 3,000 sentences; a tagger consists of 10,000 unique words. We suggest that better prediction results can be obtained when training on large amounts of data.
3. The quality of POS taggers has quite a negative effect on the predictor system. In this work, when predicting next words based on their tag sequence for a given word, the performance result is somehow decreased when the number of n tag sequences is increased. Because when the number of n-tag sequences increases it becomes complex. Therefore, we recommend considering recency of words other syntax and semantic methods along with highest frequency.
4. This research is done to predict the next word after analysing sequences of words in a given simple Tigrigna sentence by checking the grammatical arrangement which is “single word” than phrases or other sentences in a standalone platform. Thus, to develop a word sequence prediction and to predict in a phrase level in a sentence is an open research area.

# References


[ 1] Koray Ak, Olcay Taner and Yıldız, "*Unsupervised Morphological Analysis Using Tries"*, Dept. of Computer Science and Engineering, Isık University, 2011.

[ 2] Nestor Garay-Vitoria and Julio Abascal, *"Text Prediction systems: a survey"*, DOI10.1007/s10209-005-0005-9, 2005

[ 3] Nicola Carmignani, ***"****Predicting words and sentences using Statistical Models"*, Language and Intelligence reading group, 2006

[ 4] Steven Bird, Ewan Klein, and Edward Loper, 2009. Natural Language Processing with Python, O'Reilly Media ,1st ed. USA

[ 5]Shamala Gallagher, Anna Rafferty, Amy Wu "Word Prediction" retrieved fromhttp://www.cs.stanford.edu/people/eroberts/courses/soco/projects/200405/nlp/tec hniques word.html; last visited Nov 17, 2015.

[ 6]Nestor Garay-Vitoria, Julio G. Abascal, "*Word prediction to inflected language, Application to Basque Language"*,1997.

[ 7]Yael Netzer. Meni Adler. Michael Elhadad***"****Word Prediction in Hebrew-Preliminary and Surprising results",*August 6, 2008

[8] C.Aliprandi, N.Carmignani, N.Deha, P.Mancarella, M.Rubino, *"Advances in NLP applied to Word Prediction"*, University of Pisa, Italy,2007

[9] Masood Ghayoomi and Ehsan Daroodi, *"A POS based Word Prediction System for the Persian Language"* Nancy 2 University, Nancy, France and Iran National Science Foundation, Tehran, Iran 2008.

[10] C.Aliprandi, N.Carmignani, P.Mancarella, *"An Inflected-Sensitive Letter and WordPrediction System"*, International Journal of Computing and Information Sciences,Vol.5,No.2 ,University of Pisa ,Italy, August 2007

[11] Tigist Tensou Tessema, "*A Word Sequence Prediction for Amharic Language"*, Unpublished Master's Thesis, Department of Computer Science, Addis Ababa University, 2014.

[12] Copestake, Augmentative and Alternative NLP Techniques for Augmentative and Alternative Communication, Proceedings *of the ACL Workshop on NLP for Communication Aids*, 37–42, 1997.

[13] C. Aliprandi, N. Carmignani, P. Mancarella and M. Rubino,A Word Predictor for Inflected Languages: SystemDesign and User-Centric Inferface, *Proceedings of the 2nd IASTED International Conference on Human-Computer Interaction*, 2007.

[14] Nicola Carmignani, "Predicting Words and Sentences using Statistical Models" Language and Intelligence Reading Group, Department of Computer Science, University of Pisa , July 5, 2006.

[15]Tigrignalanguage,retrievedfromhttp://www.ucl.ac.uk/atlas/Tigrinya/langssuage.html, last visited Nov 21, 2015.

