ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES

SCHOOL OF INFORMATION SCIENCE

EFFECT OF MORPHOLOGICAL INFORMATION IN AFAAN OROMO WORD SEQUENCE PREDICTION

A THESIS SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCE OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION SCIENCE

By: WAKSHUM TEMESGEN GURMESSA

JUNE,2017
ADDIS ABABA

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

EFFECT OF MORPHOLOGICAL INFORMATION IN AFAAN OROMOO WORD SEQUENCE PREDICTION

By: WAKSHUM TEMESGEN GURMESSA
Advisor: WONDWOSSEN MULUGETA (PhD)

Name and Signature of Members of Examining Board

Name		Signature
1. Wondwossen Mulugeta (PhD)	Advisor	
2.		
3.		

DECLARATION

I, the undersigned, declare that this thesis is my original work; it has not been submitted and presented elsewhere for any other degree or professional qualification, and that all the sources used for the thesis have been duly acknowledged.

_____

WAKSHUM TEMESGEN GURMESSA

June 2017

I, the undersigned, would like to confirm that the thesis has been submitted for examination with my approval as the university advisor.

WONDWOSSEN MULUGETA (PhD)_____ _____

DEDICATION

I dedicate this work to my beloved parents:

My Father, ***Temesgen Gurmessa***

and

My Dear Mother, ***Astede Gelawu***

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Wondwossen Mulugeta (PhD), without his constructive comments and inspiring suggestions this research couldn't have been completed. I'm very grateful for his genuine advice from the advent to the end of the study.

Secondly, my heartfelt thanks go to Mr. Dandi Merga, Mr. Ashenafi Bekele and Mr. Mekuria Zewude for their professional assistance during the course of my research project. They deserve brotherly appreciation as a token of my gratitude.

Thirdly, I am grateful to my families-my parents, Temesgen Gurmessa & Atsede Gelaw, my sisters, Lense & Jale, my brothers-Aboma, Milko & Ebo, who have rendered me all their care, love and encouragement. Without their sustainable support, I couldn't have reached at this level of my achievement.

Finally, my special thanks also go to my friends-, Fira, Ashe, Sanyi, Free, Gutu, Mase, Abdi, Dagi, Abiyu, Tarroo, Jawwe, Bonsa, Dinqa, Solo, Gadaa, Tasfu, Afe, Benji, Shambe, Abela, Joe, DND, Ase, Abit,Tade, moti, Abdi and all the others form NPD and SPD who encouraged me by sparing their time, effort and resources during the course of my study.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AAC	Augmentative and Alternative Communication
ASCII	American Standard Coding for Information Interchange
CV	Consonant-Vowel
HMM	Hidden Markov Model
HR	Hit Rate
ICR	Intelligent Character Recognition
IR	Information Retrieval
KE	Effective Number of Keystroke s
KSS	Keystroke Saving
KT	Total Number of Keystroke s
KUC	Keystroke Until Completion
NLP	Natural Language Processing
OCR	Optical Character Recognition
POS	Parts-of- Speech
SMS	Short Message Service
SOV	Subject-Object-Verb
SVM	Support Vector Machine
SVO	Subject-Verb-Object
T9	Text on 9 keys
TC	Text Categorization
WP	Word Prediction
WTS	Word Type Saving

ABSTRACT

The purpose of conducting this study is to design Afaan Oromoo word sequence prediction model to explore the effect of morphological information on word sequence prediction. Word prediction is a natural language processing problem that attempt to predict the correct and most appropriate word in a given context; it utilizes language modeling application to guess the next word based on the context in which it has been previously used in a text. Even though, Afaan oromo is used by a large number of populations, no noteworthy work is done on the topic of word sequence prediction. Thus, in this study, word sequence prediction model for Afaan oromo was developed using statistical methods and morphological features. The researcher presented a model that predicts the most likely word based on statistical and morphological information of previous words. N-gram method was employed to construct a Bigram and Trigram language model from stem forms sequence. In addition, morphological properties of Afaan Oromoo verbs and nouns have been extracted using Hornmorph morphological analyzer to develop language model from stem form with morphological features such as tense, case, number, gender and person. Accordingly, the model was set out to suggest the next word to be typed by a user in three phases. Firstly, the most probable stem forms are predicted using language model. Secondly, morphological features are predicted for the proposed stem forms. Lastly, the proposed root or stem word and morphological features are used by morphological synthesizer to generate appropriate surface words. To evaluate the performance of the word sequence model and to demonstrate how morphological features determine the accuracy of word prediction models, the developed model was compared with a model that was developed without considering the morphological features. Accordingly, an experiment had been conducted based on Keystroke saving, and the result of the experiment indicated the better KSS is achieved with the model constructed from N-gram and morphological information. Based on the result of this study, specific research direction is recommended.

CHAPTER ONE: INTRODUCTION

1.1 Background

In today's information society, massive amount of data is stored and processed by computers and other electronic devices to assist various operations in the daily activities of humans. It has been more than half a century since computer technology is engaged in supporting activities like word processing, web browsing, e-mail, blogging, and so on. All these applications require a text entry method. Text document is entered into computer system through peripherals such as standard keyboard, touch screen, trackball, mouse and other devices. Keyboard is a dominant text entry method due to its ease of implementation, higher speed and less error rate [1][2]. Computer keyboard layout has been popularly used since typewriter was invented with QWERTY layout in late 19^{th} century. This layout was invented to solve the mechanical jamming of the key and to provide high entry rate with low error rate via the assumption that the probability that two keys next to each other in an alphabetical order would be hit in that order is minimal [3].

In the milestone of keyboard development, numerous keyboard layout has been developed to increase number of characters entry per minute by rearranging the place of each key. In this regard, QWERTY is the most commonly used for Latin scripts. Replacing Q and W by A and Z, AZERTY layout is being used in France. In this layout A and Z replace Q and W of the QWERTY layout. Similarly, in Germany, Y is replaced by Z and gives the QWERTZ layout. Furthermore, Half – QWERTY keyboard is used to type with only one hand [2] [1].

On the other hand, Trackball with on-screen keyboard is another alternative text entry method mostly used by person with little motor capacity [4]. A trackball is like a mouse that has been turned upside-down, with a mechanical ball that rolls in place. Although a plethora of studies show that for able-body users, trackballs are slower and less accurate, for person with motor disability, it is the most common solution to use virtual keyboard and assistive devices. Although, trackball with on-screen keyboards are easy to learn, they have many drawbacks. For instance, they are visually fatiguing, equivalent to typing in a "hunt-and-peck" fashion [4] [5]. Touch screen with the use of on-screen keyboard is text entry method that monopolize mobile and tablet text entry,

via detecting the pressure on contact area [6]. This is an image of a keyboard layout rendered on a touch-sensitive screen. The typical on-screen keyboard uses the familiar QWERTY layout.

