JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

(GRADUATE PROGRAM)

CONTEXT-SENSITIVE SENTENCE AUTO-COMPLETION FOR AMHARIC TEXT

MOHAMMED NURU

APRIL 2016

JIMMA UNIVERSITY

CONTEXT-SENSITIVE SENTENCE AUTO-COMPLETION FOR AMHARIC TEXT

MOHAMMED NURU

A Thesis Presented to the Department of Information Technology

(Graduate Program)

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology

APRIL 2016

JIMMA UNIVERSITY

JIMMA, ETHIOPIA

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

(GRADUATE PROGRAM)

CONTEXT-SENSITIVE SENTENCE AUTO-COMPLETION FOR AMHARIC TEXT

BY

MOHAMMED NURU

Approved by the Examining Board:

Principal Advisor	Signature
Debela Tesfaye	_____
Co-advisor	Signature
Seid Yesuf	_____
External Examiner	Signature
Dr. Fekade	_____
Internal Examiner	Signature
Teferi Kebebewu	_____

Dedication

This thesis work is dedicated to my parents, without whom, after the blessings of Allah, not all this would be possible.

Acknowledgements

I would like to thank many people that have involved themselves for the successful completion of my Master's Degree. Among these, I am very much indebted to Debela Tesfaye, my advisor, who has been giving me considerable support for my research work and without whom this paper would not have been completed. My thanks also go to Ato Seid Yesuf, my co-advisor, for his invaluable information.

This thesis work has benefited extensively from both the data collectors (i.e. Birhan, Meka, Solomon, Brhane and Yimer) and advised me (i.e. Mezmir and Hussein), staffs at Debere Berhan University. In addition, I thank Walta Information Center Agencies for providing access to the unannotated news corpus, and for discussion. I also want to thank my friends in Information Technology, and Computer Networking Departments, such as Imam N., Seid H., and Abdulkadir for their time, helped and friendship during the past two years.

Last, but not least, my thanks also go to Jimma University for its financial support to conduct this research.

List of Acronyms

CSSAC–Context-sensitive Sentence Auto-completion

IR– Information Retrieval

IPA–International Phonetic Alphabet

MLE –Maximum Likelihood Estimate

NLP– Natural Language Processing

POS – Part of Speech

QAC– Query auto-completion

QEM– Query expansion method

SLM– Statistical language model

T9 – Text on 9 keys

TF-IDF- term frequency with inverse document frequency

UKN–Unknown Word

Table of Contents

List of Figures

List of Tables

Abstract

Sentence completion is an unsolvable problem in the area of Natural Language Processing and Information Retrieval field of study. These-days, alertly increasing the number of electronic device users, who need to perform writing reports, searching files on their large-scale datasets, but have difficulty writing for different cases. Auto-completion is a general and specialized application to solve such type of problems. The main objective of auto-completion is reducing spelling error for poor spellers, keeping the syntactic structure of language, saving user's keystrokes, and the time and effort involved in typing. This paper presents a context-sensitive sentence auto-completion of Amharic text using combining features learned from the part-of-speech tagging to extract syntactic information and other features learned from frequencies, which include calculating the distance, similarity and length between input word and the possible recommendations using various techniques like tf-idf. This work completes the missed part of a sentence. The goal is then, when the user inserts the portion of a sentence, the system suggests the top five ranked sentences.

In general, the researcher has designed and implemented the prototype for three systems, such as distance similarity, pos tag and tf-idf and the hybrid of them. Finally, the researcher has also evaluated the performance of the systems, in four phases by preparing training and test set. Thus, based on the observed errors the hybrid sentence auto-completion has able to reached 81.82% completion accuracy. Unfortunately, the performance of the prototypes i.e. distance similarity, probabilistic part-of-speech tag information and tf-idf sentence auto-completion are tested using different experiments within the same input. The probabilistic distance similarity, part-of-speech tag information and tf-idf have achieved 21.21%, 31.82% and 80.03%, individually and in the order already mentioned. *Last, but not least, these methods rely on length, tf-idf and syntactical information to predict the most likely sentences. To that end, this research paper attempts to provide some recommendations that could bring about a change in the performance of sentence auto-completion in the Amharic sentence construction in order that current techniques of sentence completion could be employed from this time onwards.*

Keywords: auto-completion, prediction, sentence auto-completion for Amharic text

CHAPTER ONE

Introduction

1.1 Background of the Study

Languages are the most important things to transfer ideas and perceive the intended meaning of the writer or the speaker because of shared opinions or values. Each language offers a rich and unique insight into different ways of thoughts and lives as well as into the history of the myriad of cultures and peoples across the world. A poster on a wall of Beijing Language and Cultural University read, "If you talk to a man in a language he understands, that goes to his head. If you talk to him in his own language, it goes to his heart [64]." However, people who have access to computers in Africa tend to be educated in and socialized to some degree to use the official languages and thus less likely to actively seek to use their first languages in electronic technologies. Those people who use their first languages but not the official language tend to be not in a position to do much in this area, even if they wanted to.

Ethiopia is a linguistically diversified country where more than 85 languages are being used in day-to-day communications. Fundamentally, outsized spoken languages in Ethiopia are Amharic, Oromifaa, and Tigrigna and many more [10, 16, and 60]. Of all these languages, only Amharic has its own original scripts that inherited from Ge'ez and Saba. Even though, there are many languages being spoken in Ethiopia, Amharic is dominant, spoken as a mother tongue by a substantial segment of the population, and it is the most commonly learned second language throughout the country [10,60,61,65]. As a Semitic language descended from Ge'ez [10, 33, 34, 53, 60, 72], Amharic is spoken in the Horn of Africa. It is inherited from its common Afro-Asiatic ancestor. It is the official and working language of Ethiopia and the most commonly learnt language next to English throughout the country. This study concerns Amharic (Amharigna) text.

In any case, the ability to write a text in a mother tongue on the text editor could have great implications, as would the potential for speeding text processing. By various means, people who use a computer in the workplace have made reading and writing difficulties more obvious. However, the computer provides a possibility to aid persons with such difficulties. Even if a person can be helped by special education, s/he will continue to need support, particularly in writing. It is, therefore, necessary to develop programs with functions to aid in writing which can help users

spell correctly, to choose among possible desired words and to correct some grammatical mistakes. Research findings in Russia show that around 8% of the population has serious, specific reading and writing difficulties.

These days, Natural Language Processing plays a greater role in our day-to-day activity, especially in relation to processing natural language that includes completing the word, phrase or sentence in which the writers are intended to produce text documents using various text editors.

Text auto-completing supplications are an artificial intelligence system that used when the writer intends to write the first letter or letters of a word, or few words, predicts one or more possible words or phrases as an alternative for the users to select [3, 4, 22, and 58]. When the user is going to write a fragment of a sentence, the system will provide a list of related alternative words to complete that sentence [1, 17, 26, and 41]. Unless and otherwise the system gives the user no alternative words, the writer should input the next letter of the word or the next word of a phrase for the program to produce other set of words or phrases. Applications like mobile input interfaces, for instance, include word completion in which the user like the once used by the smart phone and other devices can complete a partially completed word to the full word intended. In this regard, scholars in the field, underlined that text completion system can play a greater role in assisting disabled persons to increase their performance of typing and speed for poor spelling [1, 17]. Other researchers have studied and designed Text Predict entry System for Amharic [20, 36, 52, and 69], Afaan Oromo [49], and Somali [70]. However, the findings of their studies were not without gap. The gap was related to the methodology they used and their scope. They were merely confined to statistical approaches for their word prediction. They were not in a position to consider syntactical approaches to handle the context of words in a sentence. Therefore, in this research work, my aim is to consider both syntactical and statistical approaches and find the neighboring context of information to reduce grammatical and spelling errors as well as to improve user satisfaction.

Sentence completion remains one of the an unsolved problems within the domain of Natural Language Processing, with the best results achieving just over 50% correctness [48] when given five alternatives. The situation also applies to our context. If we take Ethiopia, there is a limited research study on this area that could initiate the development of algorithm. As an initial point, the works in [20, 71] have included the statistical distribution of order of characters, and those in [36, 52, 69] have studied the distribution of words on Amharic language. Based on these observations

and results reported in [20, 36, 52, 69, and 71], the present researcher proposes context-sensitive sentence auto-completion for Amharic text. The excellent idea of this work is considered, whether the surrounding context is making sense. In general, the experimental result shows that sentence auto-completion is a significant application to solve the problems of sentence construction.

1.2 Statement of the Problem

In the previous section, it has been indicated that, though the researchers in [26] achieved high accuracy precision values, sentence completion remains one of the unsolved problems within the domain of Natural Language Processing, with the best results achieved just over 50% correctness [48]. The ultimate reason that motivated the researcher to conduct a research on this topic is, on the one hand, the limited knowledge the researcher has about the significance of sentence auto-completion for Amharic sentence construction. On the other hand, in our context, there is no research that has been conducted on the area so far. In short, sentence completion for Amharic languages is not an exploited research problem yet. Hence, software localization for independent language is not straightforward since the language behaviors (i.e. syntactical structure) of this language are different from English and other European languages. When we observe the Amharic sentence constructed from the perspective of different groups, some of these groups miss some characters of a word, be it in the initial, in the middle, or in the final and some words of a sentence as well. Others also miss the sentences' grammatical structures, maybe due to different factors, such as the increased typing of a text, regional dialects or/and the complex nature of longer sentences. Lack of good command of the Amharic accent can be another factor. This is the reason why this research work should concentrate on sentence completion here.

Nowadays, the number of computer users in the world, in general, and in Ethiopia, in particular, is increasing in a minute. People, in both contexts, may want to prepare and write their documents and file those documents using computer. However, in this busy world, preparing and writing a document using computer is not an easy task. A writer begins one sentence and ends with another. At this moment, problems at grammar level may occur, especially when the varied sentence constructions in the Amharic language lead to faulty subject-object-verb agreement, or they can occur purely on the level of logic, where the sentence is technically correct but makes no sense. However, societies need sentences that are complete, meaningful and acceptable by them to

communicate effectively. For this reason, sentence production cannot be viewed as an easy task since no one is perfect in his/her writing to construct a good sentence when an idea is raised.

It is obvious that Amharic, the official language at national level [10, 11, 33, 35, 37, and 60], is one of the major languages that are widely spoken and used in Ethiopia. Despite the fact that Amharic has a large number of speakers and is widely spoken and used in the country, no local research conducted on context-sensitive sentence auto-completion has been observed so far. The only research works available so far were all confined to word-prediction level. They did not see sentence auto-completion. In addition, the findings of these studies were valid for mobile users only. They under looked the demands of computer users. Undeniably, the findings of these studies might have helped the mobile users avoid their problems related to misspelling of words, time consumptions and slow typing skills and the like, thereby creating a conducive atmosphere for them to communicate effectively using the target language. However, the need of computer users is more than word-prediction. They also need sentence auto-completion. It is this situation inspired the present researcher to conduct this research.

Furthermore, most of the time, writers face difficulty deciding which word or phrase comes next to which other word or phrase and, as a result, they begin to worry about misspelling words, correctness of grammar, and entering text, just spending their time to construct a sentence in Amharic language. User interface design plays a significant role in enhancing the usability of a given product. One aspect of the user interface is the provision of efficient functionality for text input. Most applications, including the ones available on the constrained input devices (like mobile), have such facility. However, they are solely developed for few major languages in the world. Such applications are significant when cost of input is high and the input text tends to have repetitive nature. Besides, auto-completion is everywhere, including on search-engines (like Google), word processors (like MS-word, Open office), and email composition, where texts tend to be repetitive, programmed, etc.

As mentioned above, as far as the researcher knowledge goes, there are no usable Amharic sentence auto-completion techniques and tools at hand to solve the problems of the local people while they are typing texts. As a result, the development of such applications is not

straightforward, maybe as the number of words in a given phrase or sentence is unknown and very large. In addition, Amharic is known for having sentences with ambiguous meaning and many characters, making sentence auto-completion a difficult work. For this reason, the researcher's plan is to investigate the problems of sentence completion in Amharic language and forward possible solutions to these problems. Therefore, the following are the research questions that would be answered in the study:

- ✓ How a continuous Context-Sensitive Sentence Auto-completion for the Amharic language should be developed?
- ✓ Which sentence auto-completion algorithm is more appropriate for completing the remains portion of the Amharic sentence?
- ✓ To what extent the prototype works?

1.3 Objectives of the Study

The main objective of this study is to investigate and design a context- sensitive sentence auto-completion and implement a prototype of algorithm for Amharic text. Therefore, the following is a description of some of the specific objectives of this research work, which are attributed to:

- ✓ Review related literatures on the prediction and auto-completion in the area of NLP and IR,
- ✓ Determine the appropriate texts of Amharic language from the collected document to prepare for training and testing purpose,
- ✓ Identify both syntactical errors and problem of sentence clarity,
- ✓ Determine the appropriate algorithm which is related to the Amharic sentence completion,
- ✓ Build a model for predicting the Amharic sentences using the combination of futures (i.e. distance similarity, part- of-speech tag and tf-idf),
- ✓ Develop a prototype Sentence Auto-completion System for the Amharic text suggestion,
- ✓ Test and evaluate the performance of the prototype,

1.4 Scope of the Study

The study will be restricted to investigating sentence auto-completion in Amharic language using distance similarity, part-of-speech tagging and tf-idf. However, the system has trained only to use a medium-size of Amharic sentence documents. This is due to constraints, such as time and lack of readymade corpus, resources to advance work in NLP and Information Retrieval, etc. even where there is the will and expertise to implement. To train large corpus using NLP techniques requires high processing speed and large memory of the computer. Moreover, another limitation of this research work is the observed limited number of words prepared in the sentence. Further, the research does not concern the semantic methods and techniques.

1.5 Methodology of the Study

For conducting this study, that means to accomplish a task of sentence auto-completion one has to have statistical and syntactical information, such as length, the frequency occurrence of tags of a word in the sentence, and the frequency of sentence in the corpus. To achieve the expected result of the study, different approaches, such as the following are employed:

1.5.1 Review of Related Literature

Reviewed literatures from different sources (articles, books, journals, Internet and so on) to understand word and sentence completion. Moreover, related literatures will review to understand related works, conceptual works, and grammatical structures and systems in particular Amharic language.

1.5.2 Implementation Tools

In this study, we will use a number of tools in order to come up with the solution for the problem we need to address.

1.5.3 Experiment

Experiments will be performed to evaluate the performance of the developed prototype of a sentence auto-completion model. The researcher will use volunteer groups to collect and prepare test set to evaluate it.

1.6 Significance of the Study

There are benefits to be derived from the findings of this study. First and most, the findings of this study will be important to all application users on computer. Writers, especially those who have difficulties deciding which word or phrase comes next to which other word or phrase and worry about misspelling words, missing correct grammar, and entering text, just spending their time to construct a sentence in Amharic language, will find the result of this study invaluable. In addition, the result of this study will benefit the public at large. The researcher and the language user at large will use the system to assist themselves in producing text:

- It will enhance the speed of text production,
- It will enhance the reading of text,
- Improves text inputting efficiency of personal computer,
- Reduces the rate of producing misspelled words and grammatically wrong sentence,
- Increases the usability of application accepting text input from users,
- Increase the speed of communications in chatting, email composition, letter, reports etc.,
- It will be also contribute to future researches and development in the area of Natural Language Processing specifically in machine translation, speech processing, text processing, Information Retrieval, grammatical analysis, content and thematic analysis.

1.7 Organization of the Thesis

The focus of this research work is to solve the problems related to sentence auto-completion in Amharic text. This chapter introduces the problem of sentence auto-completion apart from putting it in the context of other relevant problems in Amharic language computing. The rest part of the thesis is organized in four chapters. In chapter 2, we provide a survey of review of related literature in word prediction, and related works in auto-completion that are investigated on different issues. In addition, in this chapter, the researcher will deal about some linguistic structures and syntax of the Amharic language is presented. Our selected area and proposed algorithms for the task of context-sensitive sentence auto-completion and methodology parts are presented in Chapter 3. In Chapter 4 deals about how implementation of a prototype and experimental results have achieved, in order to show the result of the developed system. Ultimately, in Chapter 5 talks about conclusion of the thesis work and finding, limitations of the current work, and recommend possible future out line works.

CHAPTER TWO

Review of Related Literature

Research takes the advantage of the knowledge that has accumulated in the past because of constant human effort. It can never be under-taken in isolation of the work that has already been done on the problems that are directly or indirectly related to a study proposed by a researcher. A careful review of the research journals, books, dissertations, theses and other sources of information on the problem to be investigated is one of the important steps in the planning of any research study [76]. To inform the current research, the researcher looked for other previous reviews of the computer science and linguistic researches. Accordingly, this chapter primarily deals with the state-of-the-art, relating to text auto-completion techniques and word predictions, which is the concern of this paper. The aim is to find out the gaps among the early papers and to get some insight into context-sensitive sentence auto-completion techniques. However, the researcher has found very important to review in advance some linguistic characteristics of Amharic language in order to lay the foundation for the concept of Amharic language.

2.1 Some Linguistic Characteristics of Amharic Language

Amharic, which belongs to the Semitic languages family, is one of the most widely spoken languages in Ethiopia. The language is spoken as a mother tongue by a large segment of the population in the northern and central regions of the country and as a second language by many others. It has its own script that is borrowed from another Ethiopian Semitic language, namely Ge'ez [21]. However, this work does not consider the spoken part of this language but its written aspect.

2.1.1 Amharic Language and its Writing Styles

It has already been noted that Amharic is one of the languages with its own scripts borrowed from the ancient Ge'ez language. Accordingly, when we see its writing style, unlike Arabic, which is from right to left, Amharic is from left to write just like English. Its scripts, like the scripts of all other Semitic forms of writing, are originally consonantal. The number and order of those consonants for Amharic and Arabic is not the same. Its vowel letters are ሸ፣ ሑ፣ ሊ፣ ሓ፣ ሔ፣ ሐ and ሖ. The consonants and the vowels must unite to yield a particular sound in the language. This,

however, does not apply to the sixth sounds of the language in the alphabet, which should sometimes stand alone depending on the sequence in which they appear in a word.

2.1.2 The Amharic Characters

The Amharic character set does not have uppercase and lower case letters. The character set has been incorporated in the International Unicode Standard. Its kind ሀ(ha) (U+1200), ሁ(hu) (U+1201), ሂ(hi)(U+1202), ሃ (ha)(U+1203), ሄ (U+1204), ህ (U+1205), ሆ(U+1206), ሇ (U+1208), ለ (U+1209), ሉ (U+120a), ሊ (U+120b), ላ (U+120c), ሌ(U+120d), ል (U+120e), ሎ (U+120f), ሏ (U+1210), ሐ(ha) (U+1211), ሑ (U+1212), ሒ(U+1213) etc. The total number of characters is around 344 [25].

As mentioned before, Amharic is the official language of Ethiopia. It is also the working language of some regions in the country. The present writing system of Amharic is taken from Ge'ez alphabet, which was the language of literature in Ethiopia in the early times. The Amharic writing system consists of a core of 33 characters, each of which occurs in a basic form and in six other forms known as orders. Each graphic symbol represents a consonant together with its vowel. The vocalic symbol cannot be detached from the consonant element. That is, Amharic does not use independent symbols for vowels. In [26], scholars discuss the Amharic script is a syllabic rather than an alphabet.

Nowadays, Arabic numerals are commonly used as Amharic numbers. Sometimes for official writing Amharic uses Ge'ez numerals [25].

2.1.3 The Amharic Syntax Structure

In this section, we introduce the syntactical arrangements and relationships between the part of speech in Amharic language and some other languages. Researchers observed in [31] noted that the knowledge required to order and group words together come under the heading of syntax.

Amharic sentences are usually shorter and simpler than English sentences. Amharic often adds syllables to nouns instead of separate words to indicate possession, verb tense, gender and number. English sentence structure is subject – verb – object agreement. If the complexities of a sentence increase, the structure also may extend to SVO-SO. Unlike English, Amharic has subject– object –

verb surface structure [24, 23]. For example, ‘my friend wants tea’ can be translated into ‘Guadegnaye shay tifeligalech’, literally translated as ‘My friend tea she wants’.

Other researchers have also discovered in their studies that the verb may not always come at last, as far as its position is concerned -it may be the object. The surface structures of certain languages such as Russian, Japannese, Hindi and English are shown in the following figure [23], showing that Russian language has a surface structure similar to the structures of English and Hindi.

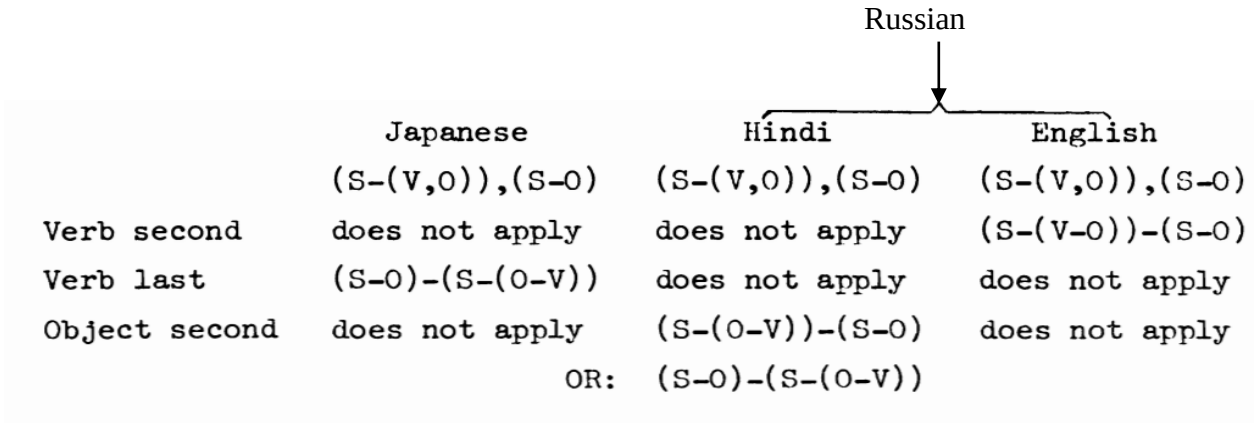

Figure 2.1: Russian, Japannese, Hindi and English language surface structures (source: [23])

In this study, however, the researcher concentrates on the syntax structre of the Amharic language only. Let us see the following examples:

A. እኔ በመኪና፤ ወንድሜ ግን በባቡር መጣ (ɨne bəməkina፤ wəndme gn bəbabur mət'a)።።((S-O)-(S-(O-V))

B. እኔ በመኪና መጣሁ፣ ወንድሜ ግን በባቡር (ɨne bəməkina mət'ahu፣ wəndme gn bəbabur) ።። (S-(O-V)-(S-O)

In Amharic, the structure of the sentence indicates that (ɨne bəməkina mət'ahu፣ wəndme Gn bəbabur). ‘In’ is a subject; ‘bemekina’ is an object; then, ‘፣’ is a punctuation; ‘wendime’ is a subject; ‘Gn’ is a conjunction; ‘bebabur’ is an object, and ‘Meta’ is a verb. Generally, it has an (S-O)-(S-(O-V)) structure. In the second sentence, B has the same meaning as A but a different structure (S-(O-V)-(S-O).

Bach's in [21] has argued that according to the preposition position shift and attachment with the verb, Amharic has a VSO syntax structure. In Amharic, the grammatical structure is SOV, superior to the grammar with SVO and VSO grammars. Therefore, determining this kind of rule is one of the specific challenges in this research.

The Amharic sentence is constructed from different tagged sets. In [24], some Amharic data contain the following tag sets: ACC = accusative, AUX = auxiliary, DEF = definite, F = feminine, IMP = imperfect, M = masculine, NEG = negation, subscribed O = object, P = prepositional suffix, PER = perfect, PL = plural, POSS = possessive, REL = relative marker, S = singular, TOP = topic. However, in this research work, when the morphological analyzer tool is supplied with **our corpus** as an input, it gives copula, and noun and verb tagged sets as an output. A noun is a word or agroup of words (other than a pronoun) used to name a class of any people, places, or things. A verb is a word used to express an action, to state an event or an occurrence and forming the main part of the predicate of a sentence, such as መጣ/mət'a, ይገናኛሉ/jgənaɲalu. A copula is a connecting word or group of words, in particular a form of the verb and connecting a subject and a complement. The other word classes that not grouped are unknown were categorized as "other".