[16] Sachin Agarwal, Shilpa Arora,"*Context Based Word Prediction for Texting Language*", Conference RIAO2007, Pittsburgh PA, U.S.A., 2007

[17] Masood Ghayoomi, SaeedehMomtazi,"An Overview on the Existing Language Models for Prediction Systems as Writing Assistant Tools", Saarland University, Germany,978-1-4244-2794-9/09, IEEE,2009

[18] Peter F.Brown, Peter V.deSouza, Robert L. Mercer,Vincent J. Della Pietra, Jenifer C. Lai,"Class based n-gram Models of Natural Language", IBM T.J. Watson Research Center,Yorktown Heights, New York 10598, Association of Computational Linguistics, Vol. 18,No.4, 1999

[19] Fredrik Lindh, "Japanese Word Prediction", Lund University, Sweden, 2011

[20] Nicola Carmignani, "Predicting Words and Sentences using Statistical Models", Department of Computer Science, University of Pisa, Language and Intelligence Reading Group July 5, 2006

[21] Nestor Garay-Vitoriaa and Julio Abascal, "Text prediction systems: a survey", 8 December 2005

[22] A. Fazly and G. Hirst, "Testing the efficacy of part-of-speech information in word completion," *Eacl*, no. 1991. pp. 9–16, 2003.

[23] Johannes Matiasekl and Marco Baroni1, and Harald Trost21 "FASTY - A multilingual approach to text prediction ",Austrian Research Institute for Artificial Intelligence, Austria, Department of Medical Cybernetics and Artificial Intelligence, University of Vienna

[24] Hisham Al-Mubaid and Ping Chen “Application of word prediction and disambiguation to improve text entry for people with physical disabilities (assistive technology)”, International Journals of Social and Humanistic Computing,1(1)1027,2008

[25] Yael Netzer. Meni Adler and Michael Elhadad “Word Prediction in HebrewPreliminary and Surprising results”, August 6th ISAAC 2008

[26] Masood Ghayoomi and Saeedeh Momtazi,”An Overview on the Existing Language Modelsfor Prediction Systems as Writing Assistance Tool” Saarland University, Saarbruecken, Germany, IEEE, 2009

[27] Nestor Garay-Vitoria and Julio G. Abascal, “Word prediction to inflected language, Application to Basque Language”

[28] Daniel Teklu,”ዘበናዊ ሰዋሰው ቋንቋ ትግርኛ”, መቐለ: Mega printing enterprise, 2000 ዓ/ም.

[29] YonasFissha,”Development of Stemming Algorithm for Tigrigna Text”, Master’s thesis, Addis Ababa University, Department of Information Science, 2011.

[30] Michael Gasser,” Semitic Morphological Analysis and Generation Using Finite State Transducers with Feature Structures” Indiana University, School of Informatics, Bloomington, Indiana, USA, gasser@indiana.edu

[31] Hailay Beyene Berhe, “Design and development of Tigrigna search Engine”, Addis Ababa University, March, 2013

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

[33] Edited by John Mason “Tigrigna Sewasew” American Evangelical Missoin ,Edition 1996

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

[35] Teklay gebregzabiher abreha, “Part of speech tagger for Tigrigna language”, Addis Ababa University, November, 2010

[36] Michael Gasser. HornMorpho 2.5 user'sguide. Indiana University, Indiana, 2012.

[37] BayeYimam,"Yamargnasewasiw"("Amharic Grammar"),Addis Ababa, Ethiopia, 2000

[38] NegaAlemayehu and Peter Willett,"Stemming of Amharic words for information retrieval",University of Sheffield, UK, Litrary and Linguistic computing vol.17,No1,2002

[39] **ዳኒኤልተኽሉረዳ**, (1996 **ዓ.ም** ). **ዘመናዊ ስዋሰው ቋንቋ ትግርኛ፣ኣ.ኣ**

[40] Michael Gassar,"HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrigna",Indiana University, USA,2009

[41] EinatMinkov, KristinaToutanova, Hisami Suzuki,"Generating complex Morphology for Machine Translation," Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 128–135, Prague, Czech Republic, June 2007