So far, numerous research works have been conducted to improve text entry method, especially to enable easy use for people with disability. According to [4], on average around 23 words per minute can be entered via standard keyboard. They added that secretaries using 10 fingers have a speed of 25 wpm. Nevertheless, for people subjects with disabilities, who use an assistive device, text input speed was only 5 wpm [4]. One of the methods to assist people with motor limit is by using assistive devices that reduce motor cost [16]. Although assistive devices render computers accessible to people with disabilities, the actual inputting of text can be very slow [4].

Simulation studies have shown that optimizing the layout of the keyboard can increase text input speed by 36–55 percent compared with a QWERTY keyboard [4]. Although many improvements are made by rearranging keyboard layout to reduce the distance between letters, they are limited to support speedy typing, spelling error detection and assisting people with disability.

AI has opened unique possibilities to create new text entry or improve upon existing ones [*3*]. AI is an interdisciplinary field of study computer speech and language processing, natural language processing, computational linguistic and other disciplines with the goal to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication or simply doing useful processing of text or speech [7].

According to Shannon human language are highly redundant. These redundancies can be captured in language models. The goal of language modeling is to capture and exploit the restriction imposed on the way in which word can be combined to form sentences. It describes how words are arranged in natural language [7]. Language modeling has many applications in natural language processing problems, such as automatic speech recognition, statistical machine translation, text summarization, and character and handwriting recognition. Word prediction, which is a natural language processing problem that attempt to predict the correct and most appropriate word in given context, utilizes language modeling application to guess the next word given in previous words [8].

The milestone of word prediction takes us back to the end of Second World War when the number of people with disability was increased dramatically. Consequently, to help them to communicate with outside world, assistance technology such as AAC Augmentative and Alternative Commination system was developed [18] [17]. Early 1980's word predictions techniques have been established as a method in the development of AAC systems. Although, word prediction application emerged with AAC. Later on, it has also been adopted in writing assistance systems with the goal of increasing keystroke save percentage, automatic spelling corrector and grammar checker.

There are various prediction systems which have been developed by employing with different approaches for different languages. Masood and Sadaeed [18] summarize the three major word prediction approaches as statistical modeling, knowledge-based modeling and heuristics modeling [18]. In statistical modeling, the choice of words is based on the probability that a string may appear in a text. The statistical information and its distribution could be used for predicting letters, words and phrases. Jurafsky and martin [7] states statistical word prediction as estimating the next word can be computed from the probability of previous sequence of few words, this approach is called N-gram; it is based on markov assumption that the probability of a word is calculated on basic of the last few words and the history is approximated by the last few words [7]. Most of existing word prediction system employs this statistical approach using n- gram with POS tags [18] [8].

Knowledge - based prediction systems that merely use statistical modeling for prediction often present words that are syntactically, semantically, or pragmatically inappropriate and impose a heavy cognition load on users to choose the intended word in addition to the decrease in writing rate. Syntactic, semantic and pragmatic linguistic knowledge can be used in prediction systems [9].

Heuristic (adaptation) method is used to make more appropriate predictions for a specific user and it is based on short term and long-term learning. In short term learning, the system adapts to a user on current text that is going to be typed by an individual user. Recency promotion, topic guidance, trigger and target, and n-gram cache are the methods that a system could use to adapt to a user in a single text [18] [9].

Word prediction research for Ethiopian languages, as far as the literature search of the researcher is concerned there is a dearth of research on word prediction, particularly on Afaan Oromo language. As a response to this short coming, in this study, we present Afaan oromo word sequence model that predict the most probable word based on statistical information and morphological information of previous words.

1.2 Statement of the problem

The purpose of conducting this research is to make Afaan Oromo users beneficiaries of computer technology that will assist Afaan Oromo text entry. Afaan Oromo is Cushitic language which is a family of Afro Asiatic languages. Afaan Oromo is being spoken by more than 30 Million peoples in Ethiopia, though the majority of native speakers live in Ethiopia, they also live in Kenya, Somalia and Egypt. As the Ethiopia 's statistical report of 2007 [13] shows there are more than 25 million speakers of Afaan Oromo in Ethiopia and this fact shows that, the language has the largest speaker followed by Amharic language. It is third largest language in Africa following Kiswahili and Hausa; 4th largest language, if Arabic is counted as Africa language [11] [12].

So far, many researches were conducted to integrate and to make Afan Oromo language beneficiary of the technology. In those studies, an attempt was made on Automatic sentence parser [14], part of speech tagging [13], morphology based spell checker [21] and rule based Afaan oromo grammar checker [11] are researches conducted on the area of NLP and Afaan Oromoo. The overall goal of this studies is to enable computer to perform useful task involving Afaan Oromoo language.

NLP has potential benefit for simplifying the interaction and communication difficulties between human-machine or human-human. Word prediction is a NLP problem that attempt to guess the next word given in previous words. The competence of predicting the most probable word has wide variety of application in IR, Speech recognition, handwrite recognition, communication aid and other systems. In addition, word prediction techniques have also explored a new method on existing text entry method. Word prediction provide engine for Auto complete and writing aid systems.

Word prediction has a wide application in the development of AAC systems to assist people with disability, to minimize physical movement required to produce a text and to reduce cognitive load of correct spelling [15]. Word prediction also serves as writing aid systems to speedy text entry

and to support non-native language learners by suggesting the correct word and auto spelling corrector [18] [19].

Word prediction researches conducted so far in Ethiopia are specific for certain language such for Amharic and Afaan Oromo. Nevertheless, as far as the knowledge of the present researcher is concerned, in Ethiopia studies on word prediction are scant. Only three research were conducted to date for Amharic and Afaan Oromo. For instance, Nesredin Suleiman [2] conducted research on Amharic word prediction for online handwriting recognition using by bi- gram model. In addition, Tigist Tensou [9] conducted a research to develop Amharic word prediction model using statistical method and linguistic rules. However, it is only Gudisa Tesema [10] who made the first attempt to design and develop Afaan Oromo word prediction on mobile phone. Gudisa employ machine learning algorithm. In his study, he identified word prediction as a classification task. By using SVM bag of word would be created for word with the same class. Predictor model was constructed using HMM for the prediction purpose. The major focus of his work was to classify word in some class without considering morphological information. In other words, feature selection does not include grammatical features such as, tense, number, gender, person, aspect and case information. However, in Afan Oromo most of the grammatical information such as tense, aspect, voice, case, gender, number and person are conveyed through affixes attached to the roots or stems [11]. Consequently, both Afan Oromo nouns and adjectives are highly inflected for number, gender and person [12]. For instance, in contrast to the English plural marker s (- es), there are more than 12 major and very common plural markers in Afan Oromo nouns (example: -oota, -ooli, -wwan, - lee, -an, -een, -oo, etc.) [13]. On the other hand, Afaan Oromo verbs are also highly inflected for gender, person, number and tenses. As a result, the prediction model provides word that is grammatically incorrect in a given sequence.