2.2 Text Prediction Entry Techniques

Text prediction, as its name implies, is used to predict the word that the user is typing with the help of grammar rules for the English language, making word prediction more precise, reducing the number of key taps required, saving the user's time and achieving an optimization over the existing systems.

There are different types of entry techniques that include speech, chorded keyboards, handwriting recognition, various gloved techniques [1], scanner, microphone, and digital camera. Predictive text is one of the data entry techniques used in mobile or computer. In this technique, the system predicts what the user most likely intends to write based on some frequencies or other information.

Scholars in [5], using a large dictionary of words for disambiguation, have developed a new text entry method for mobiles with a single key- press per letter on a standard phone pad. They have also discovered a system that predicts letters with much combination of key-presses. According to

these scholars, there should be one or two words to match a given keystroke sequence (other combinations being non-sensible). This implies that, a predictive model of text entry method, if it makes use of a large dictionary of words, can suggest valid words to the user.

Other researchers in [6] have tried to address the problem of text entry on mobile phones. To solve the problem they have designed the text entry predictive methods that utilize the 12-key keypad and the model provides individual predictions for one-handed thumb and two-handed index finger use. Further, the same researchers have discussed the three most common text entry approaches throughout the paper such as Multi-press, two-key press and T9 methods. However, they cannot provide word prediction.

The researchers in [20] have proposed a new text entry algorithm known as EasyET (ETWirelessKeyBoard) that predicts the next word while the user is typing a current word. However, they did not consider a new text entry algorithm for other devices like computer keyboard.

2.3 Prediction Systems

The prediction system work was first put forward in July 1995 [19]. Researchers in [30] have developed "a generic word prediction" model. A researcher in [27] argued that word prediction is predicting the most likely words tokens or words to follow a given segment of a text. This means that, a few keystrokes produce complete words or word sequences, and the number of keystrokes necessary to generate texts will be reduced.

Prediction can be a character, a word or a phrase. In the case of character prediction, the next character can be predicted based on the previously inserted character(s), whereas in the case of word prediction the word is predicted based on some of the characters or words that were previously inserted in the editor [2]. On the other hand, phrase prediction system is guessing the number of appropriate phrases. Other researchers in [3] have studied the character prediction and its potentials for increasing recognition accuracy and provided a character predictor based on n-gram with an optimal length of context for application to handwriting recognition.

Researchers in [4] believe that having more accurate predictions will provide a number of advantages like improving the quality as well as the quantity of message production for young people, for persons with language impairments, and for those who have learning disabilities and disambiguate sequences from ambiguous keypads and correct spelling errors. Another researcher in [66] has presented FASTY prediction system that includes several innovative features. A FASTY word prediction system have included prediction of compounds, prediction of proper inflectional form based on the use of parsing, dictionaries based on general language corpora and on users' own texts and so on. Initially, a number of language were supported these are Dutch, French, German and Swedish.

The previous works, which were related to prediction and auto-completion systems, were to concern low-inflected languages such as English. The recent work, however, aims to consider highly inflected languages such as Amharic. According to some related researchers, reported in [66, 67, 68], morphological form variation is a problem in a prediction system. Morphological forms express mainly the number and, in some cases, the gender. When there are only fewer variations of a word, it is possible to store all of them in a dictionary. However, as highly inflected languages produce many forms, it may be difficult to store all of them. This is the main reason for the search of new prediction methods in languages with a wide use of prefixes, infixes and suffixes.

To predict the next word, prediction algorithm may include different type of information about words. Some of them use only statistical information about the words in the sequence, such as unigram predictor, bigram predictor and trigram predictor. Others may also include syntactic information about the words using part-of-speech tags.

Furthermore, other applications such as word sense disambiguation, probabilistic parsing, part-of-speech tagging, etc. [see in [31]] can be used in a word prediction system to develop it. A word prediction system facilitates the typing of text for users with physical or cognitive disabilities [18, 9]. As the user enters each letters of the required word, the system displays a list of the most likely probable words that could appear in that position. However, this study uses only part-of-speech tagging for finding the most frequent tag sequences.

In [19], said that current prediction function uses the word frequency lexicon, the word pair lexicon, and the subject lexicon. To train the system, they have used around 10,000 words with frequency information. The word pair lexicon consists of 3,000 reference words, each of which has an associated list of one to nine words that frequently succeed it. According to these researchers, the purpose of the subject lexicon is to allow the prediction system to adapt to the user's language by adapting the word frequency lexicon with those words of the user that are not in the lexicon or that have a rank higher than 1000. The user input is typically a prefix of a complete query q that the user intends to enter. The algorithm returns a list of k completions, which are suggestions for queries, from which the user can select.

The same researchers showed that when the input prefix is short (1 character) and the context is relevant to the user's intended query, then the weighted MRR of Nearest Completion is 48% higher than that of the standard Most Popular Completion algorithm. The Nearest Completion algorithm suggests the user's prefix input that is most similar to the recent queries s/he has just entered. However, when the context is irrelevant, Nearest Completion is useless. To solve this, these researchers have proposed hybrid completion, which is a convex combination of Nearest Completion and Most Popular Completion. Hybrid Completion is shown to be at least as good as Nearest Completion when the context is relevant and almost as good as Most Popular Completion when the context is irrelevant.

According to these researchers, Nearest Completion computes the similarity between queries as the cosine similarity between their rich representations. Nearest Completion is designed to work well when the user input has a non-empty context and this context is relevant to the query that the user is typing. They concluded that Nearest Completion relies either on no information or on false information and, thus, exhibits poor quality.

The Reactive Keyboard [15] works by attempting to predict what the user might want to select next on the basis of its preceding input. To predict the most likely next keystrokes, the system uses the sequence of the previous keystrokes. It uses an n-gram model for characters, created from text

samples and from the user's input. The model is stored in a special tree structure that allows partial matches between context and model to be found economically. The idea is to use the n-1 previous characters to predict the n^{th} one, where possible. If matches cannot be found, the context is shortened by one character, and the processes continue. Generally, keystrokes can only be predicted with limited accuracy.

For English language, VanDyke [38] has developed a word prediction system to provide the user with a list of grammatically appropriate words. The predictor works by transversing the search space produced by constructing the parse tree of the input sentence. The parser holds all possible structures for the partial sentence entered so far, and thus at each point in the sentence, it knows what syntactic categories can be in the next position. This eliminates a number of words to choose from, resulting in predictions that are more appropriate. However, it requires a considerable amount of work to parse partially the input sentence every time the user completes a new word. There is no evidence of testing the system to see whether the use of grammar helps improve the prediction performance.

Even-Zohar and Roth [40] have incorporated additional information into the learning process of their word-prediction system in order to learn better language models in comparison to prediction systems that use n-gram models. In the proposed prediction system, the local context information along with the global sentence structure is considered. For this purpose, a very large set of features, characterizing the syntactic and semantic context in which the word tends to appear has used, is learned for each word in terms of the features, and a learning method that is capable of handling the large number of features is used. A language for introducing features in terms of the available information sources is also defined.

Each sentence is represented as a list of predicates called the information source (IS) of the sentence. Features are defined as relations over the information source, or aspects of the structure of the sentence. A few examples of features are the adjacency relations between words, word collocations, and the part-of-speech tag assigned to each word. There are also complex features

such as the dependency relations between words and the role of each word or phrase inside the sentence, i.e., if it is a subject, object, etc.

A researcher in [39] has developed a new system whose scope was extended to include part-of-speech tag trigrams and word bigrams at each prediction point. The prediction algorithm has been interacting with a first-order Markov Model for words and a second-order Markov Model for part-of-speech tags. This considered conditional probability of a word by giving the probability estimation of the tag obtained by the tag Markov Model. The idea (i.e. assumption) of the prediction algorithm was first to obtain probability estimation for the tag of the next word using the tag Markov model. In the next step, probability estimation was found for the next word using the word Markov model. The tag probability estimation from the previous step was used to promote a rank of the words with the most likely tag. The tags unigram, bigram and trigram lexicons are created from the same corpus and used to build word unigram, bigram and trigram language models. Three texts of about 10, 000 words each have been used for evaluation. The system achieved a keystroke saving of about 43.2%, when given suggestions but no adaptations were used.

Fazly in [37], a comparison algorithm has been developed. The works of this researcher has achieved 90% accuracy using Unigram predictor. The researcher also employed four other predictors such as bigram, part-of-speech tag, and tags-and-words to predict the next word. These systems were capable of achieving an accuracy ranging from around 91% to 92.75%.The highest accuracy is that of the linear predictor.

2.3.1 Adding Semantic Information

The work of adding semantic information started in February 1996 [19]. Semantic analysis can also be very useful to familiarize the search for possible words to a well-defined semantic context [59]. Nevertheless, this approach has not given good results so far due to the difficulty of including semantic content to every word to process it in short time. While it is unlikely to result in any actual keystroke savings, it seems encouraging to construct some thought in the writing process to present the user with suggestions of words that are semantically congruent with the preceding words. Likewise, this thesis also does not consider semantic information.

2.4 Related Works

Recent works showed that many researchers have done on word prediction for certain languages like English, Hebrew, and Swedish etc. Moreover, they have done sentence completion for English language by using different methods to compare and contrast the performance of the system.

In this section, though, the researcher's main concern is to discuss some related works in the Amharic language, such as Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment and Word Prediction Model for Amharic Online Hand Writing System for Amharic language are presented. In addition to this, however, with the intention of getting some insight or input into the concern of this section as well as the present work, the researcher wishes to first discuss query auto-completion, phrase auto-completion and sentence completion for another language, for example English.

2.4.1 Text Auto-completion for the English Language

This section reviews query auto-completion, phrase auto-completion and sentence completion with regard to the English language. The purpose is to get some insight into sentence auto-completion for the Amharic language, whose discussion will be held later in the section that follows.

2.4.1.1 Query Auto-completion

In [25], Query auto completion has been defined as a pair of W and D, where W is a range of words (all possible completions of the last word which the user has started typing) and D is a set of documents (the hits for the preceding part of the query). To process a query means to compute the set of all word-in-document pairs (w, d) with w in W and d in D.

A query auto-completion (QAC) algorithm can accept a user input x, which is a sequence of characters typed by the user in the search engine's search box. According to their discussions in [17], the basic type of auto-completion algorithm, including topic clustering, query co-occurrence analysis, session analysis, and user search behavioral models. Query auto-completion algorithms differ from Query recommendation algorithms in their input (a prefix of a query vs. a full query) and in their output ((mostly) completions of the user's prefix vs. arbitrary reformulations of the query). According to another researcher, the framework for query recommendation algorithm can

leverage any state-of-the-art to construct its rich query representations. Researchers in [17] have proposed the first context-sensitive algorithm for query auto-completion and updated it.

2.4.1.2 Sentence Completion

An information retrieval approach is used to find the most similar sentence to a given user text inputs. Several researchers often face information retrieval problems. These problems could be solved by using approximate string matching without index structure [41]. Other researchers in [26] have developed an algorithm using inverted index files in order to increase the speed of a string matching retrieval model. In these studies, the occurrences of the query terms are merged and an approximate string matching is performed on the corresponding text positions.

As mentioned above, Grabski and Scheffer in [26] have developed an indexing algorithm that still ensures finding the best sentence but typically has a sub-linear behavior. They extend this approach into two respects. Firstly, given a sentence fragment of length m, they have searched for sentences whose initial words are most similar to the query fragment in the vector space model (rather that differing in at most k terms). Secondly, they have presented a pruning algorithm that interleaves accessing of indexed sentences with pruning of sentences that need not be accessed because they cannot be more similar to the query fragment than the best sentence currently found. This algorithm is based on an inverse index structure (see Figure 2.2 below) that lists, for each term, the sentences in which the term occurs (the postings). The system, however, considers only the user input to be a fragment of a sentence. When the fragments miss any one of its component, the system is unlikely (or will fail) to suggest the expected sentence.

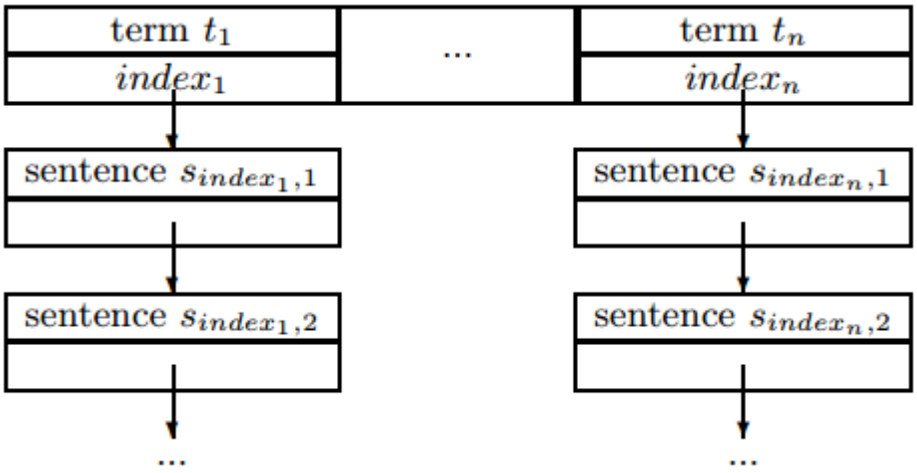

Figure 2. 2: Inverted index structure over the data (source: Grabski and Scheffer [26])

Finally, they have concluded that a substantial fraction of sentences can be completed successfully after three initial words. They have achieved up to 90% precision and 60% recall. However, they have considered only the initial fragment of the current sentence as input and have used domain specific dataset, disregarding all the preceding sentences and inter-sentential relationships. In our system, however, those problems solved by considering the input of words in the sentence may be inclusive.

Other researchers in [56] have developed an evaluation metric and protocol that is practical, intuitive, and independent of the user-specific tradeoff between keystroke savings and time lost due to distractions. According to their experimental analysis, they have concluded that N-gram based completion method has a better precision recall profile than index-based retrieval of the most similar sentence.

2.4.2 Text Input Methods for the Amharic Language

In [20], researchers have proposed the system 'Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment' that predicts the text according to the user input and facilitates message text exchange with Ethiopic scripts on a wireless phone. They have used two methods, such as Multi-press and Three-key input methods as well to develop this system. In order to determine the frequency of words in the dictionary, they have used three rating attributes. Moreover, they have developed a wireless application for Ethiopic text messaging. The same

researchers eliminated ‘superfluous’ characters, replaceable characters, Ethiopic numerals and labialized characters during text processing which reduce the number of characters in the font set. Accordingly, they have discussed two factors that have impacts on the prediction of the Amharic words. On the one hand, the structure of Amharic language and the statistical distribution of the order of characters, and the other was the habits of users to write words, phrases and sentences. Additionally, in his work, it was mentioned that to predict a word or phrase ‘apriori’ is difficult unless a dictionary is used in order to build a library of words, phrases and sentences commonly composed by the user. Eventually, they have achieved 40 % keystroke savings.

Furthermore, another researcher in [77] has designed the word prediction system that takes recognized handwritten characters from an online Amharic handwriting character recognition system. She has designed the algorithm without considering the constraints of the handheld devices such as PDAs (Personal Digital Assistant). Another researcher in [78] has considered these constraints and developed his system for PDAs. However, the works of both researchers in [77] and [78] are just to recognize and display a single character not to write Amharic text. By observing the gap, other researchers in [36] have adopted it to give the recognized character to the word prediction system as an input, and developed an online handwriting word prediction system for Amharic language. The algorithms required for assigning word frequencies from the corpus to the dictionary (lexicon) and for predicting words have designed and implemented experimentally by using N-gram models. In order to determine the frequency of words in the dictionary, he has used wordSmith tools. To predict the current or the next position of a word they only used word frequency information. In whatever way, the algorithm ignores the previous context and does not consider the syntactic structure. Finally, the researcher showed of 81.39% accuracy was achieved.

In this research work, an approach based on the syntactic analysis and information retrieval of the sentence that tries to extend the previous statistical prediction methods has described above. The algorithm may suggest words, which are not grammatically appropriate.

In general, as mentioned before, the idea of developing prediction system for Amharic is not new and couples of researchers have presented their work. The first one is a Master’s thesis focusing on word prediction online handwriting recognition for Amharic language [36] and the other is

Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment. As discussed in the above section the limitation of researchers work is related to the method in that they used frequency information only to predict the next word. In addition, another researcher in [69] design text predict entry system for Amharic text on mobile phone. This work has been used similar method in [36] but has used different training data set. The reliance on such statistical information only made the approach to lose the context of the previous word. According to recent works, researchers have manipulated word frequency effects in naming tasks are often considerably less than 100 msec. The problem of this fact is lack of context with word frequency interaction in the data. In this study, we include linguistic knowledge to solve such type of problem.

These researchers, particularly in [36, 69], found that written Amharic text has a high degree of redundancy. Based on these finding, it is natural to ask whether users can be supported in the process of writing text by systems that predict the intended next words, or sentences.

Ultimately, as mentioned above, the main ideas of learned knowledge is summarized as follows:

A. A sentence completion is an unsolvable, natural language processing problem,
B. The distribution of frequent co-occurring words is not sufficient for context-sensitive word selections,
C. The small size of a dataset has negative impacts to improve the accuracy of a prediction system,
D. To improve the performance of a system, researchers find a solution to decline the negative impacts and increase the accuracy of sentence completion system.

Based on this problem, we require different analyses to minimize the problem that Natural Language Processing and Information Retrieval Approach. In order to check and adopt this idea, the researcher designs a prototype to run different experiments and compare the performance of sentence auto-completion techniques. This context-sensitive sentence auto-completion is design to consider the ideas mentioned before.

CHAPTER THREE

Methodology of the Study

The aim of this study was to investigate and design a context-sensitive sentence auto-completion and implement a prototype of algorithm for Amharic text. Accordingly, the study has attempted to check if the written text is syntactically correct and whether there are any words missed by the writer in his/her sentences during sentence construction. It was also to suggest some possible solutions to the problems (if any) based on the findings. Accordingly, in this chapter, the researcher discusses the methods of he used to design and develop context-sensitive sentence auto-completion for Amharic text. The following are descriptions of procedures, and methods and techniques of the study and the samples used.

3.1 Data Preparation

Sentence completion is not an easy task to design a model since the length of sentences differs, syntactic information of words and sentences are ambiguous, semantic information lacks ad hoc and so on. As mentioned above, there is no ready-made corpus for Amharic, the target language. Thus, to design the model and accomplish the below tasks of sentence auto-completion, the researcher has performed various tasks as described in the subsequent sections. He first collected Amharic corpus from different sources and then cleaned the collected data.

To this end, the following subsequent sections are presents how the test and training data has been collected and prepared, and the system that designed and developed so far.

3.1.1 The Training Set

The training data set was obtained from different sources. The primary source of the data was the Amharic Wikipedia Dum[1]. This data was composed of text samples from different eras and different groups. It has a size of 1.9 GB in .xml file formats and included 12,000 Amharic articles.

[1]http:/www.mediawiki.org/xml/export-0.10/http:/www.w3.org/2001/XMLSchema-instance

http:/www.mediawiki.org/xml/export-0.10

http:/www.mediawiki.org/xml/export-0.10.xsd 0.10 am ዊኪፔዲያ amwiki http:/am.wikipedia.org/wiki...

The data contained xml and html tags, cases, invalid characters, multiple white space suppressions and UNICODE characters. Due to this, the researcher has sanitized unwanted data using the java programming language. The other sources of this corpus were Amharic newspapers, Amhara TV, and social media comments. In addition, the researcher collected 135-news texts from Walta Information Center Agency (i.e. since 2014/2015 G.C). Generally, the data set was domain-independent.

Finally, the size of this cleaned data was found to be 25.5 MB, containing 1,135,662 tokens and 243,205 lines. From this data, the researcher prepared 10,000 sentences to train the model. This corpus was named "MDS Amharic corpus." Unfortunately, the systems were trained with only half of the prepared sentences (i.e. 5,000 sentences) due to lack of enough working memory. This has convinced the researcher to include all resources. The task of information source most valid at hand was length, pos tag and tf-idf information. In general, corpus preparation is shown blow diagramatially. **Read the file→Removal of non-Amharic texts→Removal of non-sentences→Tokenization→corpus**

As shown above, the corpus preparation is not straightforward. Firstly, read the prepared .xml file extension from its directory. Then, in the data-cleaning component, clean the unwanted texts that are non-Amharic scripts and non-sentences. Next, the researcher has selected Amharic sentences. Finally, he prepared the training set as shown below.

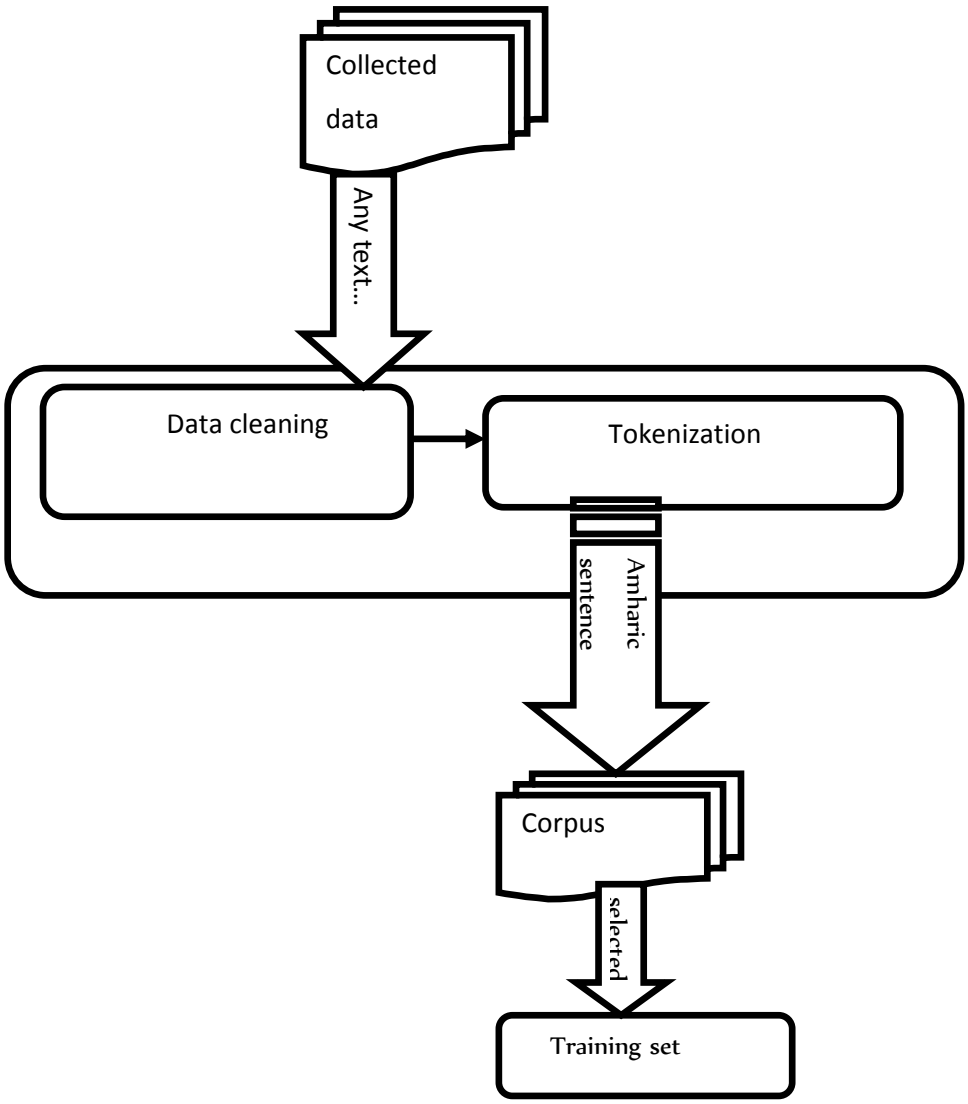

Figure 3. 1: Training data set preparation steps

The corpus contained two main items, (See Appendix A and B).