[42] Nesredin Suleman, *"Word Prediction for Amharic Online Handwriting recognition"* Addis Ababa University, Msc. Thesis, 2008

[43] C.Aliprandi, N.Carmignani, N.Deha, P.Mancarella, M.Rubino, *"Advances in NLP applied to Word Prediction"*, University of Pisa, Italy, 2007

[44] C.Aliprandi,N.Carmignani, P.Mancarella, *"An Inflected-Sensitive Letter and Word Prediction System"*, International Journal of Computing and Information Sciences, Vol.5, No.2, August 2007, University of Pisa,Italy))

[45] Johannes Matiasek, Marco Baroni. Harold Trost, "*FASTY- A multi-lingual approach totext prediction"*, Proceedings, Vol. 2398, Springer, Berlin-Heidelberg-New York, 8th International conference, ICCHP 2002, Linz, Austria

[46] Keith Trnka, "*Adaptive language modeling for word prediction"*,Keith Trnka, University of Delaware, Newark, DE 19716, Proceedings of the ACL-08: HLT

Student Research Workshop (Companion Volume), pages 61–66, Columbus, June 2008, Association for Computational Linguistic

[47] Eyas El-Qawasmeh ,*"Word Prediction via a Clustered Optimal Binary Search Tree"*, The International Arab Journal of Information Technology, Vol. 1, No. 1, January 2004

[48] A. and Hunnicutt, S. and J. and Stromstedt, G. and Wachtmeister, H., "*Constructing a database for new word prediction system"*, TMH-QPSR Vol. 37 No.2, 1996

[49] Sheri Hunnicutt, Lela Nozadze, George Chikoidze, *"Russian Word Prediction with Morphological Support"*, KTH University, Sweden

[50] Javed Ahmed Mahar, Ghulam Qadir Memon"Probablistic Analyis of Sindhi Word Prediction using N-Grams", Australian Journal of Basic and Applies Sciences, 2011

[51] Hisham Al-Mubaid,"A Learning classification based approach for word prediction", vol. 4, No.3, July 2007

[52] Ashenafi Bekele Delbeto, "*Word Sequence Prediction for Afaan Oromo"*, Unpublished Master's Thesis, Department of Computer Science, Addis Ababa University, 2018.

[53]Klund,J.and Novak,M.(2001).If word prediction can help, which program do you choose? Availableat :http: //trace .wisc.edu/docs /word prediction 2001/index.htm?

[54] J. Hasselgren, E. Montnemery, P. Nugues, and M. Svensson, "HSM: Apredictive text entry method using bigrams", 10th Conference of EACL, In Proceedings of the Workshop on Language Modeling for Text Entry Methods, Budapest, Hungary, pp. 59- 99, 2003

[55] C. L. James, and K. M. Reischel, "Text input for mobile devices: Comparing model prediction to actual performance", In Proceedings of CHI-2001, ACM, New York, pp. 365-371, 2001

[56] Zi Corporation, eZiText. Technical report, 2002.http://www.zicorp.com

[57] LexicusDivision,iTap.Technicalreport,Motorolla,2002.http://www.motorola.com/lexiu

[58] http://www.eatoni.com

[59] Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing. Daniel Jurafsky & James H. Martin.

[60] F. Lindh, L. L. Larslarmostasluse, and A. H. Arthurholmerlingsuse, "Japanese word prediction." Japanese studies, Lund University, Sweden , 2011.

[61] K. C. Arnold, K. Z. Gajos, and A. T. Kalai, “On Suggesting Phrases vs. Predicting Words for Mobile Text Composition,” Proc. 29th Annu. Symp. User Interface Softw. Technol. -UIST ’16, pp. 603–608, 2016.

[62] M. Ghayoomi and S. M. Assi, “Word Prediction in a Running Text: A Statistical Language Modelling for the Persian Language,” Proc. Australis. Lang. Technol. Work., no. December, pp. 57–63, 2005.