For example, *Jarri dhufaa jira*, subject and verb disagree in number. Jarri (*they) which is the subject of the sentence is* plural, and the verb of the sentence *jira* is the indicator for third person singular masculine [11].

In fact, word prediction is facing a very ambitious challenge, as several typical complex problems arises when dealing with Natural Language. Similarly, word prediction also inherent amounts of arising ambiguities (lexical, structural and semantic ambiguities but also pragmatic, cultural and phonetic ambiguities for speech) are complex problems to be solved by a computer.

Word prediction for inflected languages pose a harder challenge to prediction algorithms. Since, inflected languages have a large dictionary of word forms with several morphological features, produced from a root or lemma and a set of inflection rules. Accordingly, the large number of word forms makes word prediction for inflected languages a hard task. Thus, Afaan oromo shares the challenges of inflected language. Afaan Oromo is an agglutinative and morphologically rich language; each root word can combine with multiple morphemes to generate huge number of word forms. Accordingly, obtaining all vocabulary that language consist would be difficult since corpus for language modeling is not expected to include all words of specific language. For instance, in Afaan Oromo for a single verb root word ***—beek-*** ‖ _go‘; over 800 valid word forms can be formed Thus, Afaan oromo word formation (morphological process) poses other challenge on Afaan oromo word prediction model that required to be solved such as out-coverage dictionary and grammatical agreement between sequence of words.

Despite the aforementioned facts about Afaan Oromo language, the model developed by Gudisa is found to have different gaps that need to be considered in Afan Oromo word sequence prediction. The model fails to consider morphological information, as a result, it is impractical to capture all word forms due to the language's rich morphology. In order to, bridge this gap a sound research has to be conducted. Accordingly, the current researcher is motivated to conduct a study to fill the gaps that haven't been considered in the previous studies.

Research Questions

Based on the statement problem given above, this study attempted to answers for the following basic research questions.

- Does cascading language model with morphological information will improve the performance word sequence prediction model? keystroke saving?
- What is the performance of the word sequence prediction model in keystroke saving?

1.3 Objective of the Study

This study has both general and specific objectives.

1.3.1 General Objective

The general objective of this study is to explore the effect of morphological information in Afaan Oromoo word sequence prediction.

1.3.2 Specific Objectives

The specific objectives of the study are to:

- Review various approaches on word sequence prediction
- Collect document for training and testing model
- Construct a tagged training corpus with stem forms, aspect, case, tense, voice, number, gender and person.
- Construct language models for root or stem forms sequences and root or stem forms sequences with morphological features such tense, case, number, gender and person.
- To evaluate the performance of developed word sequence prediction model.

1.4 Scope and limitation of the study

This study was undertaken to construct Afaan Oromo word sequence prediction model based on statistical frequency of word sequence and morphological information to explore the effect morphological information in Afaan Oromoo word sequence prediction. The corpus for constructing language model is collected from Internet using web crawling technique. The prediction model is built to predict probably the correct word that respect (obey) syntactically rule of Afaan Oromo by cascading the language model with linguistic knowledge extracted by morphological analyzer. The linguist information captured by morphological analyzer is used to build a tagged training corpus. The morphological analyzer and generator used in this study is Hornmorph. However, this research did not deal with errors either in the training corpus or the output of Hornmorph program while building the language model. Consequently, to keep the consistence of word sequence the words which is not processed by morphological analyzer is taken as it is. Similarly, study did not cover automatic grammar relation finder and checking to remove grammatical unacceptable proposed word in prediction list.

1.5 Methodology

For the successful completion of this study, the following methods were used.

Literature review

Developing word sequence prediction model for Afaan Oromoo needs exhaustive understanding of the language. The principles and rules of the language in the area of morphology and word prediction approaches have been carefully studied. A various of, resources including books, research reports, journal articles, manuals and other published and unpublished documents including those from the Internet related to this study were reviewed and synthesized.

Document collection

A corpus for this study was collected from web documents from four websites using web scraping technique. Additionally, 89 PDF files around 54 MB of size were collected from Internet via Google search engine and converted to text file using PYPDF python library. Totally, 60.6MB text collected from https://www.afanoromo.fanabc.com, www.voaafaanoromoo.com, www.gadaa.com, https://chilot.me/regional-laws/oromia-nrs-laws/ and Google search result were stored on text file format. Thus, the corpus is composed of three genre such as news, blogs and academic document. After preprocessing and transformation the size has reduced into 12MB. Out of this, training set containing 84,952 sentences which is equivalent to 11.8MB has used to develop word sequence prediction model.

Development Tools

Hornmorph morphological analyzer and generator program was used to build tagged training corpus and to produce surface words. Similarly, to analyze user input words morphological analyzer was used. Hornmorph is the only morphological analyzer and synthesizer tool available freely for Afaan Oromoo. Moreover, Python programming language was used to develop prototype for demonstration. Python is selected because of it has a natural language toolkit module that provides a predefined function for the implementation of statistical and linguistic modeling.

Prototype Development

To develop prototype, supporting tools are required. Hence, python programming language and Hornmorph morphological analyzer and generator program will be used. As previously stated, Hornmorph will be used to morphologically analyze collected training corpus. It will also be used to morphologically analyze user entered texts from the testing data, so that required features like gender, number, and person will be captured and used to generate proposed words in correct grammatical form. Python programming language will be used to implement statistical language models. As part of the prototype development a user interface will be designed that allows users to type their text and choose from the list of suggested words.

Evaluation

In order to, demonstrate and evaluate the developed model a Keystroke Saving (KSS) is used. This means, the prediction competence is evaluated through calculation of keystroke savings. A Keystroke Saving (KSS) estimates saved effort percentage and is calculated through comparison of total number of keystrokes needed to type a text (KT) and effective number of keystrokes using word prediction (KE) [22].

$$KSS = \frac{KE - KT}{KT} * 100 - - - - - - - - - - - - - (1.1)$$

Therefore, the number of keystrokes to type texts taken from the test data with and without word sequence prediction program was counted to calculate keystroke savings accordingly. The obtained KSS will be compared for tri-gram, bi-gram and hybrid models. The model that shows maximum keystroke saving is considered as better model.

1.6 Application of Results

The beneficiaries of this study include researchers who are, or want to be, involved in increasing the capability of computer processing in Afaan Oromoo. Specially, this study benefits researchers who devoted on Afaan oromo predictive text entry project. This is because the result of the study provides them concrete concept on aspects to be considered to improve keystroke saving and reduce cognitive load of writing skill.