3.1.2 The Test Set

In order to investigate the effectiveness of the prototype, five volunteers were selected non-randomly to evaluate the proposed systems that were designed as a prototype. The researcher selected these people purposely because of their good computer background skills and long experiences in the field. The test set was prepared in two ways: by asking the evaluators to prepare 10 sentences of their own each and then by selecting 16 other sentences of my own randomly from the training set, adding up to 66. To this end, the participants were only informed about how long

their sentences in the Amharic language should be on the average. Since according to the researcher's corpus, the average length of the Amharic sentences was found to be fourteen. It was calculated as follows:

$$A = \sum T/N$$

Equation 3.1: Formula to calculate average length of a sentence in the Amharic language

While "A" refers to the number of average words in the Amharic language, particularly in this thesis work, "N" stands for the total number of sentences that exist in the corpus. "T" denotes the number of words that are found in the sentence.

Fortunately, the evaluators prepared fifty sentences of their own. Sixteen other sentences were selected randomly from the training set. Totally, sixty-six test set sentences were prepared so far. In general, a minimum of three words and a maximum of fourteen words were prepared.

3.2 Design of Context-Sensitive Sentence Auto-Completion

As the system relies on statistical information like length, co-occurring frequencies and other syntactic information like part-of-speech, the aim of collecting data was gearing towards getting these data.

Thus, the research approach used statistical and syntactical information extracted using three important features: (a) distance or length similarity, (b) POS information in constructing sentences and (c) tf-idf. To this end, the researcher developed different versions of his auto-completion system, each using only one of these features. He also combined these three features to enhance the performance and this is called the hybrid method.

The hybrid auto-completion method has three components, such as distance similarity auto-completion, probabilistic POS auto-completion and tf-idf weight auto-completion. The distance or length criterion was to calculate the difference between the length of the input string and the string that contains the fragment of sentences provided by the user. This feature moves sentences, which can be completed with a few additions, or deletion operation in the top rank. The detail is discussed in Section 3.2.1. On the other hand, using statistical information, probabilistic POS auto-

completion component works by examining the likelihood of the part-of-speech sequence exhibited by the sentences. Put differently, the system learns the most probable Amharic sentence structure in the statistical information from the part-of-speech categories of the words in the sentence. The detail is discussed in Section 3.2.2. TF-IDF auto-completion is the third component of the hybrid system that calculates the user input term occurrences within each sentence and the frequency of sentences that contains the user input. It also calculates, the overlap of words in the sentence using n-gram. The detail is discussed in Section 3.2.3. In addition, the detail of hybrid sentence auto-completion is discussed in Section 3.2.4. In general, how sentence auto-completion works looks like the following.

Suppose, we enter two or more words or phrases. The system displays the following top ranked sequence of sentences:

. . . S1, S2, S3, S4, S5

Accordingly, sentence S1 is automatically suggested first since it is the most relevant one to the user query term followed by S2, which is the second top ranked sentence, then S3 is the third top ranked, and S4 and S5 are suggested according to the sentence ranker algorithm. In addition to this, the researcher discusses the method used to design the system.

3.2.1 Distance-Similarity-Sentence Auto-completion Method

In this section, the researcher discusses how the auto-completion algorithm works to complete the user entry texts. Firstly, the algorithm counts the remaining number of words and calculates the difference between user input strings and sentences that contain these user input strings. In addition, the algorithm accepts, character by character, the character of the user input and finds the similarity of the input string to the sentence in the corpus.

The researcher focuses on the following three features in this model: (1) Counting the left length to be completed, (2) Calculating the similarity (or similarities) between the user input and the sentences in the corpus and (3) Sorting the calculated similarity (or similarities) of the two in an ascending order.

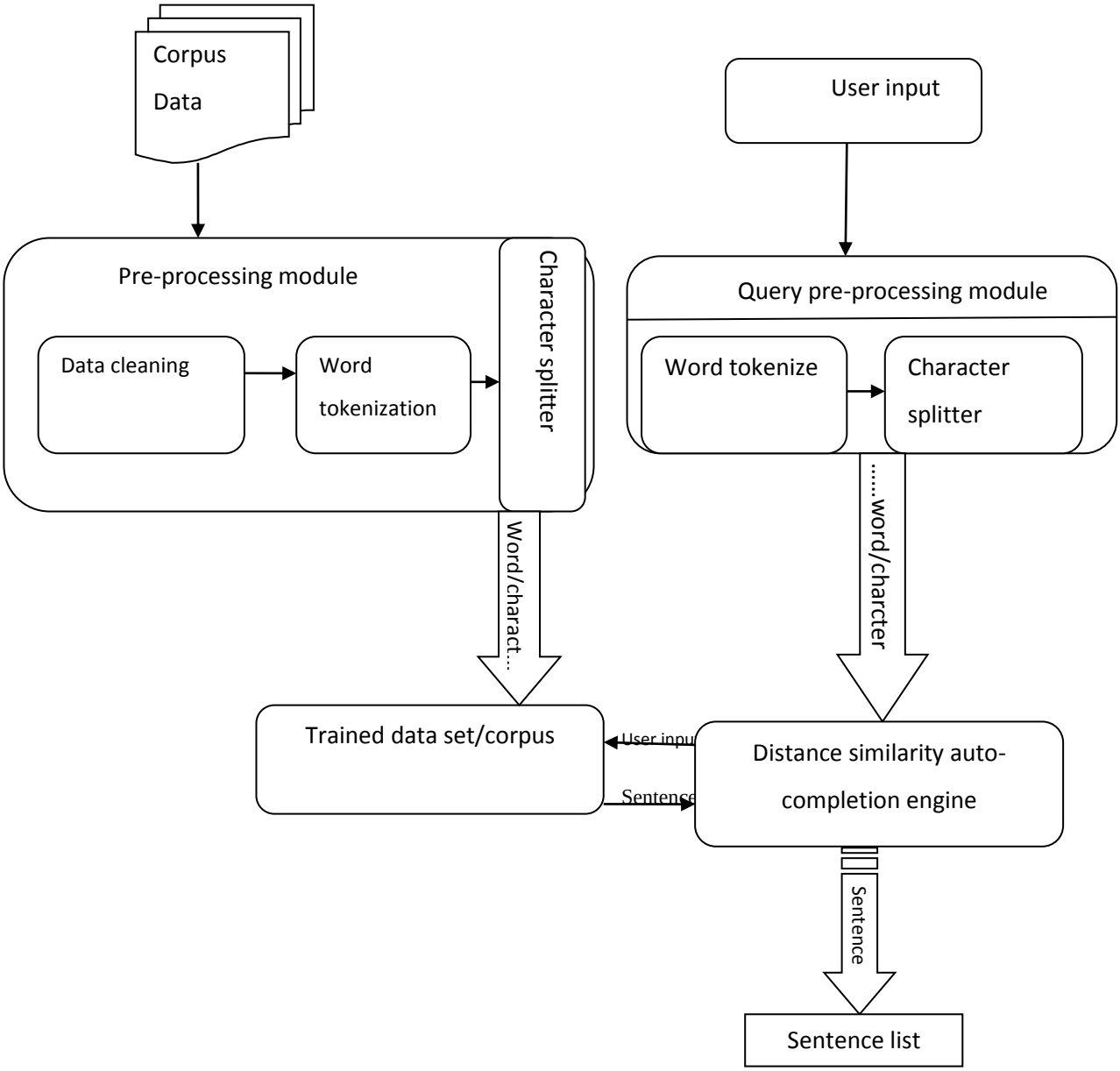

Figure 3. 2: Distance-sentence-similarity auto-completion architecture

The architecture has included collected data and training data, preprocessing module, user input, and distance similarity auto-completion. Collected and trained data are discussed in Section 3.1.1 in detail. Another component of this architecture is pre-processing module, which contains data cleaning, and word tokenization. These are used to clean unwanted data, and split document into sentences and words, respectively. The other major component of this system is distance similarity sentence auto-completion, which is used to calculate the distance similarity between user input

strings and the sentences in the corpus. Furthermore, based on the similarity value suggest the best sentences. Table two shows samples of sentences listed and their length displayed based on the following algorithm.

Input:
userQ=Fragment of sentence
Corpus=our corpus
sentL=Sentence list
Output:
sentence = list containing pairs of user input value with related sentence
Initialize:
resultL={}
similarity, Counter=0
IF ***counter*** *less than length of* ***corpus****:*
userQ *is equal to length of input*
For each ***userQ*** *in* ***sentL****:*
similarity *increment*
resultL*equal to* ***similarity***
sort ***resultL***
return ***resultL***

Figure 3. 3: Algorithm for distance similarity auto-completion

The auto-completion engine has taken statistics information from the training data set like number of words and length. To compute length between user input text and sentences, split the user input into words, and words into characters. Then compute user input characters within the sentence characters. Then after, count the similarity between these two strings. Based on similarity counted value of characters within the sentence, the system sorted the similarity value in ascending order. Lastly, auto-completion engine has produced the most likely top five optional sentences. The highest similarity length value printed out at the top, and the left follows. The challenge of this system is predicting the shortest sentences. The system is not considering the more frequent sentences. To complete the missed part of the sentence, the system concentrates only length. The other challenge of this system is, while the similarity length becomes similar values. It is a

problem whose solution the researcher suggests smallest index value displayed first. For example, if S1 is read before S2, S1 is displayed before S2; otherwise, S2 is displayed before S1.

Distance similarity auto-completion algorithm has a redundant sentence removal function that is used to handle distinct sentence list. If the sentence ranker lists the same sentences at a time, the redundancy sentence removal displays only one sentence at a time. In this system, the result or the suggested sentence is not only displayed the prefix of the entered text but to suggest the alternative sentences the user text input should be consecutive. Position of entered text in the sentence does not affect the suggestion system; it can display the sentence if the user entered is matched with the sentence fragment parts. The following figure shows the result conducted by this algorithm.

Str1=ከመልካም ባህሪያቸው መካከል (kəməlkam bahrijaʧəw məkakəl)

Str2=መልካም ባህርያት (məlkam bahrjat)

Str2 refers to query terms inserted by user and str1 refers to set of phrases (fragment of a sentence) that found in the corpus. The following figures show how to figure out the similarity between the texts character by character.

ከመልካም ባህሪያቸው መካከል (kəməlkam bahrijaʧəw məkakəl)

|||| | ||||

ከመልካም ባህርያ------ት (məlkambahrja------t)

Figure 3. 4: String similarity sample output

The distance between the two texts according to the algorithm is nine.

The following tabular Table 3.1 contains four columns and seven rows. The first row show ordinary numbers. The second column name represents Sentence List that is extracted from the training set. The third column name represents Length similarity that refers to the result of user input string and the sentence after some deletion and insertion. In addition, the last column name represents Query term that represents user input string.

Lastly, the researcher proposed the use of distance similarity by supporting corpus based retrieval. To suggest the sentence, an auto-completion engine calculates string similarity values. Here, therefore, the following table shows the similarity between user query term and sentences.

Table 3. 1: Top distinct sentence list

No	**Sentence List**	Length similarity	Query term
1.	በሩን ዘጋሁባት(bərun zəgahubat)	0.6363636363636364	በሩን ዘጋ(bərun zəga)
2.	እኔ በሩን ዘጋሁባት(ine bərun zəgahubat)	0.5	
3.	በአልማዝ በሩን ዘጋሁባት(bəalmaz bərun zəgahubat)	0.4117647058823529	
5.	እስከታወቀው ወቅት ቀን ድረስ (iskətawək'əw wək't k'ən drəs)	0.15	ቀን(k'ən)
6.	የዛሬን ቀን ምን በመስራት ላሳልፈዉ (jəzaren k'ən mn bəməsrat lasalfəwu)	0.125	

The following figures shows the values calculated by distance similarity algorithm (sample of string similarity), which were calculated similarities of user query term to every individual sentences. The following graph y-axis shows the length of number of inputted strings and x-axis shows distance similarity scored values.

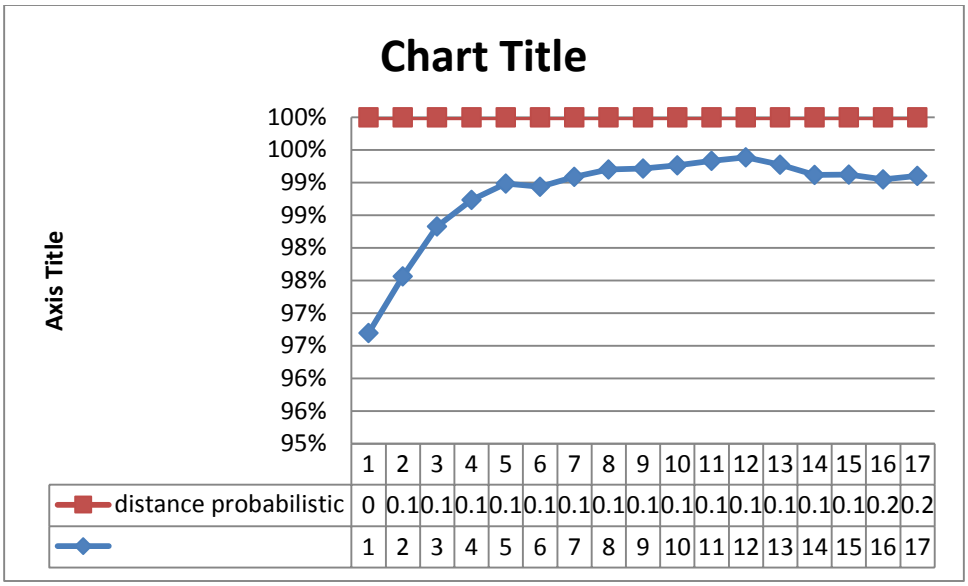

Figure 3. 5: This plots show query term vs. sentence length value

In this algorithm auto-completion engine, suggest sentences value that scored a minimum of distance similarity scored.

3.2.2 Probabilistic Part-of-Speech Tag Method

The researcher applied syntactical techniques to help the suggestion algorithm predicts the most likely word classes that are used in the sentence as much as possible ad hoc with the syntax of Amharic language. To analyze the tags, the researcher used HornMorpho[2], which is a software package developed by python programs to analyze and generate words in Amharic, Tigrinya, and Oromo [53]. This software package is freely available on the internet. It performs morphological analysis (such as possessor, part-of-speech tags, root, gender, number, and person, tense and other) of the word. For words, having more than one meaning returns the first n best analyses, which are ordered by their estimated frequency.

Firstly, the researcher imported this package and gave the corpus to the analyzer. Then, morphological analyzer (HornMorpho Software) processed and provided their part-of-speech tags for every word for each sentence inside the corpus. Then, wrote on the file and performed the frequency of tagged set at sentence level. This is used to know redundant tagged set in the

[2] http://www.cs.indiana.edu/~gasser/Research/software.html

Amharic text. In addition to this, extract basic syntactic structures based on probabilistic of sentence to attempt and adequate weight of the correct syntactical form.

The algorithm suggests the most likely sentences according to the user entry of fragment the sentence tags. To accomplish this suggestion, after accepting the user query term or inputs read the corpus files and make computation to rank the suggested sentence. The ranker sorted in ascending order based on the most consecutive tags values after and before the given input texts.

In this study, due to ambiguous meaning of words in the sentence some problems are anticipated, such as missing the correct tags name in the sentence. Ambiguous refers to words that have various possible tags (due to words having more than one meaning). Furthermore, some of the word that are entered from the user or writer tagged as "other" even it has a correct tag set categories. As mentioned before, due to the problem of the syntax analyzer, the researcher has not separated other parts-of-speech tag without verb, noun, and copula. Since the system does not have a warranty, whether the estimation it makes is right or wrong (because of the lack of all word classes). As a solution, a user can rearrange words to make towards the correct syntax. In this approach, the words that are not present in the corpus categorized as "other". This implies that the system asks special interactive session about the syntactic information associated with new words in order to complete the needed information. Figure 3.6 shows the architecture of probabilistic POS system. This diagram has included corpus data, preprocessing module, syntax analyzer, sentence auto-completion engine, user input, and training set.

It is obvious that corpus data are the collected data from different source (see in detail in Section 3.1.1). The other component of this architecture is a preprocessing module that is used to handle two major components, such as sentence and word tokenization. These components are used to split document into sentence and sentences into word, respectively. User input refers to any text that is inserted by anyone. Syntax analyzer is a software tool used to extract word tag information (see in Section 3.1.1). The other major component of this architecture is sentence auto-completion engine that is used two complete the next tag set. The context information is handled from the neighboring of inserted word classes

Thus, context linguistic information is learned from training data set to suggest the best probable sentence contextually for the user.

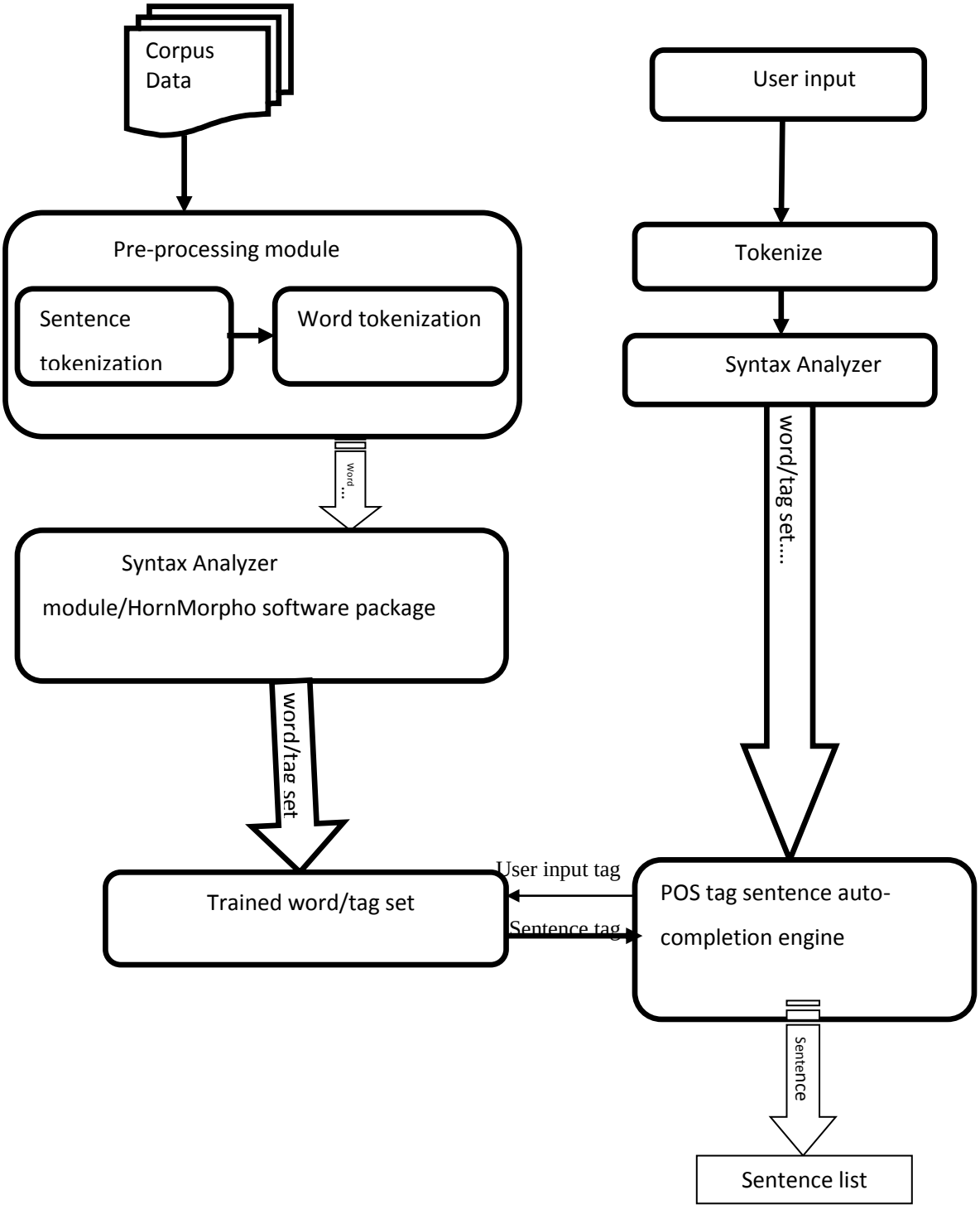

Figure 3. 6: Part-of-speech tag-based auto-completion architecture

Firstly, the system calculates the syntactical information to determine what the next tag is so. If one/someone input is provided into the system, it accepts and processes tag of every word. This task was assign to an algorithm that described in Figure 3.7. The following table shows the sample result produced by syntax analyzer and frequency counter.

Table 3. 2: Sample of list of sentences with tags and number of words in every sentence

Sentences with appropriate tag	**Number of word**	**Frequency distr.[3]**
መሸቶ /other እስኪነጋ/v ለኔ/n የዚህ/n ሰው/n ሥራ/v አዲስ/n ነው/cop [məʃto/other iskinəga/v ləne/n jəzih/n səw/n sra/v adis/n nəw/cop]	8	9
በልጅነት/n ጊዜዬ/n የአባቴን/n ሥራዎች/n እያየሁ/v እደነቅ/v ነበር/other [bəldʒnət/n gizeje/n jəabaten/n srawoʧ/n ijajəhu/v idənək'/v nəbər/other]	8	3
እሱ/n ግን/other መሥያ/n በልብ/n ነው/cop ብሎ/v በውሳኔው/n ፀና/v [isu/n gn/other muja/n bəlb/n nəw/cop blo/v bəwsanew/n s'əna/v]	8	4
የልምድ እጥረት እንጂ ከማንም አናንስም/jəlmd it'rət inʤi kəmanm ananm	5	5
በአንድ/n ወቅት/n ሁለት/other ሴቶች/n ስራ/v ለመቀጠር/n ለቃለ/n መጠይቅ/n አንድ/other መስሪያ/n ቤት/n ይገናኛሉ/v	12	3
ሁሉን/n ለመያዝ/n የሚሞክር/v አንድ/other ቀን/n እባብ/n ጨብጦ/v ይሞታል/v	8	10

Table 3.2 shows sentence with word classes and length of sentences (number of words existed for each sentence) and frequency occurrence of sentences in the corpus.

The concept of suggestion algorithm idea begins to calculate probability estimation for the tag of the next word using the tag Markov model in [37]. In the next section, 3.2.2.1 and other

[3] Distribution of sentences in the corpus

consecutive sections (i.e. section 3.2.2.2 and 3.2.2.3) presents how to estimate probabilistic values in detail.

Input:
sentL= sentence list in the corpus
Corpus= our prepared data set
User input= fragment of the sentence
Output:
Suggested sentence list
Initialize:
wordL={}
sentl={}
POSL={}
Dic=[]
*For each **sentl** the in **corpus**:*
*For each **wordL** in **sentl**:*
taggedset= " "
*For each **word** in **wordL**:*
IF word has POS in Gasser:
*For each **tuple** in **POSL**:*
*IF length of **tuple** is not equal **one**:*
*For each **str** in **tuple [1]:***
*IF **str** equal to **POS**:*
taggedset**=**taggedset**+W+**tuple[1][str]
*Return **taggedset***
*While **str1** less than length **taggedset:***
*IF **str1** is not in **Dic:***
***Dic** equal to one*
else:
***Dic**+one*
If the user input tag sequence is, matched within pos tag model
The suggest sentence is taken as significant if the user is accepted
Else
The suggested sentence has less significant
End:

Figure 3. 7: Probabilistic part-of-speech tag algorithm

The results produced by the algorithm (i.e. described in Figure 3.7) are shown in the table 3.1. To suggest the next probable tags cross check the input word and tags within the tagged training sets. In general, a system learns contextual information knowledge from probabilistic part-of-speech model.