[63] Dan'el Taklu, *Zemenawi sewasew quanqa Tigrigna*, Mega Press, Addis Ababa, 2012

# APPENDICES

## Appendix A: Geez Script writing alphabet

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ሀ | HA | ሁ | HU | ሂ | HI | ሃ | HA | ሄ | HE | ህ | H | ሆ | HO |
| ለ | LE | ሉ | LU | ሊ | LI | ላ | LA | ሌ | LE | ል | L | ሎ | LO |
| ሐ | HA | ሑ | HU | ሒ | HI | ሓ | HA | ሔ | HE | ሕ | H | ሖ | HO |
| መ | ME | ሙ | MU | ሚ | MI | ማ | MA | ሜ | ME | ም | M | ሞ | MO |
| ሠ | SE | ሡ | SU | ሢ | SI | ሣ | SA | ሤ | SE | ሥ | S | ሦ | SO |
| ረ | RE | ሩ | RU | ሪ | RI | ራ | RA | ሬ | RE | ር | R | ሮ | RO |
| ሰ | SE | ሱ | SU | ሲ | SI | ሳ | SA | ሴ | SE | ስ | S | ሶ | SO |
| ሸ | SHE | ሹ | SHU | ሺ | SHI | ሻ | SHA | ሼ | SHE | ሽ | SH | ሾ | SHO |
| ቀ | KE | ቁ | KU | ቂ | KI | ቃ | KA | ቄ | KE | ቅ | K | ቆ | KO |
| በ | BE | ቡ | BU | ቢ | BI | ባ | BA | ቤ | BE | ብ | B | ቦ | BO |
| ተ | TE | ቱ | TU | ቲ | TI | ታ | TA | ቴ | TE | ት | T | ቶ | TO |
| ቸ | CHE | ቹ | CHU | ቺ | CHI | ቻ | CHA | ቼ | CHE | ች | CH | ቾ | CHO |
| ኀ | HA | ኁ | HU | ኂ | HI | ኃ | HA | ኄ | HE | ኅ | H | ኆ | HO |
| ነ | NE | ኑ | NU | ኒ | NI | ና | NA | ኔ | NE | ን | N | ኖ | NO |
| ኘ | GNE | ኙ | GNU | ኚ | GNI | ኛ | GNA | ኜ | GNE | ኝ | GN | ኞ | GNO |
| አ | A | ኡ | U | ኢ | I | ኣ | A | ኤ | E | እ | I | ኦ | O |
| ከ | KE | ኩ | KU | ኪ | KI | ካ | KA | ኬ | KE | ክ | K | ኮ | KO |
| ኸ | HE | ኹ | HU | ኺ | HI | ኻ | HA | ኼ | HE | ኽ | H | ኾ | HO |
| ወ | WE | ዉ | WU | ዊ | WI | ዋ | WA | ዌ | WE | ው | W | ዎ | WO |
| ዐ | A | ዑ | U | ዒ | I | ዓ | A | ዔ | E | ዕ | I | ዖ | O |
| ዘ | ZE | ዙ | ZU | ዚ | ZI | ዛ | ZA | ዜ | ZE | ዝ | Z | ዞ | ZO |
| ዠ | ZHE | ዡ | ZHU | ዢ | ZHI | ዣ | ZHA | ዤ | ZHE | ዥ | ZH | ዦ | ZHO |
| የ | YE | ዩ | YU | ዪ | YI | ያ | YA | ዬ | YE | ይ | Y | ዮ | YO |
| ደ | DE | ዱ | DU | ዲ | DI | ዳ | DA | ዴ | DE | ድ | D | ዶ | DO |
| ጀ | JE | ጁ | JU | ጂ | JI | ጃ | JA | ጄ | GE | ጅ | J | ጆ | JO |
| ገ | GE | ጉ | GU | ጊ | GI | ጋ | GA | ጌ | TE | ግ | G | ጎ | GO |
| ጠ | TE | ጡ | TU | ጢ | TI | ጣ | TA | ጤ | CHE | ጥ | T | ጦ | TO |
| ጨ | CHE | ጩ | CHU | ጪ | CHI | ጫ | CHA | ጬ | PE | ጭ | CH | ጮ | CHO |
| ጰ | PE | ጱ | PU | ጲ | PI | ጳ | PA | ጴ | TSE | ጵ | P | ጶ | PO |
| ጸ | TSE | ጹ | TSU | ጺ | TSI | ጻ | TSA | ጼ | TSE | ጽ | TS | ጾ | TSO |
| ፀ | TSE | ፁ | TSU | ፂ | TSI | ፃ | TSA | ፄ | TSE | ፅ | TS | ፆ | TSO |
| ፈ | FE | ፉ | FU | ፊ | FI | ፋ | FA | ፌ | FE | ፍ | F | ፎ | FO |
| ፐ | PE | ፑ | PU | ፒ | PI | ፓ | PA | ፔ | PE | ፕ | P | ፖ | PO |