1.7 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 discusses literature review on different issues in text entry methods and word prediction approaches. Specifically, in this chapter, text entry method, history of word prediction approaches and methods are discussed. Chapter 3 is devoted to discussing related works done on word prediction developed in different languages. Chapter 4 specifies the morphological and phonological properties specific to Afaan Oromoo. Thus, many language specific issues such as the writing system, syllable structures, inflections and derivations have been extensively presented. Chapter 5 discusses Afaan oromo word sequence prediction model. This chapter presents Architecture of the proposed Word Sequence Prediction Model and its components with their respective algorithms. The implementation and evaluation issues are presented in chapter 6. Chapter 7 concludes the thesis by outlining the benefits obtained from the research work and limitations of the system. It also shows some research directions and recommendations that can be accomplished in developing a full-fledged a predictive text entry system for Afaan Oromo.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This Chapter discusses fundamental concepts of word sequence prediction and concepts related with text entry methods. An overview of the existing text entry methods and details of the word prediction techniques employed to open new opportunity over existing method discussed in the chapter. Since the main target of this study is to design and develop word sequence prediction model for Afan Oromo language, the structure of the language is considered besides the statistical distribution of words. Thus, the methods and techniques used for dealing with and modeling morphological characteristics, grammatical properties and parts-of-speech of the language identified and discussed. In order to support selection of tools, methods and techniques with justification, word prediction model development approaches in line with their performance and language structure will be compared in the literature review.

2.2 Text Entry Method

One of the most predominant and necessary techniques used as an interface between human and machine is data entry technique that is implemented using input device. This technique is helpful to enter different kinds of data such as text, voice, image and movie to the machine in order to get them processed. There are a number of data entry techniques and methods; these include: speech recognition, keyboards, handwriting recognition, scanner, microphone and digital camera.

Text entry method is the abstract description of how to accomplish text data entry. A text entry system is a concrete implementation of a text entry method. Keyboard is a common text input technique in devices like desktops, laptops and other hand-held devices [1]. Keyboards are pure selection interfaces. The user is presented with a matrix of keys which he or she can select sequentially to produce the text. There are two kinds of keyboards: virtual and standard keyboards**.** Virtual keyboard or on-screen keyboard with touchscreen is the most commonly used text input method on mobile phones.

The design for desktop computer's keyboards has been inherited from the typewriter character layout called QWERTY. This layout was invented to solve the mechanical jamming of the key and to provide high entry rate with low error rate via the assumption that the probability of two neighbor characters arranged in alphabetical order pressed correspondingly to form a word is very low. The QWERTY layout actually makes pretty good use of the human hands. While one finger is pressing a key, others can prepare for their work by moving over the following keys. It is fast and error free for many practical purposes. Although QWERTY layout is claimed as fast and error free, experiments show that keyboard layout is determined by writing system of a given language. Consequently, QWERTY is the most commonly used layout for Latin scripts. However, AZERTY layout is used for France scripts. In this layout A and Z replace Q and W of the QWERTY layout. Similarly, in Germany scripts Y is replaced by Z and gives the QWERTZ layout as, for instance, Amharic language requires combination of two or more keys to represent one letter in a language [1][2].

Similar to the structure of language, the degree of motor ability of user also determines the layout and the structure of keyboard. People with motor disabilities frequently experience difficulties using a standard keyboard. In response to this factor, Half – QWERTY keyboard was developed to support the individuals typing with only one hand [1] [4].

Many assistive technology and text entry method was developed with the aim of improving comfort of user, reducing time taken and supporting people with disability. Now day's a desktop computer without an actual keyboard have also been constructed; it is operated by sensing the finger movements by some other means such as cameras or pressure sensors, or the typing can occur without any visual guide. Trackball with on-screen keyboard, Touch Screen with the use of on-screen keyboard and other methods are also developed as alternative text entry [4][6].

Although many improvements are made by rearranging keyboard layout to reduce the distance between letters, they are limited to support speedy typing, spelling error detection and assisting people with disability. For instance, one of the methods to assist people with motor limit is by using assistive devices that reduce motor cost [16]. Although these assistive devices render computers accessible to people with disabilities, the actual inputting of text can be very slow.

In general, the text entry methods described above are called *manually text entry methods*. Manually text entry method does not include the language related issues of syntax, and neither is

the issue of semantics in the text entry. This means the method does not assist a way to deal with spelling, grammar and semantic errors of the sentences. Accordingly, the users are expected to have proper skills of the language; in addition, the users need to possess the experience of speedy writing [1].

2.3 Writing Aid

Word processors have several capabilities that may influence the writing process. A word processing application assists beyond basic processing input captured for the text entry devices including: spelling checkers, speech synthesis, word prediction, grammar, and style checkers. They can support the basic skills of producing legible texts with correct mechanics [14] [15]. In this regard, Charles A. MacArthur has undertaken a research entitled, "Using Technology to Enhance the Writing Processes of Students with Learning Disabilities". The researcher reviewed and evaluated the ways that computers can support writing for the students with learning disabilities, with an emphasis on applications that go beyond word processing. He discussed basic components of word processing applications which could enhance students' writing. These basic components are spelling checker, speech synthesis, grammar checker, word prediction, and style checkers. The experiment was conducted on twenty-six motor limited students from middle school who were randomly selected and instructed to write stories and revise their spelling using a spelling checker. The findings of the experiment indicated that the students were able to correct 82% of the errors with correct suggestions and 18% of the errors when the correct suggestion was not offered [14].

Another writing aid used in word processing is Speech Synthesis Software. The speech synthesis software translates text into speech. It is not as natural-sounding as digitized speech, which is recorded, but its advantage is that it can be used to utter any text. Word processors with speech synthesis enable students to hear what they have written and to read what others have written. This technique may support writing by allowing users with writing problems to use their general language sense to monitor the adequacy of their writing. Word prediction and a bank of words provide vocabulary to enhance the use and appropriate selection of words. Word prediction is common technique for assisting physically disabled individuals by reducing the number of keystrokes required to type words and sentences [14] [16].

2.4 History of Word Prediction

The inception of the concept of word prediction takes us back to the end of the Second World War when the number of people with disabilities was increased dramatically. In order to help them to communicate with outside world, assistance technologies such as Augmentative and Alternative Commination systems were developed. The field of Augmentative and Alternative Communication (AAC) is concerned with mitigating communication barriers that would isolate individuals from society. Basically, one way to improve communication rate is to decrease the number of keys entered to form a message, and the goal of saving keystroke requires estimating the next letter, word or phrasing that likely follow a given segment of text. As a result, early 1980's word predictions techniques have been established as a method in the development of AAC systems [17] [18]. Since 1980s, many systems with different methods were developed for different languages. According to Shannon, human languages are highly redundant, and these redundancies can be captured in language models. To this end, the goal of language modeling is to capture and exploit the restriction imposed on the way in which word can be combined to form sentences. It describes how words are arranged in natural language. Word predictions are also applied in language modeling application to guess the next word given in previous words [7] [8] [18].