3.2.2.1 Probability Estimation

To model auto-completion we used information about the counts of N-grams and assess the conditional probability of candidate words as the next word in a sequence-grams are token sequences of length *N*. N represents an arbitrary numbers 1, 2, 3… The model of the auto-completion can formulate as the ability to assess the conditional probability of the input word that given the previous words in the sequence. In this research, N is not fixed it depends on user query term word and character length to complete missing parts. If the user typed two consecutive words the system taken as a trigram (i.e. N=3).

$$P(w_{1,n}) = P(w_1)\, P(w_2 \mid w_1)\, P(w_3 \mid w_1, w_2) \ldots P(w_n \mid w_{1,n-1})$$

$$= \prod_{i=1..n} P(w_i \mid w_{1,i-1})$$

Equation 3.2: Conditional probability formula

More formally, we can use knowledge of the counts of *N*-grams to assess the conditional probability of candidate words as the next word in a sequence. A useful part of the knowledge needed to allow auto-completion could be capture using simple statistical techniques. We used statistical techniques to compute the probability of a sequence and likelihood of words co-occurring. Further, rank likelihood of sequences containing various alternatives to assess it. N-gram model uses the previous N-1 to auto-complete the rest. The N-grams work well for auto-complete of word or phrase if the test corpus looks like the training corpus.

$$P\,(w_n \mid w_{n-N+1}\, w_{n-N+2} \ldots w_{n-1})$$

Equation 3.3: The n-gram formula

3.2.2.2 Language Model

Ways of inserting words using the sequence of word are not context sensitive. The alternative ways to include the neighboring context is achieved used maximum likelihood estimate (MLE).

3.2.2.3 Maximum Likelihood Estimation

This can be used to make estimations of how probable a sequence of words is. The estimation is based on how frequent the sequence in a corpus. For the sequence of n words the MLE is.

$$P_{MLE}(S) = \frac{C(w1,\ldots,wn)}{N}$$

Equation 3.4: Maximum likelihood estimation formula

N refers to the number of sequences of length n in the corpus. For the estimation, one needs a corpus, which contains all possible sequences to produces the probability. This is a practically impossible and no corpus big enough exists.

One may decompose P(S) by:

$$P(S)=P(w_1)P(w_2|w_1)\ldots P(w_n,w_1,\ldots,w_{n-1})= \prod_{I=1..n} P(w_i \mid w_1,\ldots,w_{i-1})$$

Equation 3.5: Probabilistic sentence vs. word sequence formula

While "S" is a sentence, or sequence of words, P(S) the probability of sentence S to appear in the corpus, and P (w_n, $w_1 \ldots w_{n-1}$) which is equal to the probability of the word w_n to appear after the words $w_1,\ldots,w_{n-1}$.

The algorithm selects a word, set of words, or phrase able to complete the user input string. To this end, the algorithm relies on the probability and the N-gram information provides the pervious step. Put differently, taking the probability of the N-gram of the words in the corpus it calculates their probability and selects the one's with higher probability values. After selecting the best N-grams, the system provides the result to the user. Thereafter, the user applies selection if the required sentence is suggested so. Even the system provides the top ranked sentences the user can cross check whether the result is match or not.

3.2.3 TF-IDF Sentence Auto-completion Method

According to this study, the researcher focus on suggesting one of the given set of possible alternatives in the remaining part a sentence in syntactic information and distance similarity, for the given text. While contrast the problem of setting at hand in natural language, the researcher discusses and proceeds into information retrieval approach. For instance, if two or more sentences have one or more words in common, this is uncertain evidence of their similarity or difference.

Since to solve this problem identify tf-idf score, this is become common in information retrieval [41].

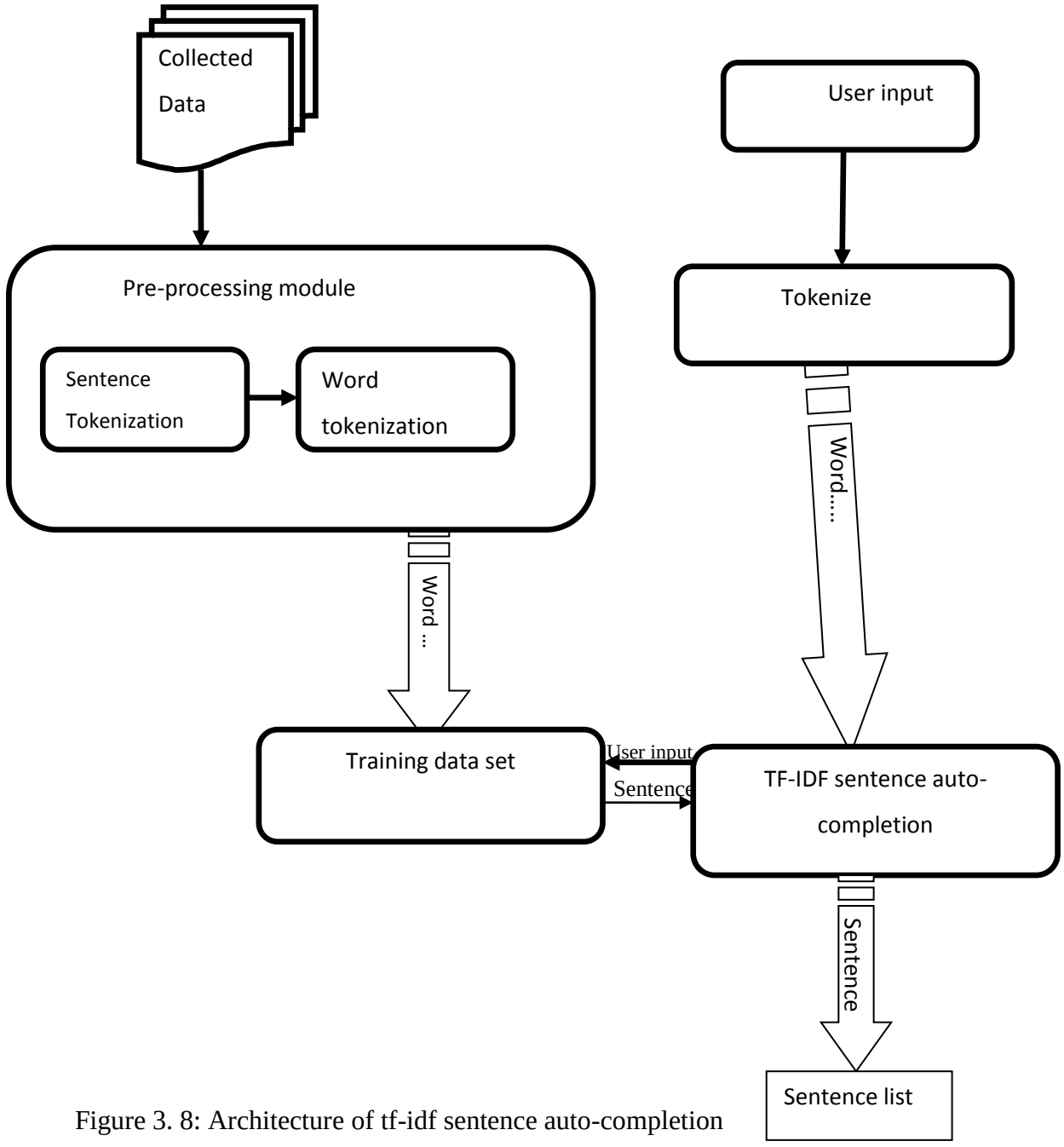

Figure 3. 8: Architecture of tf-idf sentence auto-completion

According to [28] information retrieval concerned with the construction of methods that satisfy a users' information need. However, the user has to encode the information need in a query. In this paper, the researcher discussed the sentence retrieval model to complete a remaining portion of user input string.

The above figure shows that the architecture of tf-idf, which contains several components, such as corpus, user input, sentence tokenization, word tokenization, pre-processing module, and tf-idf sentence auto-completion. Collected data is a data collected from different sources (i.e. discussed in Section 3.1.1) to train the model after pre-processing unwanted data. Obviously, user input is a string of text that given from the user. The other component is query-preprocessing package, as mentioned before, which is used to applied some text operation, such as splitting document into sentence and sentence into individual word. Sentence and word tokenization splits the document into sentences and sentences into words, respectively.

To this ends, the algorithm read the corpus from training data set and split into sentence in order to train the retrieval model. Then, calculate the statistical information of each sentence in the corpus. Similarly accept an input from the user to calculate weight and compare the value. This task is assigned for tf-idf sentence auto-completion engine. It calculates the weight of query term in the sentence and sorts that calculated similarity value to produce completed sentence. In addition, the tf-idf has frequency handle function that is used to count and handle statistical information about the sentence. Moreover, the system has another function called similarity counter that used to count sentences that contains the user query and to count the frequency of the user query term in the corpus. In general, the tf-idf auto-completion component is check whether the input is some similarity with sentences in the corpus or not. If there is some part of the query term is existed in the sentence calculate the similarity and sort it in ascending order. Then, the systems try to suggest the best probable sentences but might not the correct. To compute similarity weight of the term t in each sentence the following equation is used.

$$\boldsymbol{Tf\text{-}idf = S * log\left(\frac{N}{DF(t)}\right)}$$

Equation 3.6: Modified tf-idf similarity measure formula

While “N” is refers to the total number of sentences in the corpus, and “DF” is represents the number of sentence in which term t occurs. “S” stands for a calculated similarity weight value of a term within each sentence using n-gram overlap to solve the problem of standard term frequency in the sentence.

```
Input:

Import different modules like re, codecs,l3

    inilialize varaibles j=0, sentl, sentd={}

    corpus file

    tokenize this

    while(j<len(senl)):

        if senl[j] not in sentd:

            sentd[sentl[j]]=1

        else:

            sentd[sentl[j]]+=1

        increment j by one

    dic=[]

    for ww in (sorted(wdic.items(), key=itemgetter(1), reverse=True)):

        dic.append(ww)

      with codecs.open("SampleFreqDicModel.txt","a","UTF-8") as out:

            out.write(lll[0]+"\t"+str(lll[1]) +'\t'+'\n')
```

Figure 3. 9: Algorithms that Find Distinct Sentence List and Their Frequency

The standard definition of term frequency refers to check whether term t occurs in the sentence or not, this means that term t is the binary indicator. However, the researcher has considered may the document inclusive some part of the user input string. Therefore, to find the overlap of the term within each sentence, the researcher used word N-gram. Obviously, n = 1, 2, 3, which are referred to as a unigram, bigram, and a trigram models, respectively. While bi-gram predicts the probability of occurrence of a word based on the previous one word, tri-gram involves two words. It used n-unigram language model to compute frequency of occurrence of words in a given user input text. In this research, each distinct sentence has taken as document. Based on this the calculated results are sorted in ascending order to be suggested the best alternative sentences, if the calculated value is greater than or equal to threshold value.

3.2.4 Hybrid Sentence Auto-Completion Method

In this section, researchers present the hybrid sentence auto-completion method.

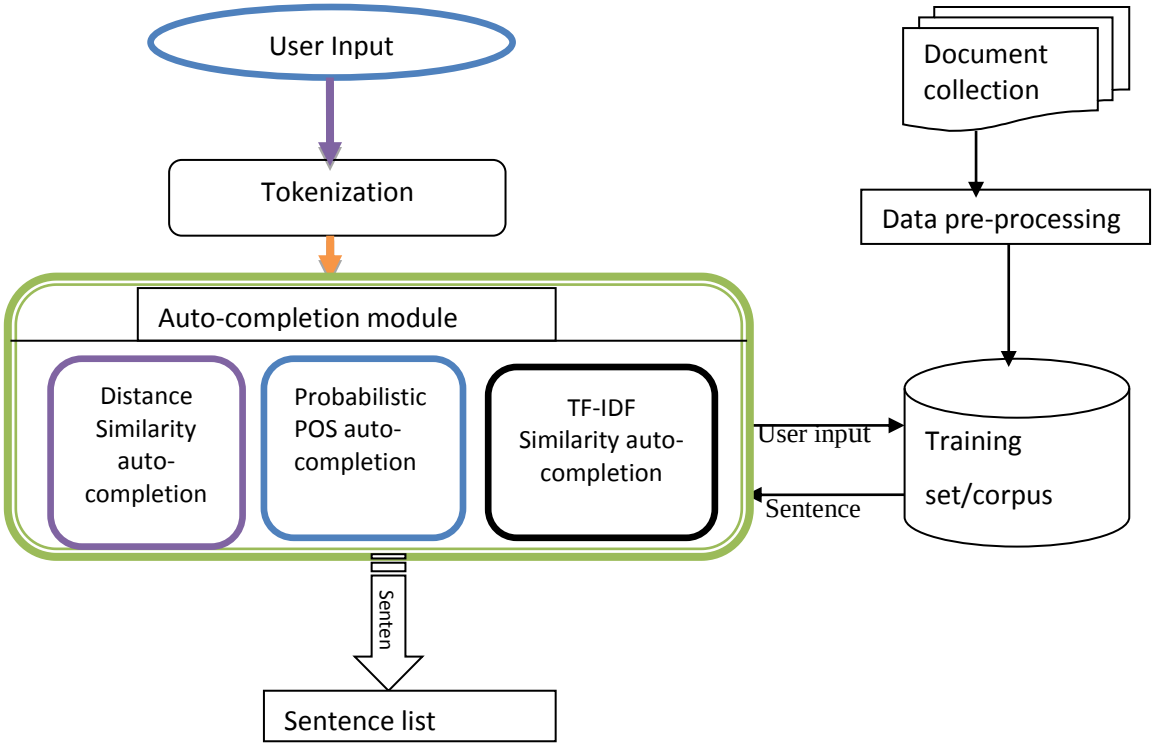

Figure 3. 10: This plot figure shows hybrid sentence auto-completion

The reason of developing and designing this system is, on one hand, the number of similarity values of the sentence existed in the corpus increased and on the other hand, the performance of

the system various when the algorithms ran individually. Accordingly, the hybrid method designed and developed to reduce the number of similarity values of the sentence and to increase the performance of the algorithm. As mentioned above, this method considers three parameters, such as length, part-of-speech tag and tf-idf. The architecture above in Figure 3.9 was designed to clarify it. The architecture contains a number of components, such as user input, tokenization, and auto-completion module, document collection, data pre-processing, training data set. While user input is a term that is inserted by writers on the graphical user interface whereas a tokenization is a process that breaks the user input string of texts into words, or other meaningful elements known as tokens usually by looking for whitespace to further processing it. The auto-completion module contains distance similarity, probabilistic POS model and TF-IDF similarity sentence auto-completion. These three methods are explained before in detail. Thereupon, the text part is send to auto-completion module. The document collection is refers to the collected data. Data pre-processing is another component of the architecture that used to clean the unwanted data. In this component, some data processing technique are applied to clean the unwanted data and tokenization is applied. As mentioned before, the nature of the collected data are not only contained Amharic text since the unwanted data was cleaned away. Then, apply tokenization, which is the act of breaking up an order of string into sentences and words. In the process of tokenization, some tokens like punctuation marks are discarded. Other tokens become the part of training set. Training data set is our corpus that is used to train the model.

As mentioned before, statistical information has taken from three basic components such as distance similarity, probabilistic POS models and TF-IDF similarity. Obviously, these components have used statistical information to complete missed part of user query term. According to user query, the system automatically suggests the best probable sentences depend on their statistical information. The novel idea of this algorithm is to predict the missing parts of a sentence complete contextually by considering neighboring of word classes or tags. Even though the algorithm completes the sentence automatically with the consideration of statistical information, it does not promise to keep the language syntax structure. It is because the tag sequence of sentences is not integrated with the three algorithms. That is why this study tried to integrate the above-mentioned algorithms to address the problem on this study.

3.2.4.1 The Hybrid Sentence Auto-completion Algorithm

Hybrid sentence auto-completion algorithm guesses the best probable sentence to satisfy user requirement based of the above listed parameters. The researcher believes that the algorithm predicts the sentence that should be more frequent, shortest length (less than or equal to the average length of Amharic, which is discussed in detail in Section 3.1.2) and having a correct syntax structure. An English grammar book defined sentence is looks like following [79]:

> *"A sentence is an arrangement of words that makes complete sense. It must be meaningful. It must, at the same time, be acceptable the speakers of the language" [79].*

Therefore, to get such type of sentence in this study has tried to combine some parameters, hereinbefore, as the name indicated the hybrid sentence auto-completion system is bundled from three different systems, such as distance similarity, part-of-speech tag and tf-idf.

Thereon, the performance of each system is a high difference among them. Consequently, this system does not miss the contribution of each system. Therefore, in order to get a good sentence, the researcher found the weight of each system to be even-handed the individual system and to advance the summation of each system rather than simply adding the similarity value as it is.

However, assigning a weight for each method is not straightforward, first taken the performance of each system from the experimental results. Then, calculate the proportional approximate value of one system over the other. As mentioned before, however, the researcher does not offer an equal chance of suggesting best sentences for all system. As a result, these systems have achieved a precision value of 21.21%, 31.82% and 80.03% performances of distance similarity, pos tag and tf-idf, respectively. From this, we can calculate a contribution weight of each method to a total. Accordingly, the three methods achieved value is 21.21%, 31.82% and 80.03% from 133.06%. However, the value from100 percentage is 16%, 24% and 60%, as order already mentioned. Therefore, the weight of distance similarity is covered with regard to tf-idf and pos tag precision, the weight of probabilistic pos tagging is covered with regard to tf-idf and distance similarity precision and the weight of tf-idf similarity is covered with regard to distance and pos tag the contribution weight of each method is 0.16, 0.24, and 0.6, respectively.

The question is how hybrid system sorts similarity value. To calculate these, firstly, accept the calculated similarity value from each system. Thence, multiple each with the above mentioned contribution weight. In general, to find the similarity of user input string for each sentence in the corpus that contains the user input string and to sort in ascending order, hybrid system used the following derived equations.

H=0.16*get simD () + 0.24*get simPOS () +0.6*get simTF-IDF ()

Equation 7: Equation used to calculate the weight in hybrid system

While "H" is refers the calculated similarity value using the above equation. The get simD (), get simPOS () and get simTF-IDF () are functions used to accept the similarity value of the sentence in distance similarity, POS tag and tf-idf algorithms.

Clearly, consider the following sample output examples, if s/he is provided an input to the system, firstly, calculate the similarity of this inputted term with each sentences in the corpus using the above three mentioned methods.

Figure 3. 11: Sample output sentence similarity vs. user input string for each system

Sentence list	Distance similarity	Pos tagging	Tf-idf	Hybrid
s1	*0.6*	0.34	0.8	0.6576
s2	0.4	0.24	0.5	0.4216
s3	0.43	0.41	0.3	0.3472
s4	0.56	0.14	0.34	0.3272
s5	0.89	0.15	0.4	0.666

Then, multiply each value with the weight of each system. Then after, add that result to sort sentences. As discussed above, the above table shows that one sentence has three values. Therefore, it requires algorithm that used to calculate the distinct weight of the sentence using hybrid method.

In the above table, the last column name i.e. hybrid is the calculated similarity values.

Input:

Read the corpus

Split into sentence

Split into word

User input- initializes

Get syntax analyzer

Tag untag words

Get the probability sequence of tag from POS probabilistic model

Get distance similarity

Get tf-idf weight

Assign the weight of each system to some variable

Calculate the similarity weight using the hybrid system

If the probability of the entire sentence greater than or equal to the expected threshold value

Suggest sentences which is taken as correctly suggested

Else

The suggested sentence might not be significant

End;

Figure 3. 12: Algorithm of hybrid sentence auto-completion

In general, the hybrid algorithm printed out the above sample sentence looks like: s5 first, then, s1, s2, s3 and s4, consecutively.

Fortunately, the following table shows the result obtained from the hybrid sentence auto-completion. The tabular table consists of two-column value and eight rows. The first column name

represents User query term, which is a combination of words typed by the user and missed parts of the sentence. The bold phrases or a word is refers to the missed parts of the sentence whereas texts that are not bold represent user inputted string. The second column name shows that the value that printed out using sentence auto-completion.

Table 3. 3: Auto-completion sample output examples

User Query Term	**Suggested Sentence**
መሸቶ እስኪነጋ ለኔ የዚህ ሰው **ሥራ አዲስ ነው**/ məʃto ɨskinəga ləne jəzih **səw sra adis nəw**	መሸቶ እስኪነጋ ለኔ የዚህ ሰው ሥራ አዲስ ነው məʃto ɨskinəga ləne jəzih səw sra adis nəw
በልጅነት ጊዜዬ የአባቴን ሥራዎች **እያየሁ እደነቅ ነበር** bəldʒnət gizeje jəabaten srawoʧ **ijajəhu ɨdənək' nəbər::**	በልጅነት ጊዜዬ የአባቴን ሥራዎች እያየሁ እደነቅ ነበር bəldʒnət gizeje jəabaten srawoʧ ɨjajəhu ɨdənək' nəbər
እሱ ግን ሙያ በልብ ነው ብሎ **በውሳኔው ፀና** ɨsu gn muja bəlb nəw blo **bəwsanew s'əna**	እሱ ግን ሙያ በልብ ነው ብሎ በውሳኔው ፀና ɨsu gn muja bəlb nəw blo bəwsanew s'əna
በዘመኑ **ዘናዊ** የህትመት መሳሪያ በሀገራችን የነበረ ቢሆንም በኖረው ባህል መሠረት በእጅ የተዘገጁ ___ ተመራጭ ነበሩ bəzəmənu zənawi jəhtmət məsarija bəhagəraʧn jənəbərə bihonm bənorəw bahl məsərət bəɨdʒ jətəzəgədʒu _____ təmərəʧ' nəbəru	በዘመኑ ዘመናዊ የህትመት መሳሪያ በሀገራችን የነበረ ቢሆንም በኖረው ባህል መሠረት በእጅ የተዘገጁ መጻህፍቱን ተመራጭ ነበሩ bəzəmənu zəmənawi jəhtmət məsarija bəhagəraʧn jənəbərə bihonm bənorəw bahl məsərət bəɨdʒ jətəzəgədʒu məs'ahftun təmərəʧ' nəbəru
ከእውነት የራቀን **መሳሳት በእርግጥ ተሳሳተ** kəɨwnət jərak'ən **məsasat bəɨrgt' təsasatə**	ከእውነት የራቀን መሳሳት በእርግጥ ተሳሳተ kəɨwnət jərak'ən məsasat bəɨrgt' təsasatə
የኢትዮጵያ እግር ኳስ ፌዴሬሽን የዲሲፕሊን ጥፋት **ፈፅመዋል ባላቸው ሁለት ተጫዋቾች ላይ የገንዘብ ቅጣት አስተላለፈ** jəitjop'ja ɨgr ʔs federeʃn jədisiplin t'fat fəs'aməwal balaʧəw hulət təʧ'awaʧoʧ laj jəgənzəb k't'at astəlaləfə	የኢትዮጵያ እግር ኳስ ፌዴሬሽን የዲሲፕሊን ጥፋት ፈፅመዋል jəitjop'ja ɨgr ʔs federeʃn jədisiplin t'fat fəs'aməwal
ሁሉን ለመያዝ የሚሞክር አንድ **ቀን እባብ ጨብጦ ይሞታል** hulun ləməjaz jəmimokr and k'ən ɨbab ʧə'bt'o jmotal	ሁሉን ለመያዝ የሚሞክር አንድ ቀን እባብ ጨብጦ ይሞታል hulun ləməjaz jəmimokr and k'ən ɨbab ʧə'bt'o jmotal

CHAPTER FOUR

Implementation, Experimentation and Evaluation

To answer the primary research questions, the researcher calculated the tag frequencies and the sentence frequencies of each sentence as well as length. Accordingly, in this chapter, the researcher will discuss the implementation of the prototype and the tools used for its implementation and the development environment it used to implement the user interface of the prototype and the experimentation of the study and its results in detail and will analyze these. To this end, the implementation of the study and the tools used in it will be discussed first.

4.1 Implementation

The purpose of developing a prototype was to demonstrate the experiment of the proposed context-sensitive sentence auto-completion system for Amharic text. In order to implement the algorithms and make the necessary experiments on this system, the researcher has used the python software with 3.4 versions. In Section 4.1.1 below, the tools used to develop the prototype and the development environment used to develop it will be presented in detail.

4.1.1 Tools Used and the Development Environment

The rationale for using the programming language and developing the prototype was, on the one hand, the fact that the nature of the collected data contained both Amharic and English texts, on the other, there was no standard evaluator to evaluate the performance of the sentence auto-completion system. Therefore, the researcher has used java and python programming languages to clean the collected data and to develop the prototype, respectively. Accordingly, Java NetBeans IDE 8.02 versions were selected to clean the corpus. As mentioned before, the python software was one with 3.4 versions. The researcher has not made any comparison between these selected programming languages.