**Appendix B:** Sample list of Bigram data

| Bigram | | |
|---|---|---|
| **Tag1** | **Tag2** | **Probability** |
| N_pm | N_s | 0.002624241 |
| N_s | N_PRP_s | 0.002460226 |
| N_PRP_s | ADV | 0.001148106 |
| ADV | ADJ_pm | 0.0001640151 |
| ADJ_pm | N_p | 0.0006560604 |
| N_p | ADV | 0.0006560604 |
| ADV | N_V | 0.002296211 |
| N_V | V_REL | 0.002952272 |
| V_REL | ADJ | 0.005084468 |
| ADJ | V_IMF | 0.003936362 |
| V_IMF | N_s | 0.001968181 |
| N_s | V_GER | 0.003444317 |
| V_GER | UnKnown | 0.0008200755 |
| UnKnown | ADJ_pm | 0.0003280302 |
| ADJ_pm | ADV | 0.0001640151 |
| ADV | N_PRP_s | 0.001968181 |
| N_PRP_s | N_PRP_s | 0.0008200755 |
| N_PRP_s | N_s | 0.001312121 |
| N_s | N_s | 0.003116287 |
| N_s | V_GER_Spm3 | 0.0004920452 |
| V_GER_Spm3 | UnKnown | 0.0003280302 |
| UnKnown | N_p | 0.0003280302 |
| N_p | ADJ | 0.001148106 |
| ADJ | N_s | 0.0141053 |
| N_s | ADV | 0.002460226 |
| ADV | ADJ_p | 0.0001640151 |
| ADJ_p | V_AUX_Spm3 | 0.0001640151 |
| V_AUX_Spm3 | UnKnown | 0.0003280302 |
| UnKnown | ADJ | 0.0004920452 |
| ADJ | N | 0.0314909 |
| N | PUN_End | 0.002788257 |
| PUN_End | | 0.04986059 |
| | PRE | 0.0006560604 |
| PRE | N | 0.001640151 |
| N | N_p | 0.002296211 |
| N_p | PUN_dcot | 0.0003280302 |
| PUN_dcot | ADJ | 0.0003280302 |
| N_s | AD_s | 0.0003280302 |
| AD_s | N_p | 0.0003280302 |
| N_p | V_AUX_Ssm3 | 0.0003280302 |