2.5 Terminology of word prediction

Prediction refers to those systems that guess which letters, words, or phrases are likely to follow in a given segment of a text. The systems typically operate by displaying a list of the most likely letters, words, or phrases for the current position of the sentence being typed by the user. As the user continues to enter letters of the required word, the system displays a list of the most probable words that could appear in that position. Then, the system updates the list according to the sequence of the so-far entered letters. Next, a list of the most common words or phrases that could come after the selected word would appear. The process continues until the text is completed. The notion of prediction can be seen in three phrases. These are: letter prediction, word prediction, and sentence prediction.

Letter prediction is a text entry technique commonly used into recent technologies such as cell-phones and PDAs. Letter prediction could be used as an aiding tool to enter a text on Short Message Service (SMS), to chat on Instant Message, and to write an email. Most of these devices could not have a single key for a letter. So, a text should be entered with a limited number of keys. For

instance, in cell-phones, a text is written with only 9 keys on the phone. This means that a key should carry three or four letters. The reduced keyboard makes it hard for the user to enter a text; so, the letter prediction method would be an efficient way. The reason to have such a system is that the user will need to press only one key for each character on the mobile phone. As a result, prediction techniques are used to disambiguate three of four letters in a single key. It considers the letters probability to disambiguate letters on one key. Since disambiguating of letters are based on the already entered characters and not on the lexical dictionary in itself, as a result the method needs a small amount of memory and it is much easier to enter new words [10] [18].

Word prediction guess the words that the user intends to use. These words are suggested in a list to the user that might be used in that position. If the required word is not available among options offered in prediction list, a user may continue writing. Differently from letter predictors, word predictors typically make use of language modeling techniques, namely stochastic models and linguistic knowledge that are able to give context information in order to improve the prediction quality. Most of the literature related to word prediction concerns non-inflected languages. Language Models and prediction techniques are presented that allow the user to save more than 50% of keystrokes. The language that the system has to model influences the prediction techniques; inflected languages pose a harder challenge to prediction algorithms, since they have to deal with a usually high number of inflected forms that dramatically decrease Keystroke Saving [10] [9].

Sentence prediction estimates a sequence of words to complete a sentence given initial fragment text. It provides user a list of possible sequence of words to complete a sentence. Given an initial text fragment, a predictor that solves the sentence completion problem has to estimate the entire remaining words based on the sentence that the user frequently constructs. As a result, it supports writing task where user is engaged in writing the sentence which is same with the sentence they construct before. It assists writing in applications with repetitive tasks such as writing emails in call centers or letters in an administrative environment [24] [23]. Information retrieval techniques is a common method used for developing sentence prediction that involves finding, in a corpus, the sentence which is most similar to a given initial fragment. Information retrieval aims to provide methods that satisfy a user's information needs. Here, the model has to retrieve the remaining part of a sentence. Research approach is to search for the sentence whose initial words are most similar to the given initial sequence in vector space representation. The similarity between two vectors is

defined by the cosine measure. The drawback of sentence prediction methods is the amount of text needed to train the model. Training corpus has to be large enough [23]. As the current study strives to conduct a study on word prediction for Afaan oromo the different approach of word prediction is discussed succinctly.

2.6 Approaches for word prediction

According to Ghayoomi and Momtazi [18] word prediction method that used for modeling nature language since 1980s would general classified into three approaches. These three major approaches are described: statistical modeling, knowledge-based modeling, and heuristic modeling (adaptive) [9] [18].

2.6.1 Statistical approach

Traditionally, predicting words has solely been based on statistical modeling of the language. In statistical modeling, the choice of words is based on the probability that a string may appear in a text. Consequently, a natural language could be considered as a stochastic system. Such a modeling is also named probabilistic modeling. The statistical information and its distribution could be used for predicting letters, words, phrases, and sentences. Several prediction systems use some form of statistical analysis, examining the likelihood of certain words being used in a sentence. Word frequency and word sequence frequency are commonly used methods in statistical prediction. Most current statistical word prediction is made based on Markov assumption in which only last n-1 word of the history affects succeeding word and it is named n-gram Markov model. Thus, it is based on learning parameters from large corpora. There are several approaches and methods to statistical prediction and each of these is discussed in this section.

Word frequency

The early predictive systems specially the ones which were used as a writing aid tool in the 1980s, used merely the frequency information of each word independently to complete a word in the current position of a sentence being typed by the user without considering the previous context (the history). In other words, the systems used unigram word model with a fixed lexicon. Such systems always come up with the same prediction suggestions for a particular word. Since solely independent statistical information has been used in this model, most of the time the suggestions might be inappropriate [25].

Word frequency method applied in various word prediction approaches. For instance, one simpler approach is fixed lexicon, this approach sorts the complete lexicon into their frequency order, and offers the few at the top of the list to the user as predictions. If the prediction that is required does not appear on the list straight away then the user types in the first letter. The list is then reduced to only those words, which have the initial letter just entered, and again, the top few are offered as predictions. This process continues until either the word has been spelt out in its entirety, or it has appeared on the prediction list, at which point the user can add it to the sentence in whichever way is made available to him by the device [10].

Other approach is an adaptive lexicon, which alters the frequency tags attached to the words contained in the dictionary as the user constructs sentences. These provide an indication of how recently a word has been used as this will increase the likelihood of it being used again. In a similar manner to the fixed lexicon, the adaptive lexicon can provide statistics of words independently of each other, paying no attention to preceding words. The only difference is that each time a word is used; its frequency and recency will be updated and stored in the dictionary for later reference. If a new word is encountered during a session, then it is added to the lexicon with a frequency of one but a high recency, so that when it is used again, it will appear acceptably high up the prediction list [10] [9].

Word sequence frequency

Using the frequency of one single word or unigram word model for early systems, it was clear that some of the suggestions are not appropriate in that position of a sentence; suggestions will be better if context or history of words is taken into account. This means word sequence history would provide a clue for appearance of the next words. Nevertheless, in most case, it is difficult to calculate the probability of entire sequence. Based on Markov assumption in which only last n-1 word of the history affects succeeding word. Two common statistical models those provide a compatible technique to compute probabilities of next words though Markov model are N-gram and HMM [25] [24].

N-gram Language models

N-gram language models are the most well-known type of language models in speech and language processing. They are important tool in NLP tasks for those requires identifying noisy ambiguous words input. Thus, N-gram language models have dominated the speech recognition area for years due to their simplicity, efficiency and robustness [26]. On other hand, in Handwriting recognition it plays vital role for estimating confusable and unreadable letters and words [2] [7]. In general, besides these sample areas, *N*-grams are also crucial in NLP foundations like art-of-speech tagging, natural language generation, and word similarity, as well as applications from authorship identification and sentiment extraction to predictive text input systems for cell phones. N-gram approximate the probability of a word sequence as a product of conditional probabilities of the current word *wi* given a history of the preceding $n-1$ words. The order of an *n*-gram model refers to the value *n*, where n is the number of words used in the probability sequences.