Python is an effective programming language with a remarkable support in order to get the prepared statistics information from linguistic data. It has an efficient and a high-level data structure with simple but effective approach to object-oriented programming. In addition, its syntactic and dynamic typing features with its interpreted nature make it a powerful language for scripting and rapid application development [61,75].

Further, python supports what is called “Unicode” standard, which is a standard designed to allow characters from almost all languages in the world. The two characters “\u” followed by the hexadecimal (base 16) code for the character in the Unicode tables will represent that character. Python has a mechanism to assist these Unicode character “modules”. A module is a collection of functions (and other materials) which can be imported into a script and used within that script.

4.1.2 User Interface

The purpose of the developed interface was to evaluate the performance of the proposed system run on the prepared corpus. The interface has some user interface components that can facilitate the process. The interface to perform this operation is shown in Figure 12 below. These components are:

- Buttons – to initiate the suggestion process and to display the alternative sentences for the writer.
- An Input Area– to write the desired fragment of the sentence or words. In addition, Menu items are included.

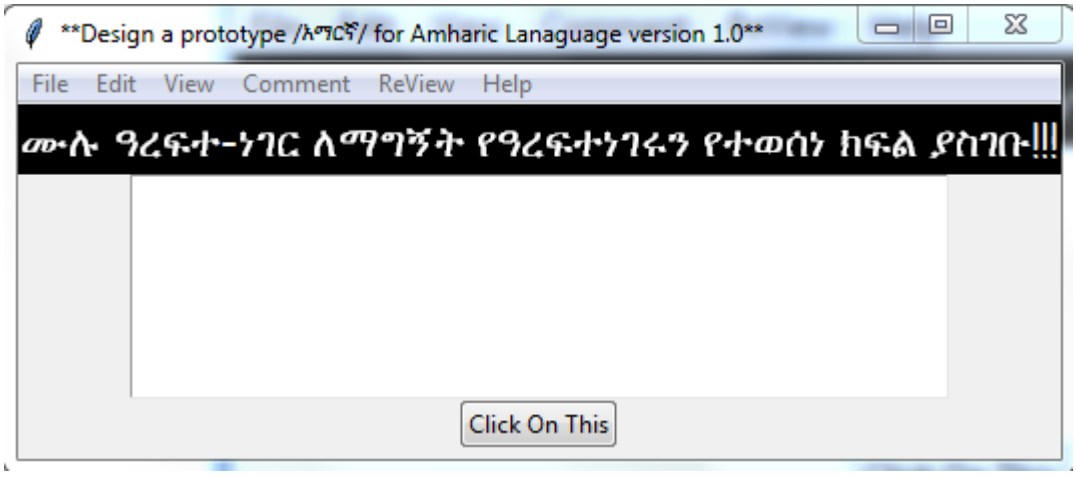

Figure 4. 1: Screen Shot of Sentence Auto-completion User Interface

The above figure shows the visible graphical user interface. A writer must write some portion of a sentence to initiate the suggestion engine. The inserted text is an unstructured text that can observe

the sample of the entered text on the graphical user interface as shown below:

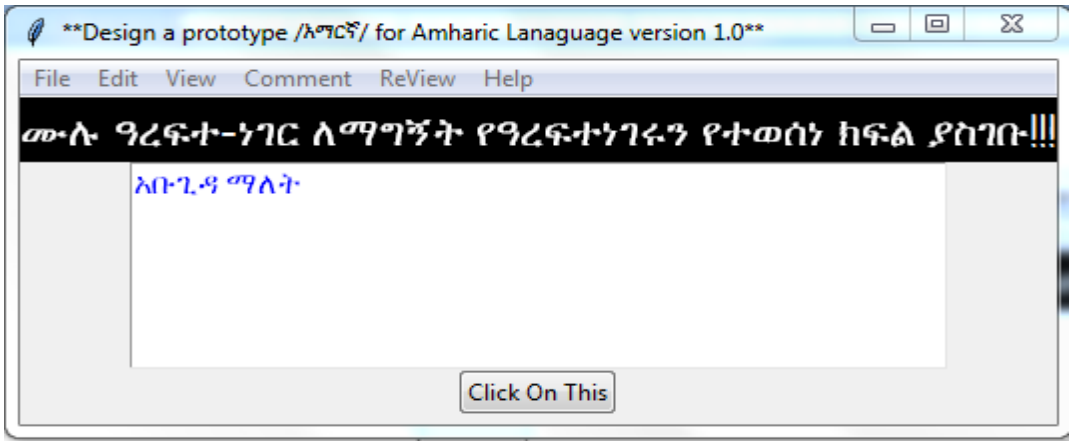

Figure 4. 2: Screen Shot of the User Input String

After inserting the user input string, a button-clicking action is applied to initiate the auto-completion engine. Then, the algorithms will calculate the probability of a sentence to provide it. As mentioned above, based on the calculated similarity score value, the system suggests the best sentences but may not be correct.

Figure 4.3 below shows the predicted sentence displayed on the graphical user interface. If the suggested sentence does not match with the sentence the user requires, the user can still write the next word or edit it.

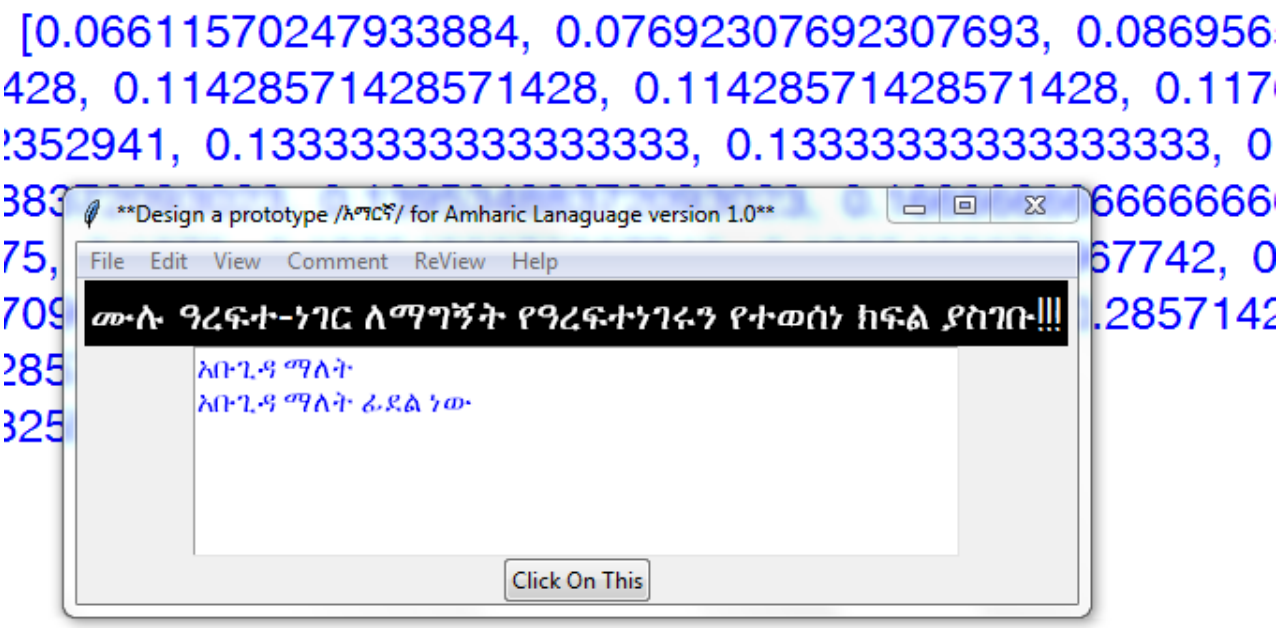

Figure 4. 3: Screen Shot of the Predicted Sentence

Based on the window size, when the user inserts a word or a phrase to get a full sentence, the system suggests incrementally up to five alternatives. As mentioned above, if the users' requirement is fulfilled, they can select the predicted sentence; otherwise, they can edit or type the next word of the required sentence. Further, the above screen shot background floating number figure denoted that the sample of the calculated similarity value.

4.2 Experimentation of the Study

The auto-completion algorithms for Amharic text will be trained on the already-prepared corpus. As mentioned above, the researcher will also prepare another test set of sentences, which is a disjoint set of training set. In general, the algorithms would be trained on the training set and evaluated on another test set.

4.2.1 Evaluation of the Methods

As mentioned and explained in Chapter Three, in order to test the algorithm, the researcher prepared sixty-six sample input sentences and a small training set containing all the sentences. The user inserts into the system the first few words or phrases of the required sentence and the system suggest the remaining part of the required sentence incrementally (i.e. plus one word or fragment of sentence each time). The system will then produce a list of words or phrases that are able to

complete the user input. The evaluators will give the same input words or phrases for each system to get the best sentences. As mentioned above, to investigate the effectiveness of the prototype, five volunteers were selected for this study through the same criteria as in [66]. These criteria are (1) having a computer ability to be utilized word processing for a minimum of two years, (2) being high school age, (3) having had no formal instruction of the use of auto-completion system, and (4) having had prior experience of accessing a computer (either standard keyboard or alternate access devices).

For each system, evaluators entered 77 inputs trails to measure the performance of a prototype. The number of input has selected randomly from the test set. In order to complete missed part of the sentence the user entered various input length on the GUI of the prototype. If the system output does not match with the required sentence, s/he can type the remains part of the sentence to initiate and refresh the suggestion engine.

The correctness of the systems having depends up on the majority of user acceptance of the suggested sentence. That means from five evaluators, if the three of them could accept even two of them does not accept, the researcher takes this as the result is correctly suggested by the system. Further, the experimental results are show in the following section.

4.2.4 Result of the Experiment

In this section, we mentioned and explained how the result is performed after the experiment was done. The experimental results were explained in tubular form as shown in the table 4.1, table 4.2, table 4.3 and table 4.4; these every all table contained 6 columns and 11 rows. To calculate the result every table groups categorized into correctly suggested and incorrectly suggested, sentences, based on the number of input typed on the system. It also showed the number of correctly suggested sentences and the number of missed sentences among the test set. The column name numbers of sentences test represent the total number of test sentences. The column name Length of sentence represents total number words contained in each sentences. The column name Number of Input represents the total number of inputs required to be suggest missed word for each sentences.

In general, we discuss the result for every model in the following consecutive sections. In Section 4.2.4.1, we show the experimental results using distance similarity matrix. In Section 4.2.4.2, shows the experimental result that provided by probabilistic part-of-speech tagging. In Section

4.2.4.3, shows the result that produced by similarity weight matrix using frequency of terms and sentences in detail. Finally, in Section 4.2.4.4, presents our novel method that is known as hybrid sentence auto-completion results and summarized precision results for every model in a table.

4.2.4.1 Results of distance-similarity-sentence auto-completion

The experimental result of distance similarity model was presenting as follow.

Table 4. 1: Test results of distance similarity auto-completion

Order number	Number of sentence	Length of sentence	Number of Input	Correctly suggested	Incorrectly suggested
1	1	9	8	0	1
2	2	13	12	1	1
3	2	14	13	0	2
4	3	11	10	2	1
5	4	8	7	1	3
6	5	10	9	1	4
7	7	4	3	1	6
8	10	6	5	1	9
8	12	7	6	2	10
9	20	5	4	5	15
Total	66	87	77	14	52

In Section 4.2.4, the columns name represented the order number, number of sentence, and length of sentence, number of input, correctly suggested, and incorrectly suggested. The last two columns name, correctly suggested and incorrectly suggested represents the number of sentences match with user required and missed the required sentences respectively based on user input. Since, therefore, table four shows the number of sentence that correctly suggested 14 from the 66 sentences. However, the system was suggested 52 sentences incorrectly. In general, the system result shows that some improvements wants to make towards correct sentence.

4.2.4.2 Results of Probabilistic POS Sentence Auto-completion

In this section, the result of the probabilistic part-of-speech tagging model has presented. This model shows the sentence that produced based on this model after certain inputs have inserted from the user.

Table 4. 2: Test results of probabilistic POS sentence auto-completion

Order number	Number of sentence	Length of sentence	Number of Input	Correctly suggested	Incorrectly suggested
1	1	9	8	0	1
2	2	13	12	0	2
3	2	14	13	0	2
4	3	11	10	1	2
5	4	8	7	1	3
6	5	10	9	3	2
7	7	4	3	2	5
8	10	6	5	3	7
8	12	7	6	5	7
9	20	5	4	6	14
Total	66	87	77	21	45

As mentioned before, the above table shows the result provided by probabilistic part-of-speech tagging. As a result, the system suggests missed part of sentences contextually. The system was observing the consecutive tags of user inputs to make correct sentences grammatically. In general, the above table shows the system was predicting 21 sentences from 66 test sentences. Unlikely, the remaining 45 sentences were incorrectly predicting.

In this system, the number of errors occurred while text processing, such as object-verb inversion, subject-verb agreement missed (i.e. like plural subject is inserted but the verb showed singular) since it expected check whether the sequence of consecutive word and tag are correctly matched or not.

4.2.4.3 Results of tf-idf Similarity Sentence Auto-completion

In this section, the experimental result is presented that was provided by tf-idf similarity using frequency of query term sentences as follow in table 4.3.In this system the numbers of incorrectly suggested sentences were 13. The system suggests 53 numbers of correct sentences from 66 trails. In general, this system has a better result achieved rather than the above-mentioned systems.

Table 4. 3: Test results of tf-idf similarity sentence auto-completion

Order number	Number of sentence	Length of sentence	Number of Input	Correctly suggested	Incorrectly suggested
1	1	9	8	1	0
2	2	13	12	1	1
3	2	14	13	2	0
4	3	11	10	2	1
5	4	8	7	2	2
6	5	10	9	3	2
7	7	4	3	6	1
8	10	6	5	8	2
8	12	7	6	10	2
9	20	5	4	18	2
Total	66	87	77	53	13

4.2.4.4 Results of Hybrid Sentence Auto-completion

In this section researcher, present the result produced by hybrid sentence auto-completion. Thus, sentence auto-completion achieved a better result than the three auto-completions. As shown in the following table, the result correctly suggested sentence by the system was 54 and the system incorrectly suggested was 12 from 66 trails.

Table 4. 4: Test results of hybrid sentence auto-completion

Order number	Number of sentence test	Length of sentence	Number of Input	Correctly suggested	Incorrectly suggested
1	1	9	8	1	0
2	2	13	12	2	0
3	2	14	13	1	1
4	3	11	10	2	1
5	4	8	7	3	1
6	5	10	9	4	1
7	7	4	3	6	1
8	10	6	5	8	2
8	12	7	6	10	2
9	20	5	4	17	3
Total	66	87	77	54	12

In general, the following table shows the summarized result produced by each sentence auto-completion system and the calculated precision values. In this work, only the precision results have calculated for each system.

The tabular table contains five column names, such as number of trail inputs, precision of Model A, precision of Model B, precision of Model C and precision of Model D. The first column name represents the number of trails that inserted into the system to evaluate a prototype. The second column name represents Model A, which is stands for distance similarity auto-completion precision value for each trail. The third column name represents Model B this refers to probabilistic part-of-speech tagging auto-completion precision value for each trail and the fourth column name represents Model C refers to tf-idf auto-completion precision value for each trail. In addition to this, the last column represents Model D, which refers to the hybrid auto-completion precision value for each trail.

The precision of the algorithms for a test set is the number of correctly suggested that is accepted by the user over the number of all proposed test sets (Equation 8). To this ends, the researcher calculated the precision value using the following formula:

$$Precision = \frac{\textbf{Number of correctly suggested sentence}}{\textbf{Number of test sentence}} * 100\%$$

Equation 8: Formula used to calculate the precision

Table 4. 5: Precision results for each system

Number of trail Inputs	Precision % model A	Precision % model B	Precision % model C	Precision % model D
8	0	0	100%	100%
12	66.67%	0	66.67%	100%
13	20%	0	80	80%
10	37.5%	12.5%	75%	75%
7	33.33%	25%	66.67%	75%
9	29.41%	23.53%	73.33	76.47%
3	25%	25%	72.72%	79.17%
5	20.59%	29.41%	75%	79.41
6	19.57%	32.61%	73.91%	80.43
4	21.21%	31.82%	80.03%	81.82%

As it can be noticed from the table, the result was calculated only precision score based on the number correctly suggested sentence and number of test. Accordingly, the above table noticed that the precision value of distance-similarity-sentence auto-completions achieved 21.21% from the total number of 66 trails. Whereas the probabilistic pos tag sentence auto-completion achieved 31.82% and tf-idf auto-completion achieved 80.03%. Further, the hybrid method achieved 81.82%. Therefore, this table result shows that the hybrid system is significant for Amharic sentence-completion system than methods that are particularly run.

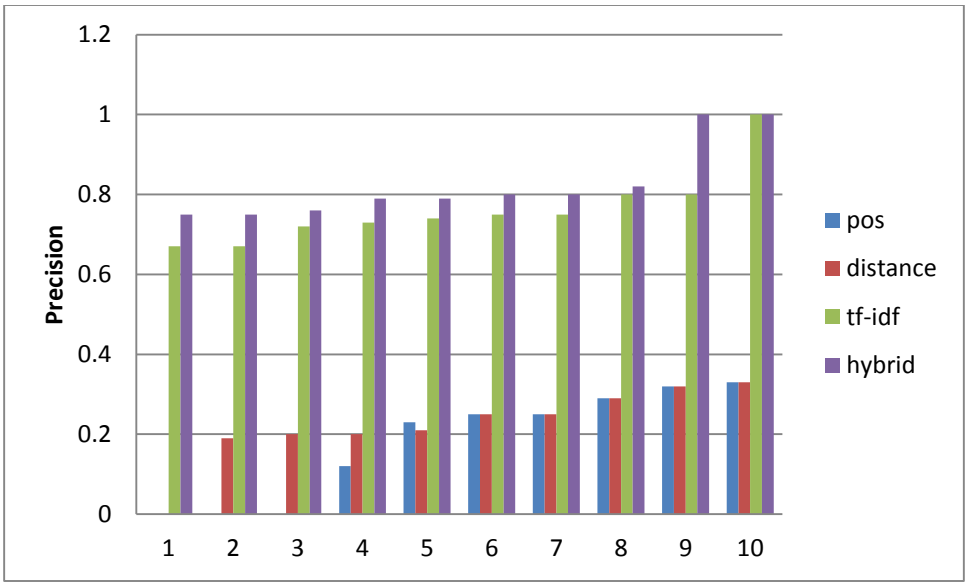

Figure 4. 4: This plot shows number of input vs. precision value

The above plotted figure noticed that both Model C and Model D have shown a better precision value than the other. In contrast, pos tag and distance similarity has shown low precision. The other models also show their significance performance precision value on the graphs.

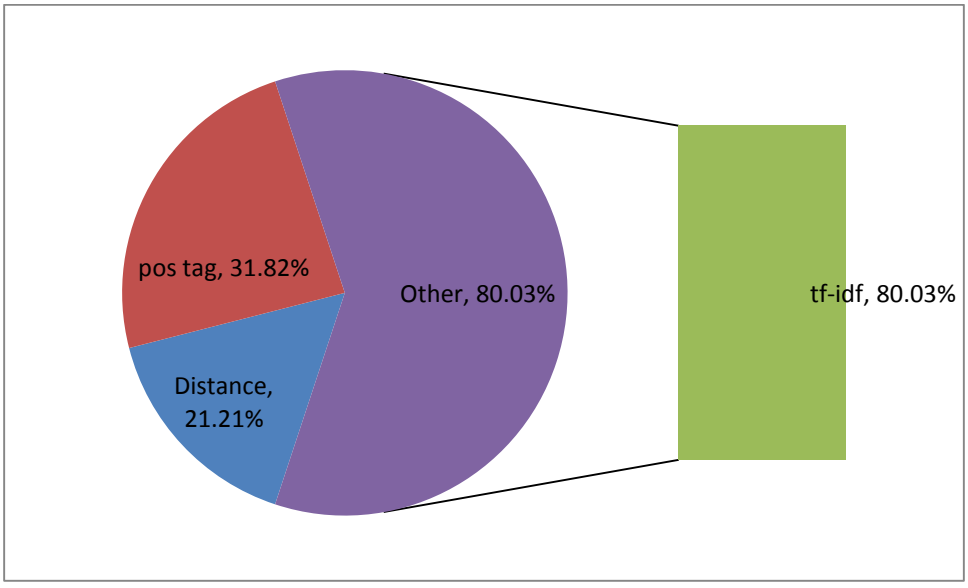

Figure 4. 5: This Pie Chart Shows the Significance of distance similarity, pos tag and tf-idf methods

In addition, the following figure shows the contribution of each method to a total.

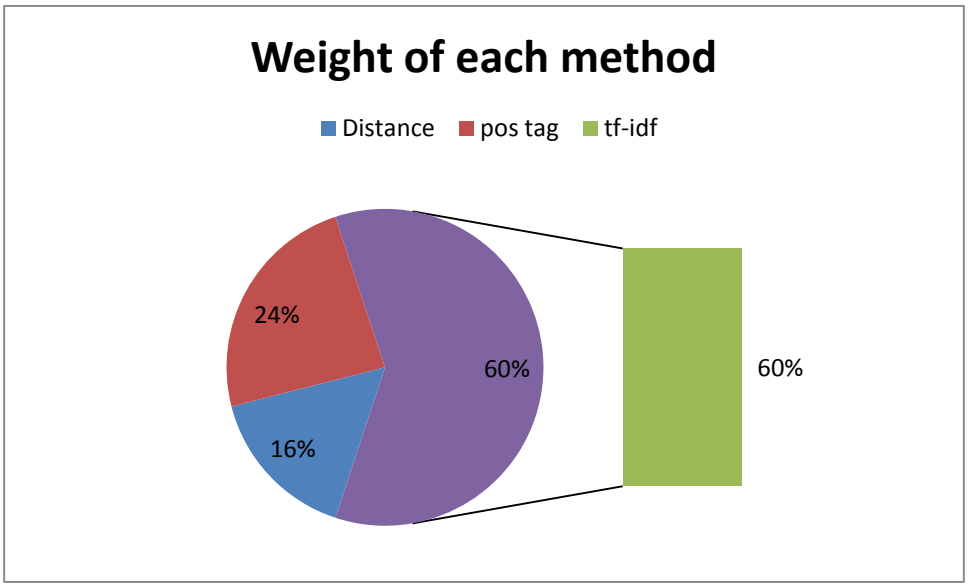

Figure 4. 6: This Pie Chart Shows the Contribution Weight of each Method to a Total

Further, in Figure 4.7, shows the significant precision percentage value of all methods that used in this study.

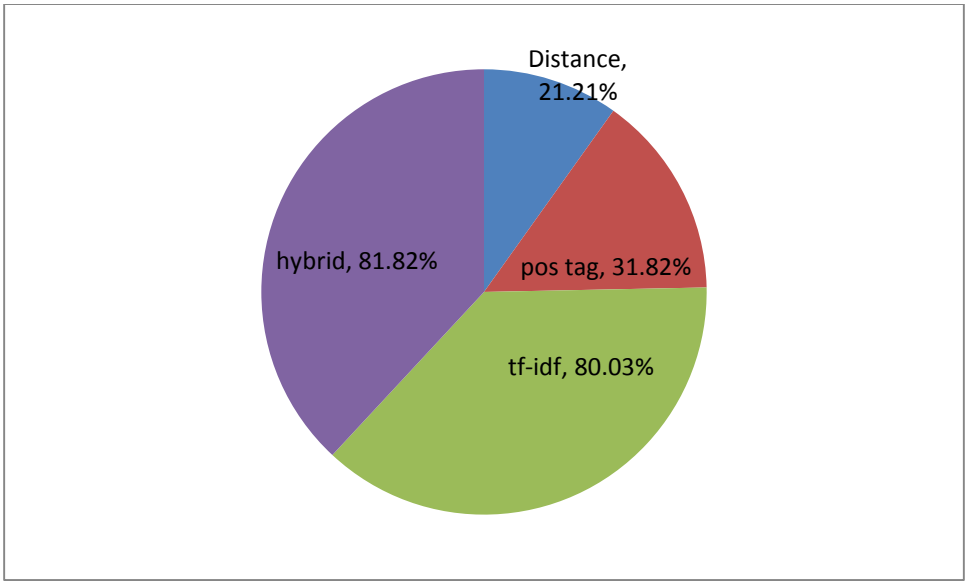

Figure 4. 7: This Pie Chart Shows the Significance of all Methods' Used in this Study

4.3 Discussion

As it is already indicated in the previous section, the experimental result shows that there is a difference in all methods with regard to suggesting the required sentence. Therefore, Humans can edit and determine the clarity of the sentence easily. Indeed, the researcher discusses the summarized experimental results for different scenarios that were conducted in each algorithm with different analysis of experimental result.