**Appendix C:** Sample list of Trigram data

| Trigram | | | |
|---|---|---|---|
| **Tag1** | **Tag2** | **Tag3** | **Probabilty** |
| N_pm | N_s | N_PRP_s | 0.0006924009 |
| N_s | N_PRP_s | ADV | 0.0003462005 |
| N_PRP_s | ADV | ADJ_pm | 0.0001731002 |
| ADV | ADJ_pm | N_p | 0.0001731002 |
| ADJ_pm | N_p | ADV | 0.0001731002 |
| N_p | ADV | N_V | 0.0003462005 |
| ADV | N_V | V_REL | 0.001038601 |
| N_V | V_REL | ADJ | 0.0006924009 |
| V_REL | ADJ | V_IMF | 0.0005193007 |
| ADJ | V_IMF | N_s | 0.0006924009 |
| V_IMF | N_s | V_GER | 0.0006924009 |
| N_s | V_GER | UnKnown | 0.0003462005 |
| V_GER | UnKnown | ADJ_pm | 0.0001731002 |
| UnKnown | ADJ_pm | ADV | 0.0001731002 |
| ADJ_pm | ADV | N_PRP_s | 0.0001731002 |
| ADV | N_PRP_s | N_PRP_s | 0.0001731002 |
| N_PRP_s | N_PRP_s | N_s | 0.0005193007 |
| N_PRP_s | N_s | N_s | 0.0005193007 |
| N_s | N_s | V_GER_Spm3 | 0.0003462005 |
| N_s | V_GER_Spm3 | UnKnown | 0.0003462005 |
| V_GER_Spm3 | UnKnown | N_p | 0.0001731002 |
| UnKnown | N_p | ADJ | 0.0001731002 |
| N_p | ADJ | N_s | 0.0001731002 |
| ADJ | N_s | ADV | 0.0006924009 |
| N_s | ADV | ADJ_p | 0.0001731002 |
| ADV | ADJ_p | V_AUX_Spm3 | 0.0001731002 |
| ADJ_p | V_AUX_Spm3 | UnKnown | 0.0001731002 |
| V_AUX_Spm3 | UnKnown | ADJ | 0.0001731002 |
| UnKnown | ADJ | N | 0.0001731002 |
| ADJ | N | PUN_End | 0.0005193007 |
| N | PUN_End | | 0.002942704 |
| | PRE | N | 0.0001731002 |
| PRE | N | N_p | 0.0001731002 |
| N | N_p | PUN_dcot | 0.0001731002 |
| N_p | PUN_dcot | ADJ | 0.0001731002 |

**Appendix D:** Sample implementation source code

```
for (int it = 0; it < CandidateT.Count; it++)


   if (CandidateT[it].Contains('N'))
   {
       for (int m = Itagtype.Count-1; m >= 0 ; m--)
       {
           if (Itagtype[m].Contains('A') || Itagtype[m].Contains('U') || Itagtype[m].Contains('D'))
           {
               string Noun = CandidateT[it].ToString();
               string modifier = Itagtype[m].ToString();
               char[] splch = new char[] { '_' };
               string[] Nouns = Noun.Split(splch);
               string[] Modifiers = modifier.Split(splch);
               if (Nouns[0].Contains('m') && Modifiers[1].Contains('f'))
               {
                   CandidateT.Remove(CandidateT[it].ToString());
                   CandidateW.Remove(CandidateW[it].ToString());
                   break;
               }
               if (Nouns[0].Contains('f') && Modifiers[1].Contains('m'))
               {
                   CandidateT.Remove(CandidateT[it].ToString());
                   CandidateW.Remove(CandidateW[it].ToString());
                   break;
               }
               if (Nouns[0].Contains('p') && Modifiers[1].Contains('s'))
               {
                   CandidateT.Remove(CandidateT[it].ToString());
                   CandidateW.Remove(CandidateW[it].ToString());
                   break;
               }
               if (Nouns[0].Contains('s') && Modifiers[1].Contains('p'))
               {
                   CandidateT.Remove(CandidateT[it].ToString());
                   CandidateW.Remove(CandidateW[it].ToString());
                   break;
```

**Signed Declaration Sheet** 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 acknowledge.

**Declared by:**

Name: SENAIT KIROS BERHE

Signature: _____

Date: _____

**Confirmed by advisor:**

Name: YAREGAL ASSABIE(PhD)

Signature: _____

Date: _____