$P\,(W = w1, w2\ldots, wn) = \prod_{i=1}^{n} P(wi|wi-n+1,\ldots,wi-1)$,..... (2.1)

According to Maximum Likelihood Estimation the n-gram probability is the relative frequencies of $wi-n+1,wi$ to $wi-n+1,..,wi-1$ in the training data.

$$P(wi|wi-n+1,\ldots,wi-1) = \frac{C(wi-n+1,\ldots,wi-1,wi)}{C(wi-n+1,\ldots,wi-1)} \text{ ----------(2.2)}$$

where $C(wi-n+1,\ldots,wi-1,wi)$ is the frequency count of the string $wi-n+1,\ldots,wi-1,wi$ in the training data.

Bigram and Trigram

Bigram and trigram are the most successful form of N-gram method in that predictor engine considers only the probabilities of a given previous one or two words in the sentence. A lexicon, which stores the probabilities of word-pairs, is known as a bigram. One, which uses word-triples, is known as a trigram. In fact, it is possible to extend the number of words in probable sequences, but the combinations will become unmanageable. High order N-gram is employed to narrow down the „envelope" of words from which the system can make its choice. Thus, the use of this technique embodies the syntax and context of a given topic of discussion without having to directly address any specific problems of either. On other hand as the order of n-gram increased large training set is require maintaining the performance [10] [27].

Drawback of N-gram

N-gram language model of Markov models or assumption is insufficiency for prediction since linguistic information and proficiencies has long distance syntactic dependencies. According to Chomsky English cannot be modeled by a Markov chain because of long distance syntactic dependencies. On other hand the probabilities of an *N*-gram come from the corpus it is trained on. Thus, it is very dependent on the training corpus. Training corpus has to be large enough to ensure that each valid word sequence appears a relevant number of times. N-gram language models are challenged by data sparseness, and by cross-domain brittleness, the effectiveness of an n-gram prediction model is highly dependent upon the size of the training text. Since the accuracy of n-gram prediction methods is highly dependent upon their statistical reliability, the effect of training text size on performance. The great amount of computational resources is needed especially if the number of words in the lexicon is big [28] [27].

2.6.2 Knowledge Based Word Prediction

Word prediction systems that merely use statistical modeling for prediction often provides words that are syntactically, semantically, or pragmatically inappropriate in a given context. As a result, they impose a heavy cognition load on the user to choose the intended word. Consequently, its shortcoming is reducing the writing rate. Omitting inappropriate words from the prediction list and providing orthographical assistance would give more comfort and confidence to the user. Pertaining to the linguistic knowledge that could be used in prediction systems would be discussed below [25] [29].

Syntactic Prediction

Syntactic prediction is a method that tries to present words that are appropriate syntactically in that position of a sentence. Some of these systems consider part-of-speech tag information of words as syntactic information, while others use a parser to build the syntactic structure of the whole sentence. Regarding this, statistical syntax and rule-based grammar are two general syntactic prediction methods [30] [29].

Statistical Syntax

Statistical syntax uses the sequence of syntactic categories and POS tags for prediction. Therefore, a probability would be assigned to each candidate word by estimating the probability of having this word with its tag in the current position and using most probable tags for previous one or more words. This means the appearance of a word in this method is based upon the correct usage of syntactic categories. In other words, the Markov assumption about *n*-gram word tags is used. In the simplest method, the POS tags are sufficient for prediction. Therefore, a probability would be assigned to each candidate word by estimating the probability of having this word with its tag in the current position and regarding the most probable tags for the previous words [9].

In another approach, the predictor tries to estimate the probability of each candidate word according to the previous word and its POS tag, and the POS tag of its preceding words. In other words, the system uses word bigram and POS trigram model. A linear combination model of POS tags tries to estimate the probability of POS tag for the current position according to the two previous POS tags. Then it attempts to find words that have the highest probability of being

in the current position according to the predicted POS tag. Then, it combines this probability with the probability of the word given the previous word. So, there are two predictors in which one predicts the current tag according to the two POS tags and the one that uses bigram probability to find the most likely word [25].

In general, statistical syntax prediction employee the Markov assumption about *n*-gram methods that can be used to obtain statistical knowledge about the syntax using one the methods like POS tags only, previous word and two previous POS tags, and linear combination

In rule based grammar approach, syntactic word prediction would be made by using the grammatical rules of the language. A parser will parse the current sentence by using the grammar of the language to reach to its categories. The parsing method can be either top-down or bottom-up. Phrase Structure Rule Grammar (PSRG), Context Free Grammar (CFG), and Head-driven Phrase Structure Grammar (HPSG) are the methods that could be used in prediction systems based on grammatical rules.

Semantic Prediction

Naturally, in every language, language items which are syntactically correct may be semantically incorrect. Thus, some of the predicted items in the prediction list could be wrong semantically even though they are syntactically right. So, suggesting the words that are syntactically and semantically correct would increase the accuracy of the predictions. To attain this goal, a great semantic knowledge is tagged to the words and phrases in a corpus. Mostly in semantic prediction appearance of specific word with special content is a clue to increase the probability of appearing other words that have semantic relationships to that word [25].

2.6.3 Heuristic Modeling

To make predictions more appropriate for a specific user requires systems that learn from users' writing activities. This approach tries to get adapted the system to every individual user. A word prediction system intended to support writing especially on mobile phone and other handheld devices are commonly used adaption method. This approach is based on the assumption users dictionary is limited; they used some word frequently rather than using new word. Thus, it provides an indication of how recently a word has been used as this will increase the likelihood of it being used again. There are two general methods that make the system adapted to the users. One of the methods is short-term learning and the other one is long-term learning [10] [25] [31].

Short-term Learning

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

Recency Promotion is a one short learning method that emerged from the cognitive psychology concept. This is based word that has already occurred in a text will be given a higher probability of use; thus, more likely to be used in that text again. Such a method usually assigns dynamically higher probabilities to the words that recently are used in the text; so, it does not only take into account what words have been typed; but further, how recent they have been used [23].

Topic Guidance approach is a way of adapting the predictor to the overall subject of the current text. To do so, the general lexicon is complemented with a domain specific lexicon that contains words which are frequently occurring within certain domains, though not very common in general [25]

N-gram Cache is other short leaning approach, based on the assumption that if a word is used once, it is more likely to be used again. In other words, the previous use of a word in a context increases the probability of that word to be used again. Using n-gram cache is a way to capture the most common words and sequences that are frequently used. These words would be put in the cache to get an increased probability.