4.3.1 Discussion on the Result of Distance-Based-Sentence Auto-completion

As we have seen from the experiment result, completion of the remaining part of the sentence using distance similarity achieved a low performance. The reason behind this is that the system considers only length. This implies that it has a low significance for sentence auto-completion. However, if we integrate it with the alignment of a word, the performance of a system may increase. That means rather than length it is also better to consider also the position of words. In general, it is impossible to say the system has not significant to sentence prediction. Even though the system achieved a significance of 21.21% for sentence auto-completion, it is not significant for the system that requires the frequency of the sentence in the corpus.

4.3.2 Discussion on the Result of Part-of-Speech Tag-Based-Sentence Auto-completion

While using probabilistic part-of-speech tag method, the algorithm achieved a better performance result than distance similarity. It used us to handle the syntactic information of the sentence. However, the performance of the system is still low. The reason is, on the one hand, when we give the list of word to syntax analyzer; it produced most words of tags as a noun. On the other hand, it produced a tag of words as a noun or a verb even the word in the sentence has other tags category. In addition, the probabilistic part-of-speech models include longer sequences of tags. It is difficult to train the model using these researcher methodologies.

In general, the morphological analyzer accepts an input of sentences of a word and produces an output of sentences of four basic word tags, such as noun, verb, copula, and the unknown part of word as others. The other factor is redundancy of sentences and tags are not equivalent. The performance of probabilistic part-of-speech tag sentence auto-completion achieved 31.82%. This implies that it is low significance to sentence auto-completion for Amharic text but better than distance similarity sentence auto-completion method.

4.3.3 Discussion on the Result of TF-IDF Similarity-Based Sentence Auto-completion

Among the three various methods used to investigate the significance of sentence auto-completion , the researcher believes that the best sentence prediction method is tf-idf. Thus, tf-idf method is the most significant method for this study. The performance this method has achieved 80.03% precision value. Other researchers in [26] have also achieved up to 90% precision. However, they have considered only the initial fragment of the current sentence as input and have used domain specific dataset, disregarding all the preceding sentences and inter-sentential relationships. However, in his work, the researcher considers the input of words in the sentence may be inclusive.

In short, this implies that tf-idf based sentence auto-completion is high significance for the Amharic sentence auto-completion than the other two already mentioned. However, this performance is not sufficient for Amharic sentence completion. It requires several improvements like integrating the alignment of words and trained by a large data set.

4.3.4 Discussion on the Result of Hybrid Sentence Auto-completion

As it can be noticed in Table 4.5, the systems suggest a good accuracy result in tf-idf similarity system. However, in principle, the fact that the methods of sentence auto-completion for Amharic language involve completing user input string does not mean that prediction based upon sentence tf-idf. It does not mean that length or syntactical information frequency only. Instead, a particular method of sentence auto-completion, for this target language, the system can learn knowledge to predict the best sentence from all methods.

Scholars in [39] suggest that if syntactical information was integrated with other techniques, it might increase the accuracy of the system. Another researcher also in [56] argues that N-gram was a better result achieved for sentence completion. This implies that sentence auto-completion may require hybrid method to achieve better result. Accordingly, in this research work, the performance of the developed hybrid system was also achieved 81.82%. This system achieved higher precision value than the other did. This implies that hybrid sentence auto-completion is better for the Amharic text suggestions. However, we cannot say the performance is sufficient. In general, to

some extent, in this novel method, the error rate was reduced and more correct sentences were produced than the others that particularly run system.

Furthermore, in this method, the word order does not matter on sentence completion. Instead of said does not match, the system suggests right answer you are not looking it. For example, if s/he follows unusual writing sentence like "ẖede tmhrt atənak'o", the system rearranges the position of words in the sentence to make acceptable sentence that make (i.e. tmhrt atənak'o hede).

However, this system might not solve sentences that have ambiguous logic, for example like "እድሜዬ የእናቴን ሁለት እጥፍ ነዉ", ("ɨdmeje jəɨnaten hulət ɨt'f nəwu"). In this particular sentence, there is no syntax error but the logic may not be true. The proposed sentence auto-completion system might not solve such type of grammar error. The syntactic error was handled by probabilistic pos tag model. Furthermore, the researcher observes some writers are challenged by the problem like object-verb inversion. The verb appears before the object, and this make the sentence become out-of-your depth. However, in this thesis work, the novel idea of the system is word order does not matter. Since, such type of error has been tried to be handled using tf-idf method. This is not the only option to accept suggested sentence; if you want, you can edit the sentence to become toward the correct sentence. The other thing might not have knowledge about tenses that make the correct tense in the sentence.

Generally, as mentioned above, the test data consists of 77 inputs. The researcher ran each experiment on all texts and observed the significance of the similarity score value for each system. Accordingly, the test shows that:

- ✓ The hybrid auto-completion algorithm is significant if the statistical similarity score value is greater than 0.3.
- ✓ Other three auto-completion algorithms, such as distance similarity, pos tag and tf-idf is significant, if the calculated similarity score value should be greater or equal to 0.5.

In addition to this, some challenges have faced when the auto-completion algorithm suggests the best alterative sentence to satisfy user requirement. From the list of faced challenges, the first major challenge encountered was which words are suggested to complete a sentence whether a

word suggested by distance similarity or tf-idf similarity. That means both distance similarity and tf-idf similarity techniques suggested that the missed part is verb but the predicted word is different. Thus, selection and sorting of such type of ambiguity was the challenge of this thesis work. The reason why this is that we do not using and include semantics information in our corpus. For example, the first reason of reduced performance of the system was the ambiguous nature of the phrase like "s'gerədawa abəba", such type of fragment using this system is difficult to predict the next word.

CHAPTER FIVE

Conclusions and Recommendations

In this chapter, the methods followed to conduct the study are summarized and the results found are stated in brief. The chapter also deals with what should be done to solve the problems indicated.

5.1 Conclusion

Based on what has been found out as a result of the research study which has been stated in the previous chapters, the following conclusions are drawn:

Currently, one of the known research area in artificial intelligence is auto-completion. Auto-completion is a familiar feature, while text processing, which was used in several applications such as UNIX shells, modern text editors, web browsers and so on. However, this study has produced a context-sensitive sentence auto-completion for Amharic text. Sentence auto-completion is a research area, which is used to complete the missing part of a sentence.

The first system that means by using distance similarity auto-completion, the algorithm was achieved 21.21% and the second system means by using probabilistic part-of-speech model the algorithm was achieved 31.82%. However, from the experiment results these two methods imply that they have low significance for sentence auto-completion that runs in particular. Inside the major problem, there are several sub problems. As a solution to solve this sub problem, the researcher adds and test using tf-idf and N-gram overlap. This combined algorithm achieved a better result than the other two algorithms. The algorithm was achieved 80.03%. This implies that tf-idf has achieved better result than the other two. It has a better significant to sentence auto-completion for Amharic language. Additionally, this study has designed and developed the hybrid sentence auto-completion technique that includes a number of constraints, such as length, pos tag and tf-idf sentences information from the corpus. According to the experimental result, a better accuracy of 81.82% has achieved in hybrid method. To get these results, in general, this study has conducted collections of domain independent data set from Internet and Walta Information Center Agencies. In addition, according to the study, it can be concluded that if one/someone inserts a small portion of the sentences, the system can be completed successfully. Clearly, the hybrid

system has presented and tested for the Amharic language to complete the required sentence; and the system achieves a good performance and meets the objective of the study.

As mentioned and explained above, hybrid method achieved a better result but it requires some improvement to raise the effectiveness of sentence auto-completion.

5.2 Contributions of the Study

This study introduces context-sensitive sentence auto-completion models for Amharic text that propose the most likely ranked suggestion sentences for the user. In this study, different related recent works for English, Hebrew, Swedish, and German languages has been observed. However, the morphological and distributional characteristics of the Amharic language provides a challenge to directly apply the methods proposed for languages like English and German for Amharic sentence auto- completion.

As the aim of this research work was to investigate and design a context-sensitive sentence auto-completion and implement a prototype of algorithm, it has also the following minor contributions:

- ✓ Prepared and analyzed Amharic tagged sentence of the corpus for sentence completion,
- ✓ Develops a sentence auto-completion method for Amharic, based on distance similarity, pos tag and tf-idf based measure, which can minimize the amount time complexity during text processing,
- ✓ Development of a sentence completion method for Amharic based on morphological analysis (part-of-speech category of a word), this can provide an efficient grammatical arrangement of words in the sentence without human intervention,
- ✓ Investigated and implemented an integrated method called hybrid from distance similarity, Part-of-speech tags and TF-IDF similarity,
- ✓ Develop and evaluate our new proposed methods to compare the results.

5.3 Limitation of the Study

In this study, the hybrid techniques achieved a better result rather separately running techniques but due to different factors, the researcher considered only the Amharic language.

In addition to this, thought experiment, the system also lacks intuition to predict a sentence that includes a word like tomorrow is Monday or Tuesday. Instead of this, the user can add and edit their words to make correct sentences. In addition, the researcher does not consider the order of words in the sentence and punctuation marks.

Another limitation of this thesis work is that it does not consider the user pattern. Moreover, the evaluation has focused on the probability of correctly suggested sentence that lies within the same equivalence class as the sentence that a user is inserting. The time that is needed to make a prediction is also a very important factor. If s/he is typing faster than the system is retrieving proposals, then it is useless [56].

Even though the researcher implements and evaluates some mentioned methods, the study cannot include all other parameters and test using different methods. However, to some extent to develop and design sentence auto-completion used three techniques such as distance similarity, probabilistic part-of-speech tag and tf-idf. In addition, the researcher also employs hybrid of them.

5.4 Recommendation

Based on the research findings of this study, the researcher puts the following as major recommendations:

To increase the performance of sentence auto-completion, additional works should be conducted on semantics analyzer to consider the order of words in a sentence and to consider the meaning of sentences, automatic Amharic parser, and spell checker and word sense disambiguation and so on. Furthermore, the part-of-speech method used a long sequence of tags since it is difficult to handle using this researcher method. Therefore, it is better to use neural network to train a model.

In addition to this, the current system achieved nice performance just because it completes the sentence started by user learning from the corpus. However, users also want if the machine completes their ideas or thoughts. Since people sometimes start an idea and could not complete

which might arise from lack of deep understanding about the issue or forgetting the idea or soon. Therefore, as mentioned above, auto-completion system should also take in-depth morphology and syntactic and even semantic information in to account to help users in completing their ideas.

Moreover, the researcher also recommends designing the system able to learn the user's pattern from a long time usage to make the system adapt the user behavior. Besides, future researchers may integrate this proposed system for mobile phones and speech recognition. In addition, to adopt the system for other languages, it requires some modification like dataset, and grammars. The researcher has also observed no standard evaluation performance measurement for non-inflected languages. Hence, it is necessary to establish a standard evaluation that allows the comparison between different systems. Finally, the researcher believes that the above-mentioned should be integrated to produce a meaning full and best sentence into writing packages

References

[1]. B. McCaul and A. Sutherland, “Predictive Text Entry in Immersive Environments, "in *Proceedings of the IEEE Virtual Reality*, pp. 241, 2004.

[2]. Wikipedia contributor, (2007, April 13). "Predictive Text,” [online]. Available: http://www.wikepedia.com.

[3]. S. Homayoon and M. Beigi, "Character Prediction for On-Line Handwriting Recognition," Canadian Conf. on Electrical and Computer Eng., Toronto, Canada, vol. 2, pp. 13-16, 1992.

[4]. W. Gregory *et al.*, (2007, May 24). Effects of N-gram Order and Training Text Size on Word Prediction, [online]. Available: http://www.dynavoxtech.com/files/papers/LeMoHi99.pdf

[5]. M.D. Dunlop and A. Crossan , "Predictive Text Entre Methods for mobile,”*Springer-Verlag London Ltd personal technology*, pp. 134-143, 2000.

[6]. S. Miikaet *et al.*,"Predictive Text Entry Speed on Mobile Phones," Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI 2000, pp. 9-16, New York , 2000.

[7]. S. Abebeet *et al.*, "Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment", ITEs, Addis Ababa, Ethiopia, 2004.

[8]. A. Solomon, "Unsupervised Machine Learning Approach for Word Sense Disambiguation to Amharic Words," M. Sc. thesis, University of Addis Ababa, Addis Ababa, Ethiopia, 2011.

[9]. M.M. Behrmann and J.H. Graff, Word Prediction Software for Students with Writing Difficulties, 2006.

[10]. B.Philip, the Bradt Travel Guide, Bradt Travel Guides Ltd: England, 2002.

[11]. R.A. Allred, "Gender differences in spelling achievement in Grades 1 through 6,"*Journal of Educational Research*, pp.187-193, 1990.

[12]. C. Tam and D. Wells, "Evaluating the Benefits of Displaying Word Prediction Lists on a Personal Digital Assistant at the Keyboard Level," *Assistive Technology*, 21, pp. 105-114, 2009.

[13]. D. Ansonet *et al.*,"The Effects of Word Completion and Word Prediction on Typing Rates Using On-Screen Keyboards," Assistive Technology, no 18, pp.146–154, 2006.

[14]. D.R. Beukelman and P. Mirenda, "Supporting Children and Adults with Complex Communication Needs, "Augmentative and Alternative Communication, (3rd Ed.) Baltimore, pp. 77.

[15]. I. H. Wittenet *et al.*, "*The Reactive Keyboard*," Cambridge University Press, Cambridge, UK:ISBN 0-521-40375-8, pp. 43–44, 1992.

[16]. M. Herold *et al.,* "Typing speed, spelling accuracy, and the use of word-prediction," South African Journal of Education: EASA, pp. 117–134, 2008.

[17]. Z. Bar-Yossef and N. Kraus, "Context-Sensitive Query Auto-Completion," International World Wide Web Conference Committee (IW3C2), no. ACM 978-1-4503-0632-4/11/03, 2011.

[18]. K.Sundarkantham and S.M. Shalinie, "Word Predictor Using Natural Language," 2007.

[19]. A. Carlberger *et al.,* "Constructing a database for a new Word Prediction System," TMH-QPSR, vol. 37, no. 2, pp. 101-104, 1996.

[20]. S. Abebe *et al.,* "Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment," ITEs, Addis Abeba, 2004.

[21]. B. Emmon, "IS Amharic an SOV language?" journal of Ethiopian Studies, pp.9-20, 1970.

[22]. S.E. P. Cagigas, "Contribution to word prediction in Spanish and its integration in technical aids for people with physical disabilities," PhD diss., University of Madrid, Madrid, 2001.

[23]. G. Hudson, Why Amharic is not a VSO language. Studies in Africans linguistics. Vol.3, March 1972.

[24]. A. Eilam," Intervention Effects: Why Amharic Patterns Differently," Proceedings of the 27th West Coast Conference on Formal Linguistics, MA: Cascadilla Proceedings Project, pp.141-149, 2008.

[25]. B. McCaul, "Predictive Text Entry In immersive Environment," PhD diss., Dublin City University, Dublin, January, 2005.

[26]. K.Grabski and T. Scheffer, "Sentence Completion," *South. Med. J.*, vol. 48, no. 2, pp. 207, 1955.

[27]. M.Wester, "User Evaluation of a Word Prediction System," M. S. thesis, Uppsala University, Uppsala, 2003.

[28]. R. Baeza-Yates and B. Ribeiro-Neto,*Modern Information Retrieval*, Addison Wesley, 1999.

[29]. T.S.Hasselbring and M.E.Bauch,"Assistive technology for reading," *Education Leadership*, pp. 72-75, 2006.

[30]. A. Bosch, "Scalable classification-based word prediction and confusible correction," *TAL*, vol. 46, pp. 39-63, 2005.

[31]. D. Jurafsky and J.H. Martin, "Speech and Language Processing. PrenticeHal,"2000.

[32]. T. Kroch *et al.*,"The Absence of Intervention Effects in Amharic: Evidence for a Non-Structural Approach Aviad Eilam," pp. 1–77, 2006.

[33]. A. Ibrahim and Y. Assabie, "Hierarchical Amharic Base Phrase Chunking Using HMM With Error Pruning," pp. 328–332, 2010.

[34]. M.D. Dikken *et al.*, "Amharic relatives and possessives : Definiteness, agreement and the linker," pp. 1–30, 2004.

[35]. A. Eilam, "Intervention Effects : Why Amharic Patterns Differently," Proceedings of the 27th West Coast Conference on Formal Linguistics, ed. Natasha Abner and Jason Bishop, pp. 141–149, 2008.

[36]. N. Suleiman, "Word Prediction for Amharic Online Handwriting Recognition," M. S. thesis, University of Addis Ababa, Addis Ababa, 2008.

[37]. A. Fazly, "The Use of Syntax in Word Completion Utilities," M. S. thesis, University of Toronto, Toronto, 2002.

[38]. VanDyke, "A syntactic predictor to enhance communication for disabled users," Tech. Rep., Department of Computer and Information Sciences, University of Delaware, pp. 92-03, 1991.

[39]. A.Carlberger *et al.*, "Probability-based word prediction for writing support in dyslexia," Proceedings of Fonetik97 Conference, vol. 4, 1997.pp.17-20.

[40]. Y. Even-Zohar and D. Roth, "A classification approach to word prediction," In Proceedings of the Conference, 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, ACL, 2000.

[41]. G. Landau and U. Vishkin, "Fast string matching with k differences," Journal of Computer Systems Science, 37:63–78, 1988.

[42]. J.Gubbins and A. Vlachos, "Dependency Language Models for Sentence Completion," In *EMNLP*, pp. 1405-1410. 2013.

[43]. S. Carmelo *et al.*, "A word prediction methodology for automatic sentence completion," In *Semantic Computing (ICSC), 2015 IEEE International Conference on*, pp. 240-243. IEEE, 2015.

[44]. G. Zweig and C.J. Burges, "The Microsoft Research sentence completion challenge," Technical Report MSR-TR-2011-129, Microsoft, 2011.

[45]. I. Tenney, "A general-purpose sentence-level nonsense detector," 2014.

[46]. G. Vitoria *et al.*, "Intelligent word-prediction to enhance text input rate (a syntactic analysis-based word-prediction aid for people with severe motor and speech disability)," In Proceedings of the 2nd international conference on Intelligent user interfaces, pp. 241-244. ACM, 1997.

[47]. W. Lemayehu, "Rule Based Syntactic Disambiguation Parser for Amharic Sentence," M. S. thesis, Addis Ababa University, Addis Ababa, 2005.

[48]. G. Zweig *et al.*, "Computational Approaches to Sentence Completion," 2012.

[49]. G.Tesema, "Design and Implementation of Predictive Text Entry Method for Afaan Oromo on Mobile Phone," PhD diss., Addis Ababa University, 2013.

[50]. J.Cowell and H.Fiaz, "Two template matching approaches to Arabic, Amharic and Latin isolated characters recognition," *Machine Graphics and Vision*, vol. 14, no. 2, p. 213, 2005.

[51]. W. Alemu and F.Siegfried, "Handwritten Amharic bank check recognition using hidden Markov random field," In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, IEEE, vol. 3, pp. 28, 2003.

[52]. N. Suleiman and S. Atnafu , "Word Prediction Model for Amharic Text Input Methods," *Building Software in Developing World: Ethiopian ICT Forum*, Adis Abeba, 2011.

[53]. M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," In *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011.

[54]. "WordSmith Tools, Lexical Analysis Software for Ltd," [online] 2007,http://www.lexically.net/wordsmith/index.html (Accessed:8 September 2015).

[55]. J. J. Li and A. Nenkova, "Fast and Accurate Prediction of Sentence Specificity," In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[56]. S. Bickel *et al.*,"Predicting sentences using n-gram language models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Berlin, 2005.

[57]. D.B. Payne and H.G. Gunhold. "Digital sundials and broadband technology," in Proc. IOOC-ECOC, 1986, pp. 557-998.

[58]. C. Aliprandi, "An inflected-sensitive letter and word prediction system," *International Journal of Computing & Information Sciences* 5, pp. 79-85, 2007.

[59]. S. Hunnicutt, "Using Syntactic and Semantic Information in a Word Prediction Aid," Proc. Europ. COnf. Speech Commun. Paris, France. September 1989, pp. 191-193.

[60]. Dillmann, Ethiopic language Grammar, U.S.A: Wipf & Stock, 2005.

[61]. Z. Mekuria, "Design and Development of Part-of-speech Tagger for Kafi-noonoo Language," M. Sc. thesis, University of Addis Ababa, Addis Ababa, 2013.

[62]. A.V. Aho, "Algorithms for finding patterns in strings," in Handbook of Theoretical Computer Science, vol. A (J. van Leeuwen, ed.), Elsevier, Amsterdam, 1990, pp. 255-300

[63]. Epstein *et al.*, "The Language of African Literature," in *Paper presented at the Annual Conference of African Linguistics, Harvard University*, Cambridge, 1998.

[64]. J. Worne, "Languages for the future," British council , London , 2013.

[65]. T. mindaye *et al.*, "The Need for Amharic WordNet," Adis Abeba.

[66]. C. Beck *et al.*, "Fast user test results with the predictive typing system FASTY," in *Computers Helping People with Special Needs*, paris, Springer Berlin Heidelberg, 2004, p. 813–819.

[67]. J. Carlberger, "Design and implementation of a probabilistic word prediction program," M. S. thesis, University of Nada (Royal Institute of Technology), Sweden, 1997.

[68]. N. Garay-Vitoria and J. Abascal, "Text prediction systems: a survey," *Universal Access in the Information Society* , vol. 4, no. 3, pp. 188-203 , 2006.

[69]. A. Mulu and V. Goyal, "Amharic Text Predict System for Mobile Phone," *International Journal of Computer Science Trends and Technology (IJCST)* , vol. 3, no. 4, pp. 113-118, 2015.

[70]. M. Daud and V. Goyal, "Predictive Text Entry Method for Somali Language on Mobile," *International Journal of Computer Science Trends and Technology (IJCST)* , vol. 3, no. 3, pp. 274-280, 2015.

[71]. W. Alemu, "Application of OCR Technique to the Amharic Script," Adis Abeba,1997.

[72]. Y. Assabie and J. Bigun, "HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation," in *10th International Conference on Document Analysis and Recognition*, Halmstad, 2009.

[73]. H. Alemayehu, ፍቅር እስከ መቃብር/fk'r ɨskə mək'abr/, Adis Abeba: Mega Printers, 1996.

[74]. B. Yimam, yäamarňa säwäsäw, Addis Ababa: E.M.P.D.A., 1986.

[75]. Steven Bird, Ewan Klein, and Edward Loper, Natural Language processing with python, Cambridge: O'Reilly Media, 2009.

[76]. L. Koul, "Methodology of Educational Research," in *Review of the Related Litrature*, Bombay, Vikas Publishing House PVT LTD, 1988, p. 83.

[77]. A. Shimeles "Online Handwriting Recognition for Ethiopic Characters," M. S. thesis, Department of Computer Science, Addis Ababa University, June 2005.

[78]. A. Abebaw, "Implementation of Online Handwriting Recognition System for Ethiopic Character Set", Masters Project, Department of Computer Science, Addis Ababa University, June 2007.

[79]. R. Greenbaum *et al.,* A grammar of contemporary English, London: Longman, 1972.