Long term learning

In long term method, the system gets adapted to the user by considering not only the current text, but previous texts that are produced by the user. As a result, gradually by using the system more, it adapts to the user heuristically. Some of the methods for heuristics adaptations that are language specific are adding new words, automatic capitalization, providing inflected form of words, and compounding [25].

Adding New Words is long term learning method mostly used for adding unknown word to dictionary. It adapts unknown words to the lexicon of the system whenever the user types unknown words to the system. The added new words could be called in the prediction list for future use. Thus, it provides a technique to outclass the problems of lexicon coverage [30].

Other long term learn method is called Automatic Capitalization. Depending on the language that the system is running for, some letters should be capitalized. For example, the first letter of a word at the beginning of a sentence and also proper words must be capitalized. Automatic capitalization allows the user to save more keystrokes [23].

2.7 Word prediction for inflected language

Word prediction is facing a challenge, as several typical complex problems arises when dealing with Natural Language. The inherent amounts of arising ambiguities not only lexical, structural and semantic ambiguities but also pragmatic, cultural and phonetic ambiguities for speech are complex problems to be solved by a computer. Especially for inflected language, as language that the system has to model influences the prediction techniques; inflected languages pose a harder challenge to prediction algorithms.

Word prediction is a challenge for inflected languages, that is languages that have a large dictionary of word forms with several morphological features, produced from a root or lemma and a set of inflection rules. The large number of word forms makes word prediction for inflected languages a hard task. Since it requires dealing with a usually high number of inflected forms that dramatically decrease Keystroke Saving [30].

Dictionary coverage is another dominant factor affecting prediction word especially for inflected languages. Typically, for inflected languages obtaining all vocabulary that language consist would be difficult since corpus for language modeling is not expected to include all words of specific language. Thus, inflected language poses other challenge on word prediction that required to be solved such as out-coverage dictionary.

In practice, it is possible to incorporate some syntactic and semantic information due to the dependencies between words that are captured and estimated by n-grams words models; usually, as the vocabulary size is very large. Although, n-gram models of words particularly trigram have been used successfully for many NLP tasks but they suffer the well-known drawback of being inadequate for inflected languages since the parameters space becomes too wide, both for the vocabulary size and the training corpus.

Many researches efforts have been experimented and several core NLP tasks have been employed for improving the performance of prediction model for inflected language including Language Modeling, Part-of-Speech (POS) Tagging, Parsing and Lemmatization. Thus, different new models where introduced with the purpose of reducing the parameters space of inflected language. Some researches explored a model that enables the n-gram model making use of n-grams of Part-of-Speech: via a linear combination POS trigrams and simple word bigrams the context is forced into an equivalence class determined by a function.

$$Pr(Wi|\varphi[Wi-n+1,\dots,Wi-n+1])$$

Part-of-Speech tags are considered as function ϕ to restrict exponential increase of the context. Such tags capture many different word forms, so contextual dependencies are represented in smaller set of n-grams. Using POS tags, a larger surrounding information may be taken into consideration but there is a loss in semantics since different words may be captured in one word class and tags only inform about sequences of words classes and not which particular word is typically connected with previous words or words classes. This technique also claimed as it limited to grant a lexical coverage.

To simplify the task of predicting the correct form, some techniques provide a two-step procedure, choosing first only among word “roots”, and proposing all the possible word forms only when the user selects a root [9]. Instead other researches provide a one-step procedure, by considering Part-of-Speech (POS) and related morph-syntactic information to provide the user a list of word forms.

This method employed a large morpho-syntactic tagged corpus to train the language model and a Part-of-Speech tagger that annotates on-the-fly words with their POS and related morpho-syntactic information. This procedure enriches the language model with deep morphological information and combined it with on-the-fly POS tagging that enables a model to boost performances, cutting off of the prediction list all words whose gender, number, tense or mood are not consistent with the sentence context [29].

2.8 Performance measurement for word prediction system

Word prediction is an application of language modeling to speeding up text entry. Word prediction enhances text entry rate by reducing the number of keystrokes required to produce a message. Thus, as the goal of word prediction systems is to reduce the number of keystrokes, the primary metrics for evaluation of word prediction is keystroke saving (KS). The common trend in research is to simulate a "perfect" user that will never make typing mistakes and will select a word from the predictions as it appears without use of backspace and edit the word. Keystroke saving is calculated by comparing the total number of keystrokes needed to type the text without the help of the word prediction and the effective number of keystrokes saved using word prediction. A higher value for keystroke saving implies a better performance. [22] [32]

$$\text{KS} = \frac{Keys\ normal - keys\ with\ prediction}{keys\ normal} * 100\%$$

Keystrokes until Completion (KUC) is another metrics to evaluate word prediction systems. It computes the average number of keystrokes that a user enters for each word before it appears in the prediction list. Being $c1 \ldots cn$ the number of keystrokes for each of the n words before the desired suggestion appears in the prediction list. Lower value of KUC shows better performance.

$$\text{KUC} = \frac{c1+c2+c3}{n}$$

Hit Rate (HR) is word sequence prediction measuring metrics. It is defined as the percentage of correct words that appear in the suggestion list without entering any letter of the following word. In other words, it is the relation between the number of times that a word is guessed and the number of written words. The higher hit rate implies a better performance.

Accuracy is also metrics to evaluate word sequence prediction. It calculates the percentage of words that have been successfully completed by the program before the user reached the end of the word. A good completion program is one that successfully completes words in the early stages of typing.

CHAPTER THREE: RELATED WORK

Word prediction is one of the most popular research areas in Augmentative and Alternative Communication (ACC) and writing aid via the application offered from the field of natural language processing. Many researches have been conducted for different languages using different approaches. Most of them have employed statistical approach of the development of language model. And other researches have employed statistical approach with linguistics knowledge of the language to improve the performance of word prediction models. The next section presents word prediction studies conducted for both inflected and non-inflected language with regard of the works those systematically studied to look for the best approach for Afan Oromo language

3.1 Word Prediction for Asia Languages

3.1.1 Automated Word Prediction in Bangla Language Using Stochastic Language Models

Haque and Habib [33] studied word prediction on Bangla language using N-gram language model such as unigram, bigram, trigram, deleted Interpolation and back-off models for auto completing a sentence by predicting a correct word in a sentence. The corpus of 0.25 million words containing 14,872 word forms from Bangla newspaper called as "Prothom Alo" was constructed for the study. The prepared corpus was divided in to training and test set: two-thirds for training and one-third for testing. In order to avoid model over fitting problem, such as training error and generalization error, holdout method was used as validation set. In accordance, the original training data split into two subsets with two-thirds of the training set for model building while the remaining one-third was used for error estimation.