Appendix A

Corpus sample in IPA

jət'or hajlu nbrətoʧ məkakəl bərusija jətəsəru təwagiˈ bomb t'ajna agʷagʷaʒ awroplanoʧ jgəɲalu꞉ twasənaləʧ꞉ jəangola amakaj jəbahd dar jəajər huneta bəkrəmt sentigredna bəbəga sentigred nəw꞉ angola hulət wək'toʧ alʷat꞉ ɨnəzihm dərək'na znabama naʧəw꞉jəawroplan marəfijawoʧ kənəzihm jəasfalt məndərdərija jalaʧəw angola jəsəfa jəməgʷagʷaza awtar binoratm bəgize maləfna t'ornət mknjat məngədoʧ asfəlagi t'gəɲa altədərəgəbaʧəwm꞉ bəandand botawoʧ aʃkərkariwoʧ mət'fo botawoʧn ləmaləf silu kəməngəd wʧ' jnədalu꞉ ɨndəzih kəmadrəg bəfit gn bəməngəd dar jalu bəməret wst' slətək'əbəru fəndʒiwoʧ jəmijastə'nək'k'u mlktoʧn mastəwal jasfəlgal꞉ jəangola ikonomi tlk' ləwt' asajtʷal꞉ jərub mɨtə ʔamət t'ornət jasadərəbət təs'ɨnon alfo zare bəʔaləm wst' bəft'nət bəmadəg laj kalut ikonomiwoʧ jmədəbal꞉ɨ.e.a. jəʧajna eksimbank ɸ bilijon br ləangola abədrʷal꞉ jh gənzəb ɨndə məngədoʧ jalutn jəangola məsərətawi tək'ʷamoʧ ləmaʃaʃal jəmiwl nəw꞉lalibəla bəitjop'jaˈ bəamara kll bək'ədmow jəwəlo kflə hagər jəmtgəɲ kətəma nəʧ꞉ laj jəmtgəɲəw jəlalibəla kətəma kəbahr tə'ləl bəlaj metr kəfta jalat sthon jəhzbum bzat wədə nəw꞉ lalibəla itjop'ja wst' kalut k'dusan kətəmawoʧ məkakəl kəaksum k'ət'la bəhulətəɲanət dərədʒa jəmtgəɲ kətəma sthonˈ ləabzaɲoʧu jəkrstna ɨmnət təkətajoʧ ɨndə wana jəɨmnət maɨkəl bəməhon tagələglaləʧ꞉ jəlalibəla nəwariwoʧ bəabzaɲaw jəitjop'ja ortodoks təwahdo betəkrstijan ɨmnət təkətajoʧ naʧəw꞉ jəlalibəlan kətəma bəwanəɲanət tawak'i jarəguwat kək.l. bəhʷala məto kfləzəmən ɨndə təsəru jəminəgərlaʧəw abjatə-krstijanat naʧəw꞉ bəitjop'ja twfit məsərət ɨnəzih abjatə-krstijanat bəngus lalibəla zəmən bək'dusan məlaʔɨkt ɨndətəsəru jəmitamən sihon grham hankok jətəbaləw ɨnglizawi s'əhafi gn ɨea ʔa.m basatəməwna bətəbaləw məs'hafu abjatə-krstijanatun bəmanəs'u sra laj templarsjəmibalut jəməsk'əl t'orəɲoʧ təkafləwal sil attʷal꞉ ɨnəziha abjatə-krstijanat wst' arba tnnʃ betəkrstijanoʧ alu꞉ ngusu lalibəla jəmiləwn sm jagəɲəwˈ siwələd bənboʧ slətəkəbəbə nəw jbalal꞉ lal malət mar malət sihonˈ lalibəla malətm -lal jbəlal (mar jbəlal) malət andəhonə jnəgral꞉ wk'r betəkrstijanatun ngusu tə'rbo jəsraʧəw kəmələɨkt ɨgəza gar ɨndəhonə bəitjop'ja ortodoks ɨmnət təkətajoʧ jnəgral꞉ kəflə zəmən awropawi təgʷaʒ lalibəlan təməlkto «jajəhutn bnagr manm ɨndəne kalajə bəfs'um ajamnəɲm» sil tənagro nəbər꞉ bəlalibəla wk'r ʔabjatə krstijanat jalu sihon kənəzihm wst' betə gijorgis (balə məsk'əl k'rs'u) sitaj whalkvn jətə'bək'ə jməslal꞉ betə mədhane ʔaləm jətəbaləw dəgmo kəhulum tlk'u nəw꞉ʔalama bək'ʷami laj ɨndiwləbələb jəmisək'əl ʧə'rk' nəw꞉ bədro gize mərədʒa mələwawət' wanəɲa t'k'mu nəbərə꞉ bəahunu gize gn jəandn hagər wəjm drʤt ləməwəkəl jagələglal꞉ ato kəbədə mikael msale - ɠna bətəbaləw drsətaʧəw laj꞉- səndək' ʔalam jənəs'anət mlkt jəand hzb matəbˈ jəhbrət masərija t'bk'

harəg nəw꞉ təməlkət ʔalamahn꞉ təkətəl aləkˈahn꞉ bləw asfrəwtal꞉'''aksum''' bəsəmen itjopˈja bətgraj k/hagər kəadwa tərarawotʃ atəˈgəb jəmtgəɲ kətəma nətʃ꞉ bəkrstos ldət bəfit təməsrto jənəbərəw jəaksum srwə məngst maɨkəl nəbərətʃ꞉ jəaksum srwə məngst məto kflə zəmən akababi ɨjətədakəmə simətˈa jəmaɨkəlawiw məngst wədə dəbub tənkˈəsakˈəsə꞉ səba amst bəməto jəmihonəw jəkətəmaw nəwari jəortodoks hajmanot təkətaj nəw꞉ jətəkˈərut nəwariwotʃ jəsuni ɨslmna təkətajotʃ natʃəw꞉ balatʃəw tarikawi təfəlaginət kətəma wstˈ jəmigəɲut jəaksum srwə məngst kˈritotʃ jəaləm kˈrs bota təbləw təsəjməwal꞉ aksum bəitjopˈja jə tgraj kll kətəma sihon bəmaʔɨkəlawi zonna bəlaʔɨlaj majtʃəˈw wərəda jgəɲal꞉ maʔɨkəlawi jəstatstiks baləsltˈan ɨndətəmənəw səw mənorija sihon ɨnəsum wəndotʃna setotʃ maʔɨkəlawi jəstatstiks baləsltˈan, bəlela gələltəɲa tmna dəgmo səw mənorija tədərgo təgəmtʷal꞉

and gize ɨnardʒ ɨnawga wərəda fələgə brhan kətəma klinikvn gʷad kˈomtʃˈe ambaw mərkˈəw kəfətu jlna ɨne bəmalastədadrbət bəbure ʃkvdad mərəkˈə blo gazetəˈɲaw təsasto awəra꞉꞉setotʃ꞉ ɨrguzotʃ꞉ dəkamotʃ꞉ ɨjələfu nəw bje kasmamahu bəhʷala wəftʃˈown kəadis abəba jəmijamətˈalɲ atˈahu꞉꞉ kvrat꞉ tbit꞉ gubəɲa꞉ mkˈəɲa꞉ məzbari athuɲ btʃa꞉꞉ hzbu jwədʃal꞉꞉ bəihapa gize jəgətəˈməwot tʃgr nəbər aluʔ awo! wərədawn alnəgrʃm꞉꞉ btʃa bəzija akababi jətəˈnətəˈnə and ʃfta alə꞉ andu bet gəbahu꞉꞉ kəzija nəgəru dəs slalaləɲ wətadər lke lela bet ɨndizəgadʒlɲ adrge wədə lela bet təzawərhu꞉꞉ jəgondər꞉ jəwəlo꞉ jəgodʒam hulu jətmhrt betu jəlmat komite astəbabari adərəguɲ꞉꞉ dʒenəralun baləsltˈanun sajkˈər abəba mətkəja gudgʷad təˈwat təˈwat askˈofrəwaləhu꞉꞉ ɨne tuta ləbʃe wha nəbər jəmatəˈtˈa꞉꞉ ɨnəsu wərkˈatʃəwn lezəratʃəwn ləbsəw jʃəllalu꞉꞉ ato bəgaʃaw atalaj ləɨjandandu kadre hulət hulət kvntal buna ləsnkˈ sətˈtəwatʃəw nəbər꞉꞉ ɨne jatʃw dəmoze nat꞉꞉ ɨna ɨnəsu biratʃəwn jtəˈtˈalu꞉꞉ kˈofru slatʃəw mətʃ misəmuɲ honu dəgmo bətmrtus jəwaza məsəlhuʃ? let təkˈən atəˈna nəbər꞉꞉ tmhrtu ajkəbdm nəbər? ərʷa ajkəbdm! jəmitawəkˈ ajdəl꞉꞉ bətˈam kˈəlal nəbər꞉꞉ jəinperijalistuna jəsoʃalistun huneta nəw꞉꞉ jəamara bherawi kllawi məngst jətˈbkˈna fəkˈad sətəˈɲ꞉꞉ mətʃem lətˈurəta mədəgomija jbəkˈal꞉꞉ godʒamma ɨsum wənd꞉ ɨnem wənd꞉ kəman anʃe nəw təˈbəkˈa mgəzaʔ jlal꞉꞉ lətˈbkˈna sra ʃəga adis abəba nəw꞉꞉ jəɲa səw nəgər awakˈi nəɲ blo təˈbəkˈa makˈom zkˈtəɲanət jməsləwal꞉꞉ ɨna ɨndalhuʃ ɨrkˈ ssəra nəw jəmwləw꞉꞉ bəakvriw bahlatʃn məsərət snastarkˈ nəw jəmwl꞉꞉ ahun bəɨdr bəmahbərawim honə bətələjajə jəmahbərat astəbabarina məri nəɲ꞉꞉ bəihadegs altəʃələmum? səmtʃaləhu srawotn꞉꞉ tˈru nəw꞉꞉ səwn astəmrut꞉꞉ ʃmagle nəwot꞉꞉ tlkˈ səw nəwot꞉꞉ jləfalu꞉꞉ ajzowot.. aluɲ꞉꞉ ɨndihu andəze jədə/markˈos junivərsti simərəkˈ nggr ɨndadərg təgabʒe tənagərkv꞉꞉ ..bher bherəsəbotʃ jəmitəwawəkˈubət koledʒ təsəralnə꞉꞉ jəsəratʃhuln məmarija bet jəɲa nəw꞉꞉ bʷanbʷaw məskotu ɨndajsəbər ɨntəˈbkˈalən꞉꞉ kəlela agər jəmətˈa dəbal sˈəbaj kalə ɨɲa godʒamotʃ anfəlgm꞉ gurorown ankˈən ləfrd ɨnakˈərbalən.. alɦtʃəw꞉꞉ bəitjopˈja bəahunu wəkˈt slaləw jəpolətika huneta mn jlaluʔ itjopˈja wstˈ jəmajabara tˈornət nəbər꞉꞉ jane prezidantun ..sltˈan agaru꞉ tədəradəru.. bjatʃəw nəbər gn ..drdr jələm.. bləw ɨmbɲ alu prezidantu꞉꞉ jəmigərmʃ bədərg wəkˈt tˈoru ɨndajwaga jəmikˈəsəkˈs muzikˈa nəbər꞉꞉ ..ahun

jəɨne mənorˑ mənoru nəw wəjˑ gutʃ gutʃ jalə t'ut and k'ən salaj.. jəmil k'sk'əsa wət'atu jsəma nəbər:: t'or meda? ɨhsa:: bət'or meda k'sk'əsa sidərəgbət wət'atu bək'a ɨzija mək'ojət ajfəlgm:: bək'a ɨjətəwə məmt'at ʤəmərə:: ʃnfətu ajələ:: lelaw t'ornətu dəgmo jəand agər t'ornət nəbərə:: jaw ahun bəslt'an laj jaləw məngst aʃənfʷal:: nəgər gn janze jənəbərəw t'ornət tarikawi təblo ajjazm:: hulət wəndmamatʃotʃ nəbəru jətək'aməsu:: jəɨrs bərs t'ornət sləhonə:: tarikawi t'ornət jəmtjw kəertraˑ kəsumaleˑ kət'aljanˑ turkotʃˑ gbs'otʃˑ ɨnglizotʃ gar jənəbərəw t'ornət malət nəw:: and t'ornət tarikawi jəmibaləw t'ornət bəmijawk'u sajntistotʃˑ jət'or tə'bəbtotʃ sigəməgəm nəw:: ahun jaləw ləsələm lədimokrasiˑ ləməlkam astədadər jək'omə astədadər nəw bibalm kəlaj jəwət'aw məmərija tkkl hono salə kətatʃ gn jʃərarəfal:: kətatʃ jajəʃ ɨndəhon məmərijawotʃˑ hgə məngstuˑ lelotʃ nəgərotʃ ɨjətəʃərarəfu jgəɲalu:: tək'ot'at'ro ləmastəkakəl t'bk' kttl jasfəlgal:: bəihadeg zəmən bəmrʧ'a ləmn altəwədadərum? aj!məngstat bəslt'anatʃəw jəmimət'abatʃəw ajwədum:: ɨnem dəmo kəɨngdih jəagər ʃmagle nəɲ:: təmərtə'ʃ parlama bəmtgəbibət gize bijans jəməʤəmərija digri linorʃ jgəbal:: lənəgəru lətəsatfom məmar t'ru nəw:: gn tmhrtun ləmn ɨskə digri algəfubətm? ɨh ɨngdija! ɨne bətəlʔɨko bəhg jət'bk'na diploma jʒaləhu:: tə'/mi mələs zenawin agɲtəwatʃəw jawk'alu? ərə jələm:: tə'/ministrun jəmaməsəgnatʃəw jəitjop'jana jəertran t'ornət bəaʃənafinət bəməwət'atatʃəw nəw:: jək'ədmo prezidant məngstu bəmijastədadru gize jəsomale t'or ..awaʃ nəw dnbəre.. blo mət'to nəbərə:: ləwəre nəgari sajk'ər ɨndə agəda kmr tək'atə'lʷa! tə'/mi mələsm bihonu t'ornət bajfəlgum t'orəɲotʃn ləməkəlakəl jəmijadərgut jəmidənk' nəw:: jhm dəgmo as'i tewodrosˑ as'i johans hjwətatʃəwn jəsəwt nbrət tə'ftʷatʃəw nbrət fləga sajhon hagər ləmətə'bək' nəw:: tadija səw ləməfəwəs bəbərəhaw ɨjəhedə bəwha t'm ɨskə səwotʃ motəwal:: bətʃ'aka gəbtəw məngədu tə'ftʷatʃəw:: ɨna tə'/ministru ɨgziabher kəzija hulu ʃmk' wgijaˑ ahunm bihon andand kəftəɲa məsənakloʧ agat'mʷatʃəw jətəwət'ut bəɨgziabher hajl sləhonə . . . jəɨnəzihn talalak' səwotʃ agər ..abaj flwhan.. balətʃwot ak'm bəɨgziabher bjəwaləhu jasəruln:: ..abaj flwha.. məngədu bisəra kəftəɲa jəɨmnəbərəd kmʧtˑ kəftəɲa jəbrət mrtˑ jək'batna jəsəlit' ɨndihumˑ jəbərbəre mrt bəbzat jaləbət nəw:: kədə/eljas abaj fl wha ki.me rk'ət nəw jalə:: bəitjop'ja jətəgəɲu jət'bəb səwotʃ bəsrawotʃatʃəw jətunm jahl məlәkija bik'əmət'latʃəw ləgə/krstos gn jh k'ərəw ləmalət mat'afijaw jat'r jhonal:: bəhasabuna bəsmetu təfətləw bəget'ut srawotʃu jətədənək'nəwn jakl kəastəwajnətuˑ kəsbɨnawna kəmannətu ɨndəzihum kətə'nkara səratəɲnətu jəmnmaratʃəw bzu k'umnəgərotʃ alu:: bətəwələn srawotʃu k'ən k'ənn ɨjəwələdəw sik'ət'l jəzih səw srawotʃ wrs honəw ləhul gizem abrəwn jnoralu:: gəbrə krstosn bəsʔɨl sraw ʔaləm awk'otal:: jh btʃa ajdələm ləsmetum bihonˑ bəmat'atna bəmagɲət ˑ bəmot ʧ'kane ˑ bəbtʃəɲnətˑ bətkazena fk'r bəatə'k'alaj bəsəw lʤ tə'baj tələjtəw jəminəsu mannətotʃn nək'so…bəgt'mo jətəwəln srawotʃu zəmən təʃagari natʃəw:: jəgəbrəkrstos dəstan nəgər basbəw sasbəw ɨndijaw dərso andatʃ jəmalfatəw hasab bəwst'e jrawət'al:: məʃto ɨskinəga ləne jəzih səw sra adis nəw:: bəza zəmən jətənəsu asdənak'iwotʃu bɨrəɲotʃ bəaləm

təwədadəri məhonatʃəwn smməskr kvrat aləɲ። nəgərgn bək't'u ləməzəkər jəbək'ut ɨʤg t'k'itoʧu məhonatʃəwn srəda adəra bəla məhonatʃn jtajəɲal። jhe asdənak'i səw ləagəru jənəbərəwn fk'r itjop'ja ɨnate sil bəsət'at bota jəagərun nafk'ot agəre ɨndəgəna bətəsəɲu gt'moʧu ɨndəzihum jəagər k'uʧ'tun jəs'om k'ən ɨna bəbaɨd agər bətəsəɲut gt'moʧu bəmigəba gəls'otal። bətələj agəre jətəsəɲəw gt'mu anʤət jbəlal። mn jdərəg jh talak' səw bəsdət ɨndəmasənə jəagərun afər ləmək'məs aləmətadəlu jask'oʧ'al። gəbrə krtos kəabatu kəalək'a dəsta nəgəwo kəɨnatu kəwə/ro as'ədə marjam wəndmagəɲ ʔa.m bəmsrak'u jəhagəraʧn kfl harər kətəma təwələdə። kəzər jəmiwərəs andaʧ nəgər kalə abatu alək'a dəsta bəitjop'ja betəkrstijan hajmanotawi tmhrt lik'na bəbrana jəʔɨʤ s'hufoʧaʧəw bəbahlawiw sʔɨloʧaʧəw jətədənək'u səw ɨndə nəbəru jnəgəral። gəbrə krstos wəlaʤ ɨnatun (wə/ro as'ədə marjam wəndmagəɲ) gəna bəlʤnətu bəmot tənətə'k'ə። mətʃəm jəɨnat mot kəbad nəw። bʧa amlak ʧə'rso alkəfam ajatu ɨmahoj brk'nəʃ sasahu lk ɨndə ɨnat masrəʃa…lk ɨndənat honəw asadəgut ። jəabatun ɨwk'ət blom jəsʔɨl sra ɨjadənək'ə ladəgəw tnʃu gə/krstos jəhjwət t'riwn jələjəw gəna bətə'watu nəbər። bətələj jəmidənk'əw ɨzihga nəw! ɨko mn? kalaʧhu gəna jəsdst amət lʤ ɨjalə ”ɨnatnət” blo jəsəjəmatn jəɨrsas sʔɨl ajən። gəna bzu jəbzu bzu jasajənal። gəbrə krstos ləabatu jənəbərəw fk'r mətʃəm jətələjə nəw። jətunnm jahl adnak'ot ”ɨrəft adrg ahun” bəmiləw gt'mu gəls'otal። anʤət jəmibəla gt'm nəw! gəbrə krstos kə hed sidni gar badərəgəw aʧ'r k'alə mətə'jk' slə abatu jəmikətəwn blo nəbər፡- abəte k'dusan məs'ahftn bəɨʤu js'f nəbər። bəzəmənu zəmənawi jəhtmət məsarija bəhagəraʧn jənəbərə bihonm bənorəw bahl məsərət bəɨʤ jətəzəgəʤu məs'ahftun təmərəʧ' nəbəru። ləməs'əhaftu zgʤt jəmijasfəlgəwn məs'afijam honə k'ələmun jazəgaʤ jənəbərəwna ɨrasu sihon bək'dusan məs'ahftu wst' jəmikatətu mslotʃn jsəra nəbər። bəlʤnət gizeje jəabaten srawoʧ ɨjajəw ɨdənək' nəbər።

Appendex B

Amharic language without tranlslated

የጦር ኃይሉ ንብረቶች መካከል በሩሲያ የተሰሩ ተዋጊ፣ ቦምብ ጣይና አጓጓዥ አውሮፕላኖች ይገኛሉ።ትዋሰናለች። የአንጎላ አማካይ የባህድ ዳር የአየር ሁኔታ በከረምት ሴንቲግሬድና በበጋ ሴንቲግሬድ ነው። አንጎላ ሁለት ወቅቶች አሏት። እነዚህም ደረቅና ዝናባማ ናቸው። የአውሮፕላን ማረፊያዎች ከነዚህም የአስፋልት መንደርደሪያ ያላቸው አንጎላ የሰፋ የመጓጓዛ አውታር ቢኖራትም በጊዜ ማለፍና ጦርነት ምክንያት መንገዶች አስፈላጊ ጥገኛ አልተደረገባቸውም። በአንዳንድ ቦታዎች አሸከርካሪዎች መጥፎ ቦታዎችን ለማለፍ ሲሉ ከመንገድ ውጭ ይነዳሉ። እንደዚህ ከማድረግ በፊት ግን በመንገድ ዳር ያሉ በመሬት ውስጥ ስለተቀበሩ ፈንጂዎች የሚያስጠነቅቁ ምልክቶችን ማስተዋል ያስፈልጋል። የአንጎላ ኢኮኖሚ ትልቅ ለውጥ አሳይቷል። የሩብ ምእተ ዓመት ጦርነት ያሳደረበት ተጽእኖን አልፎ ዛሬ በዓለም ውስጥ በፍጥነት በማደግ ላይ ካሉት ኢኮኖሚዎች ይመደባል።እ.ኤ.አ. የቻይና ኤክሲምባንክ ፪ ቢሊዮን ብር ለአንጎላ አበድሯል። ይህ ገንዘብ እንደ መንገዶች ያሉትን የአንጎላ መሠረታዊ ተቋሞች ለማሻሻል የሚውል ነው።ላሊበላ በኢትዮጵያ፣ በአማራ ክልል በቀድሞው የወሎ ክፍለ ሃገር የምትገኝ ከተማ ነች። ላይ የምትገኘው የላሊበላ ከተማ ከባህር ጠለል በላይ ሜትር ከፍታ ያላት ስትሆን የህዝቡም ብዛት ወደ ነው። ላሊበላ ኢትዮጵያ ውስጥ ካሉት ቅዱሳን ከተማዎች መካከል ከአክሱም ቀጥላ በሁለተኛነት ደረጃ የምትገኝ ከተማ ስትሆን፣ ለአብዛኞቹ የክርስትና እምነት ተከታዮች እንደ ዋና የእምነት ማእከል በመሆን ታገለግላለች። የላሊበላ ነዋሪዎች በአብዛኛው

የኢትዮጵያ ኦርቶዶክስ ተዋህዶ ቤተክርስቲያን እምነት ተከታዮች ናቸው። የላሊበላን ከተማ በዋነኛነት ታዋቂ ያረጉዋት ከክ.ል. በኋላ መቶ ከፍለዘመን እንደ ተሰሩ የሚነገርላቸው አብያተ-ክርስቲያናት ናቸው። በኢትዮጵያ ትውፊት መሰረት እነዚህ አብያተ-ክርስቲያናት በንጉሥ ላሊበላ ዘመን በቅዱሳን መላዕክት እንደተሰሩ የሚታመን ሲሆን ግርሃም ሃንኮክ የተባለው እንግሊዛዊ ፀሃፊ ግን እኤአ ዓ.ም ባሳተመውና በተባለው መጽሃፉ አብያተ-ክርስቲያናቱን በማነፁ ሥራ ላይ ቴምፕላርስየሚባሉት የመስቀል ጦረኞች ተካፍለዋል ሲል አትቷል። እነዚህ አብያተ-ክርስቲያናት ውስጥ አርባ ትንንሽ ቤተክርስቲያኖች አሉ። ንጉሡ ላሊበላ የሚለውን ስም ያገኘው፣ ሲወለድ በንቦች ስለተከበበ ነው ይባላል። ላል ማለት ማር ማለት ሲሆን፤ ላሊበላ ማለትም -ላል ይበላል (ማር ይበላል) ማለት አንደሆነ ይነግራል። ውቅር ቤተክርስቲያናቱን ንጉሡ ጠርቦ የስራቸው ከመላእክት እገዛ ጋር እንደሆነ በኢትዮጵያ ኦርቶዶክስ እምነት ተከታዮች ይነግራል። ከፍለ ዘመን አውሮፓዊ ተጓዥ ላሊበላን ተመልክቶ «ያየሁትን ብናግር ማንም እንደኔ ካላየ በፍጹም አያምነኝም» ሲል ተናግሮ ነበር። በላሊበላ ውቅር ዐብያተ ክርስቲያናት ያሉ ሲሆን ከነዚህም ውስጥ ቤተ ጊዮርጊስ (ባለ መስቀል ቅርፀ) ሲታይ ውሃልኩን የጠበቀ ይመስላል። ቤተ መድሃኔ ዓለም የተባለው ደግሞ ከሁሉም ትልቁ ነው።ዓላማ በቋሚ ላይ እንዲውለበለብ የሚሰቀል ጨርቅ ነው። በድሮ ጊዜ መረጃ መለዋወጥ ዋነኛ ጥቅሙ ነበረ። በአሁኑ ጊዜ ግን የአንድን ሀገር ወይም ድርጅት ለመወከል ያገለግላል። አቶ ከበደ ሚካኤል ምሳሌ - ፩ኛ በተባለው ድርሰታቸው ላይ፡- ሰንደቅ ዓላም የነጻነት ምልክት የአንድ ሕዝብ ማተብ፤ የጎበረት ማሰሪያ ጥብቅ ሐረግ ነው። ተመልከት ዓላማህን፣ ተከተል አለቃህን። ብለው አስፍረውታል።"'አክሱም"' በሰሜን ኢትዮጵያ በትግራይ ክ/ሃገር ከአድዋ ተራራዎች አጠገብ የምትገኝ ከተማ ነች። በክርስቶስ ልደት በፊት ተመስርቶ የነበረው የአክሱም ስርወ መንግስት ማእከል ነበረች። የአክሱም ስርወ መንግስት መቶ ክፍለ ዘመን አካባቢ እየተዳከመ ሲመጣ የማእከላዊው መንግስት ወደ ደቡብ ተንቀሳቀሰ። ሰባ አምስት በመቶ የሚሆነው የከተማው ነዋሪ የአርቶዶክስ ሃይማኖት ተከታይ ነው። የተቀሩት ነዋሪዎች የሱኒ እስልምና ተከታዮች ናቸው። ባላቸው ታሪካዊ ተፈላጊነት ከተማ ውስጥ የሚገኙት የአክሱም ስርወ መንግስት ቅሪቶች የአለም ቅርስ ቦታ ተብለው ተሰይመዋል። አክሱም በኢትዮጵያ የ ትግራይ ክልል ከተማ ሲሆን በማዕከላዊ ዞንና በላዕላይ ማይጨው ወረዳ ይገኛል። ማዕከላዊ የስታትስቲክስ ባለስልጣን እንደተመነው ሰው መኖሪያ ሲሆን እነሱም ወንዶችና ሴቶች ማዕከላዊ የስታትስቲክስ ባለስልጣን, በሌላ ገለልተኛ ትምና ደግሞ ሰው መኖሪያ ተደርጎ ተገምቷል።

አንድ ጊዜ እናርጅ እናውጋ ወረዳ ፈለገ ብርሃን ከተማ ከሊኒኩን ጓድ ቆምጬ አምባው መርቀው ከፈቱ ይልና እኔ በማላስተዳድርበት በቡሬ ሽኩዳድ መረቀ ብሎ ጋዜጠኛው ተሳስቶ አወራ::ሴቶች፣ እርጉዞች፣ ደካሞች፣ እየለፉ ነው ብዬ ካስማማሁ በኋላ ወፍጮውን ከአዲስ አበባ የሚያመጣልኝ አጣሁ:: ኩራት፣ ትቢት፣ ጉበኛ፣ ምቀኛ፣ መዝባሪ አትሁኚ ብቻ:: ህዝቡ ይወድሻል:: በኢህአፓ ጊዜ የገጠመዎት ችግር ነበር አሉ? አዎ! ወረዳውን አልነግርሽም:: ብቻ በዚያ አካባቢ የጠነጠነ አንድ ሽፍታ አለ፤ አንዱ ቤት ገባሁ:: ከዚያ ነገሩ ደስ ስላላለኝ ወታደር ልኬ ሌላ ቤት እንዲዘጋጅልኝ አድርጌ ወደ ሌላ ቤት ተዛወርሁ:: የጎንደር፣ የወሎ፣ የጎጃም ሁሉ የትምህርት ቤቱ የልማት ኮሚቴ አስተባባሪ አደረጉኝ:: ጀነራሉን ባለሥልጣኑን ሳይቀር አበባ መትከያ ጉድጓድ ጠዋት ጠዋት አስቆፍረዋለሁ:: እኔ ቱታ ለብሼ ውሃ ነበር የማጠጣ:: እነሱ ወርቃቸውን ሌዘራቸውን ለብሰው ይሸልላሉ:: አቶ በጋሻው አታላይ ለእያንዳንዱ ካድሬ ሁለት ሁለት ኩንታል ቡና ለስንቅ ሰጥተዋቸው ነበር:: እኔ ያችው ደሞዜ ናት:: እና እነሱ ቢራቸውን ይጠጣሉ:: ቆፍሩ ስላቸው መች ሚሰሙኝ ሆኑ ደግሞ በትምርቱስ የዋዛ መሰልሁሽ? ሌት ተቀን አጠና ነበር:: ትምህርቱ አይከብድም ነበር? ኧሯ አይከብድም! የሚታወቅ አይደል:: በጣም ቀላል ነበር:: የኢንፔሪያሊስቱና የሶሻሊስቱን ሁኔታ ነው:: የአማራ ብሔራዊ ክልላዊ መንግሥት የጥብቅና ፈቃድ ሰጠኝ:: መቼም ለጡረታ መደጐሚያ ይበቃል:: ጎጃምማ እሱም ወንድ፣ እኔም ወንድ፣ ከማን አንሼ ነው ጠበቃ ምዝ? ይላል:: ለጥብቅና ሥራ ሽጋ አዲስ አበባ ነው:: የኛ ሰው ነገር አዋቂ ነኝ ብሎ ጠበቃ ማቆም ዝቅተኛነት ይመስለዋል:: እና እንዳልሁሽ እርቅ ሥሠራ ነው የምውለው:: በአኩሪው ባህላችን መሠረት ስናስታርቅ ነው የምውል:: አሁን በእድር በማህበራዊም ሆነ በተለያየ የማህበራት አስተባባሪና መሪ ነኝ:: በኢህአዴግስ አልተሸለሙም? ሰምቻለሁ ሥራዎትን:: ጥሩ ነው:: ሰውን አስተምሩት:: ሽማግሌ ነዎት:: ትልቅ ሰው ነዎት:: ይለፋሉ:: አይዞዎት.. አሉኝ:: እንዲሁ አንደዜ የደ/ማርቆስ ዩኒቨርስቲ ሲመረቅ ንግግር እንዳደርግ ተጋብዤ ተናገርኩ:: ..ብሔር ብሔረሰቦች የሚተዋወቁበት ኮሌጅ ተሰራልነ:: የሠራችሁልን መማሪያ ቤት የኛ ነው:: ቧንቧው መስኮቱ እንዳይሰበር እንጠብቃለን:: ከሌላ አገር የመጣ ደባል ፀባይ ካለ እኛ ጎጃሞች አንፈልግም፤ ጉሮሮውን አንቀን ለፍርድ እናቀርባለን.. አልኳቸው:: በኢትዮጵያ በአሁኑ ወቅት ስላለው የፖለቲካ ሁኔታ ምን ይላሉ? ኢትዮጵያ ውስጥ የማያባራ ጦርነት ነበር:: ያኔ ፕሬዚዳንቱን ..ሥልጣን አጋሩ፤ ተደራደሩ.. ብያቸው ነበር ግን

..ድርድር የለም.. ብለው እምብኝ አሉ ፕሬዚዳንቱ፡፡ የሚገርምሽ በደርግ ወቅት ጦሩ እንዳይዋጋ የሚቀሰቅስ ሙዚቃ ነበር፡፡ ..አሁን የእኔ መኖር፤ መኖሩ ነው ወይ፤ ጉች ጉች ያለ ጡት አንድ ቀን ሳላይ.. የሚል ቅስቀሳ ወጣቱ ይሰማ ነበር፡፡ ጦር ሜዳ? እህሳ፡፡ በጦር ሜዳ ቅስቀሳ ሲደረግበት ወጣቱ በቃ እዚያ መቆየት አይፈልግም፡፡ በቃ እየተወ መምጣት ጀመረ፡፡ ሽንፈቱ አየለ፡፡ ሌላው ጦርነቱ ደግሞ የአንድ አገር ጦርነት ነበረ፡፡ ያው አሁን በሥልጣን ላይ ያለው መንግሥት አሸንፏል፡፡ ነገር ግን ያንዜ የነበረው ጦርነት ታሪካዊ ተብሎ አይያዝም፡፡ ሁለት ወንድማማቾች ነበሩ የተቃመሱ፡፡ የእርስ በርስ ጦርነት ስለሆነ፡፡ ታሪካዊ ጦርነት የምትይው ከኤርትራ፣ ከሱማሌ፣ ከጣልያን፣ ቱርኮች፣ ግብፆች፣ እንግሊዞች ጋር የነበረው ጦርነት ማለት ነው፡፡ አንድ ጦርነት ታሪካዊ የሚባለው ጦርነት በሚያውቁ ሳይንቲስቶች፣ የጦር ጠበብቶች ሲገመገም ነው፡፡ አሁን ያለው ለሰላም ለዲሞክራሲ፣ ለመልካም አስተዳደር የቆመ አስተዳደር ነው ቢባልም ከላይ የወጣው መመሪያ ትክክል ሆኖ ሳለ ከታች ግን ይሸራረፋል፡፡ ከታች ያየሽ እንደሆን መመሪያዎች፣ ህገ መንግሥቱ፣ ሌሎች ነገሮች እየተሸራረፉ ይገኛሉ፡፡ ተቆጣጥሮ ለማስተካከል ጥብቅ ክትትል ያስፈልጋል፡፡ በኢህአዴግ ዘመን በምርጫ ለምን አልተወዳደሩም? አይ!መንግሥታት በሥልጣናቸው የሚመጣባቸው አይወዱም፡፡ እኔም ደሞ ከእንግዲህ የአገር ሽማግሌ ነኝ፡፡ ተመርጠሽ ፓርላማ በምትገቢበት ጊዜ ቢያንስ የመጀመሪያ ዲግሪ ሊኖርሽ ይገባል፡፡ ለነገሩ ለተሳትፎም መማር ጥሩ ነው፡፡ ግን ትምህርቱን ለምን እስከ ዲግሪ አልገፉበትም? እህ እንግዲያ! እኔ በተልዕኮ በህግ የጥብቅና ዲፕሎማ ይዣለሁ፡፡ ጠ/ሚ መለስ ዜናዊን አግኝተዋቸው ያውቃሉ? ኧረ የለም፡ ፡ ጠ/ሚኒስትሩን የማመሰግናቸው የኢትዮጵያና የኤርትራን ጦርነት በአሸናፊነት በመወጣታቸው ነው፡፡ የቀድሞ ፕሬዚዳንት መንግሥቱ በሚያስተዳድሩ ጊዜ የሶማሌ ጦር ..አዋሽ ነው ድንበሬ.. ብሎ መጥቶ ነበረ፡፡ ለወሬ ነጋሪ ሳይቀር እንደ አገዳ ከምር ተቃጠሏ! ጠ/ሚ መለስም ቢሆኑ ጦርነት ባይፈልጉም ጦረኞችን ለመከላከል የሚያደርጉት የሚደንቅ ነው፡፡ ይህም ደግሞ አፄ ቴዎድሮስ፣ አፄ ዮሐንስ ህይወታቸውን የሰውት ንብረት ጠፍቷቸው ንብረት ፍለጋ ሳይሆን ሀገር ለመጠበቅ ነው፡፡ ታዲያ ሰው ለመፈወስ በበረሃው እየሄደ በውሃ ጥም እስከ ሰዎች ሞተዋል፡፡ በጫካ ገብተው መንገዱ ጠፍቷቸው፡፡ እና ጠ/ሚኒስትሩ እግዚአብሔር ከዚያ ሁሉ ሽምቅ ውጊያ፤ አሁንም ቢሆን አንዳንድ ከፍተኛ መሰናክሎች አጋጥሟቸው የተወጡት በእግዚአብሔር ሃይል ስለሆነ . . . የእነዚህን ታላላቅ ሰዎች አገር ..አባይ ፍልውሃን.. ባለችዎት አቅም በእግዚአብሔር ብየዋለሁ ያሰሩልን፡፡ ..አባይ ፍልውሃ.. መንገዱ ቢሠራ ከፍተኛ የእምነበረድ ክምችት፣ ከፍተኛ የብረት ምርት፤ የቅባትና የሰሊጥ እንዲሁም፣ የበርበሬ ምርት በብዛት ያለበት ነው፡፡ ከደ/ኤልያስ አባይ ፍል ውሃ ኪ.ሜ ርቀት ነው ያለ፡፡ በኢትዮጵያ የተገኙ የጥበብ ሰዎች በሥራዎቻቸው የቱንም ያህል መለኪያ ቢቀመጥላቸው ለገ/ክርስቶስ ግን ይህ ቀረው ለማለት ማጣፊያው ያጥር ይሆናል፡፡ በሀሳቡና በስሜቱ ተፈትለው በጌጡት ስራዎቹ የተደነቅነውን ያክል ከአስተዋይነቱ፤ ከሥብእናውና ከማንነቱ እንደዚሁም ከጠንካራ ሠራተኝነቱ የምንማራቸው ብዙ ቁምነገሮች አሉ፡፡ በተወለን ሥራዎቹ ቀን ቀንን እየወለደው ሲቀጥል የዚህ ሠው ሥራዎች ውርስ ሆነው ለሁል ጊዜም አብረውን ይኖራሉ፡፡ ገብረ ክርስቶስን በስዕል ሥራው ዓለም አውቆታል፡፡ ይህ ብቻ አይደለም ለስሜቱም ቢሆን፤ በማጣትና በማግኘት ፣ በሞት ጭካኔ ፣ በብቸኝነት፣ በትካዜና ፍቅር በአጠቃላይ በሠው ልጅ ጠባይ ተለይተው የሚነሱ ማንነቶችን ነቅሶ...በግጥም የተወልን ሥራዎቹ ዘመን ተሻጋሪ ናቸው፡፡ የገብረክርስቶስ ደስታን ነገር ባስበው ሳስበው እንዲያው ደርሶ አንዳች የማልፋተው ሀሳብ በውስጤ ይራወጣል፡፡ መሸቶ እስኪነጋ ለኔ የዚህ ሰው ሥራ አዲስ ነው፡፡ በዛ ዘመን የተነሱ አስደናቂዎቹ ብእረኞች በአለም ተወዳደሪ መሆናቸውን ስመስክር ኩራት አለኝ፡፡ ነገርግን በቅጡ ለመዘከር የበቁት እጅግ ጥቂቶቹ መሆናቸውን ስረዳ አደራ በላ መሆናችን ይታየኛል፡፡ ይሄ አስደናቂ ሰው ለአገሩ የነበረውን ፍቅር ኢትዮጵያ እናቴ ሲል በሰጣት ቦታ የአገሩን ናፍቆት አገሬ እንደገና በተሰኙ ግጥሞቹ እንደዚሁም የአገር ቁጭቱን የጾም ቀን እና በባእድ አገር በተሰኙት ግጥሞቹ በሚገባ ገልጾታል፡፡ በተለይ አገሬ የተሰኘው ግጥሙ አንጀት ይበላል፡፡ ምን ይደረግ ይህ ታላቅ ሰው በስደት እንደማሰነ የአገሩን አፈር ለመቅመስ አለመታደሉ ያስቆጫል፡፡ ገብረ ክርቶስ ከአባቱ ከአለቃ ደስታ ነገዎ ከእናቱ ከወ/ሮ አፀደ ማርያም ወንድማገኝ ዓ.ም በምስራቁ የሀገራችን ክፍል ሀረር ከተማ ተወለደ፡ ፡ ከዘር የሚወረስ አንዳች ነገር ካለ አባቱ አለቃ ደስታ በኢትዮጵያ ቤተክርስቲያን ሀይማኖታዊ ትምህርት ሊቅና በብራና የዕጅ ጽሁፎቻቸው በባህላዊው ሥዕሎቻቸው የተደነቁ ሰው እንደ ነበሩ ይነገራል፡፡ ገብረ ክርስቶስ ወላጅ እናቱን (ወ/ሮ አፀደ ማርያም ወንድማገኝ) ገና በልጅነቱ በሞት ተነጠቀ፡፡ መቸም የእናት ሞት ከባድ ነው፡፡ ብቻ አምላክ ጨርሶ አልከፋም አያቱ እማሆይ ብርቅነሽ ሳሳሁ ልከ እንደ እናት ማስረሻ...ልከ እንደናት ሆነው አሳደጉት ፡፡ የአባቱን እውቀት ብሎም የሥዕል ሥራ እያደነቀ ላደገው ትንሹ ገ/ክርስቶስ የሕይወት ጥሪውን የለየው ገና በጠዋቱ ነበር፡፡ በተለይ የሚደንቀው እዚህጋ ነው! እኮ ምን? ካላችሁ ገና የስድስት አመት ልጅ

እያለ "እናትነት" ብሎ የሥየማትን የእርሳስ ሥዕል አየን፡፡ ገና ብዙ የብዙ ብዙ ያሳየናል፡፡ ገብረ ክርስቶስ ለአባቱ የነበረው ፍቅር መቸም የተለየ ነው፡፡ የቱንንም ያህል አድናቆት "እረፍት አድርግ አሁን" በሚለው ግጥሙ ገልጾታል፡፡ አንጀት የሚበላ ግጥም ነው! ገብረ ክርስቶስ ከ ሄድ ሲድኒ ጋር ባደረገው አጭር ቃለ መጠይቅ ስለ አባቱ የሚከተውን ብሎ ነበር፡- አበቴ ቅዱሳን መጻህፍትን በእጁ ይጽፍ ነበር፡፡ በዘመኑ ዘመናዊ የህትመት መሳሪያ በሀገራችን የነበረ ቢሆንም በኖረው ባህል መሠረት በእጅ የተዘገጁ መጻህፍቱን ተመራጭ ነበሩ፡፡ ለመጻሀፍቱ ዝግጅት የሚያስፈልገውን መጻፊያም ሆነ ቀለሙን ያዘጋጅ የነበረውና እራሱ ሲሆን በቅዱሳን መጻህፍቱ ውሥጥ የሚካተቱ ምስሎችን ይሠራ ነበር፡፡ በልጅነት ጊዜዬ የአባቴን ሥራዎች እያየው እደነቅ ነበር፡፡

APENDEX C

Sample of the prototype code

```
from tkinter import *

from operator import itemgetter, attrgetter

#import l3

import codecs

import re

import stringEditll

import DynamicProgramming

from decimal import *

from tkinter import END,Listbox, Tk


def gu():

    def NewFile():

        print ("New File!")

    def OpenFile():

        name = askopenfilename()

        print( name)

    def About():
```

```
    print( "This is a sample of prototype menu")

root=Tk()


menu = Menu(root)

root.config(menu=menu)

filemenu = Menu(menu)

menu.add_cascade(label="File", menu=filemenu)

filemenu.add_command(label="New", command=NewFile)

filemenu.add_command(label="Open...", command=OpenFile)

filemenu.add_separator()

filemenu.add_command(label="Exit", command=root.quit)


helpmenu = Menu(menu)

editmenu = Menu(menu)

viewmenu = Menu(menu)


menu.add_cascade(label="Edit", menu=editmenu)

menu.add_cascade(label="View", menu=viewmenu)

menu.add_cascade(label="Comment", menu=editmenu)

menu.add_cascade(label="ReView", menu=viewmenu)

menu.add_cascade(label="Help", menu=helpmenu)

helpmenu.add_command(label="About...", command=About)

root.title("**Design a prototype /አማርኛ/ for Amharic Lanaguage version 1.0**")

root.geometry("500x350")
```

```
 var = StringVar()
 label = Label( root, textvariable=var, bg="black",fg="whiteSmoke",font=('Comic Sans MS', 16) )
var.set("ሙሉ ዓረፍተ-ነገር ለማግኘት የዓረፍተ-ነገሩን የተወሰነ ክፍል ያስገቡ!!!")
 label.pack()
 text = Text(root, bg="white",fg="blue", height=6,width=45,font=15)
 #text = Text(root, bg="whiteSmoke",fg="blue", height=6,width=45,font=15)
 text.pack()
# scroll=Scrollbar(root,command=text.yview)
 text.tag_add("here", "1.0", "1.4")


 def helloCallBack():
     import codecs
     varr=text.get(1.0, END)
     import obo
     import stringEdit11
     import codecs
     from operator import itemgetter, attrgetter
     dicf=[]
     f=codecs.open("C:/Python34/thesis/Documentation/Training Set.txt","r","UTF-8")
     f=f.read()
     #print("file name:",f)
     #a=f.split(" ")
    a=[text.strip() for text in re.split('[!?።፤፣]', f) if text]
     y=varr
```

```
#print("splitted:",f)

c=0

prdct={}

dic=[]

#while(c<len(a)):

for aa in a:

    yy=stringEdit11.Distance(aa,y)

    xx=PosP.posTag(aa,y)

    zz=TF-IDF.tf-idfFunc(aa,yy)

    H=0.16*yy + 0.24*xx +0.6*zz ()

   dic.append(H[1])

    #print(yy[1])

    prdct[a[c]]=H[1]

    #print(a[c])

    #print("result:",y,prdct,yy[1])

    #print(y,stringEdit.simDistance(aa,y),aa)

    c=c+1

dic.sort()

print("++++++++++=",dic)

# reverse increasing order

view=reversed(sorted(dic))

for lll in view:

    print("wer:",lll)

print("results 2:",y,prdct,aa,yy)
```

```
ll=(sorted(prdct.items(), key=itemgetter(1), reverse=True))

print(ll)

rr={}

dic=[]

ik=0

mm={}

for pp in ll:

    print("popop",pp)

    sdt=pp[0]

    dic.append(sdt)

    #text.insert(INSERT,pp)

    #print("ssss",sdt)

    mm[ik]=sdt

    ik=ik+1

    if ik==5:

        break

    #print("waa",ll)#print(aa)


#text.delete('1.0', END)

#text.insert(INSERT,mm)

#print(mm)

for item4 in dic:

    #listbox.insert(END,item4)

    #print("real:",item4)
```

```
            # text.delete('1.0', END)
             print("itemmmm",item4)
             #text.insert(INSERT,item4)
             #print("single4",uy)
             #print("dic",vd)

          If H[1]>0.025
             text.insert(INSERT,item4)
         # rr=obo.getNGrams(f, 5)
          #for www in rr:
          #print("ngrams",www)
        #print(stringEdit.simDistance("የአንበሳዉን ድርሻ የያዘዉ ግብርና ነዉ"," ድርሻ "))
        #print(stringEdit.simDistance("የአንበሳዉን","የአንበሳዉን ድርሻ የያዘዉ ግብርና ነዉ"))
    from tkinter import ttk
    #root = tkinter.Tk()
    style = ttk.Style()
    style.map("C.TButton",foreground=[('pressed', 'red'), ('active', 'blue')],background=[('pressed', '!disabled', 'black'),
('active', 'white')])
 colored_btn = ttk.Button(text="Click On This" ,style="C.TButton",command = helloCallBack).pack()
    #B.pack()
root.mainloop()
```

Declaration

This thesis work is the original work of the researcher. The ideas taken from other works have either paraphrased or quoted directly and acknowledged in the reference section properly.

Mohammed Nuru _____

(Name) Signature

_____ _____

(Pricipal Advisor) Signature of confirmation

Place: Jimma Institute of Technology, Jimma University

Date of Submission: April 2016