Finally, N-gram model of word prediction was constructed using unigram, bi-gram and tri-gram by counting frequencies of words in a training corpus. To solve the problem of zero sentence probability, they applied back-off and deleted interpolation model. In the back-off method for a trigram model, the word sequences will follow trigram probabilities at first; if it could not match, then word sequences will follow bigram model; if it also could not match, then word sequence will follow unigram model and predict at least a word.

The deleted interpolation algorithm combining different N-gram orders by linearly interpolating all three models when they are computing any trigram. The findings of the study showed that

language models have performed almost in the same trend-line. The Bigram model performed modest; whereas the unigram model performance was very poor. The average accuracies of models were 21.24%, 45.84%, 63.04%, 63.50% and 62.6% for Unigram, Bigram, Trigram, Back-off and Delete interpolation respectively [33].

3.1.2 A Stochastic Prediction Interface for Urdu

Qaiser Abbas [34] conducted research entitled "A *Stochastic Prediction Interface for Urdu*" to develop a model that predict word like t9 and a sequence of word. The research was aimed to extend the application of t9 by addressing the limitation of t9 that only predict a word after typing initial characters. The researcher assumed prediction Suffix Tree (PST) is the best strategy for prediction. But due to non-availability of resources, he decided to move into consecutive steps and as a first step; the N-gram approach has been adopted in the construction of PST model. Thus, the N-gram models were developed using unigram model used for T9, and bigram and tri-gram models were used for word sequence prediction.

Unigram model was trained on a merged corpus of 5000 most frequent words of Urdu collected from a 19.4 million words corpus and other corpus containing 1 million words. These two different corpora were merged to build a sufficient amount of data. In contrast of the unigram, only 1M portion of the corpus was used for bi-gram and tri-gram models because the document of 5000 most frequent words contained only the unique unigram counts and no any raw text of Urdu. In general, all models were constructed from corpus that was divided into training and the test data individually according to the standard division of 80% and 20% respectively. Then from both training and test data 10% and 50% were respectively reserved for a held-out data for an experimental purpose.

The performance of the models was evaluated using percentage of keystrokes saved, keystrokes until completion and a percentage of time saved during the typing. Two different performance evaluations were performed on unigram, bigram and trigram models with respect to parameter L, which is the length of the predicted text in characters including the typed keystrokes. These evaluations were performed on different lengths ranging from 15 to 50.

Firstly, the performances of the models were evaluated on the test sets. Thus, the experiment result showed that 52.77% average KS was gained with unigram model when l is short, but the result decreased as l length increased. On the other hand, bi-gram model scored the average KS of

34.61%. While 7.8% and 26.23% were gained using KUC and WTS respectively when l was short, but it increased as l became long. Similarly, trigram model gains better performance when the desired text becomes long. Secondly the performances of the models were evaluated with the 10% held out data. The models trained on the training and the held-out data sets. The models predicted almost equal in case of unigram, bigram and trigrams when the length L is kept less than or equal to 20. However, when the length of text was raised beyond 20, then it started very rapidly [34].

3.1.3 Probabilistic Analysis of Sindhi Word Prediction using N-Grams

Mahar and Memon [35] conducted a research on word prediction model based on statistical method for Sindhi language. They identified Sindhi language is highly homographic language and text is written without diacritic symbols, which is a challenge for word prediction task. The proposed method to develop the prediction model was using three N-gram models i.e. bigram, trigram and 4-gram. The corpus for their study was collected from different sources like newspapers, magazines and books from Internet. And those documents had different genre arts, sports, politics, environment and music. The corpus containing 3 million tokens were divided into training set containing 2924967 word tokens and 338831 word types and test set containing 135362 word tokens and 26426 word types. As the documents were collected from different sources having different file format, they observed spelling error and the absence of short vowel symbols during file conversion process. Thus, they corrected those mistakes manually. N-gram models of bigram, trigram and 4-gram were employed to estimate the probability in given history of words. Then the models trained on training set by counting and normalizing it using Add-one smoothing technique to assign non-zero probabilities to all N-grams having zero probabilities. After that, they conducted experiment on test set through randomly selection of sentences from corpus. Then, to evaluate the three n-gram models, they applied n-gram grammars and each n-gram results stored on separate database. Finally, every N-gram models were compared in terms of accuracy to find out which model was more suitable for test set of the language by using perplexity. The experiments' result on test corpus indicated that the 4-gram model was suitable for Sindh language as the 4 –gram had lower perplexity value than other N-grams [35].

3.2 Word Prediction for European Languages

3.2.1 Context based word prediction

Agarwal and Arora [*20*] conducted research entitled "*Context based word prediction for texting or language*" for mobile phones that assists SMS (Short Message Service) compose in order to predict the most appropriate word for a given code (On a phone keypad, multiple words are mapped to same numeric code). They proposed a Context Based Word Prediction system for SMS messaging in which context was used to predict the most appropriate word for a given code. They also extended this system to allow informal words (short forms for proper English words). The mapping from informal word to its proper English words was done using Double Metaphone Encoding based on their phonetic similarity. During the study, three random variables namely the code, word and its POS were used. Context-based word prediction system for formal languages was developed by Graphical Model using bi-gram (Model-I) and Graphical Model using HMMs (Model I, II, III). Model-I and three models for HMM were used on the informal data. The experiment for formal language was conducted by using 19,000 emails for training and 1900 emails for testing purpose. The results of the experiments were evaluated by using the average error rate of each model in relation to the average error rate of the frequency based approach. As a result, the average error for Model-I (first model) was 5.54%, for HMM-I 6.69%, for HMM-II 11.97%, for HMM-III 8.05% and for Frequency based prediction the average error was 8.04%.
Comparing to frequency-based method, the average error rate was reduced for the first model, Model I and HMM- I model by 31% and 16.8% respectively. On the other hand, the error for HMM-II was greater than the frequency-based approach by 39% while the error rate for HMM-III model was almost alike to the error rate for frequency-based approach. SVM was assessed in order to test how it performs in classifying words for a given code. So, it was tested on 10 codes, referring to few very frequent English words. The result of SVM was also compared with frequency-based approach and first graphical model, Model-I based on the selected words. As a result, the average error for SVM was reduced by 18.62% as compared to the frequency-based method, hence; SVM performed better. The average error for Model-I was reduced 35.75%.as compared to SVM. At this point, Model-I perform better over SVM. Informal language models the dataset that consists of 850SMS messages was used. On the informal data, Model-I and three models for HMM were tested. Among these models, Model-I performed the best for informal language by reducing the

average error by 22.33% as compared to the frequency-based model. The researcher concluded that the performance of the Context Based Word Prediction system was better than frequency based method. For the problem identified, the combination of SVM and HMM model (SVMHMM) used for sequence tagging was identified was unsuitable due to the large number of classes. Then researcher suggested that the bi-gram model used in graphical model performs better than others. [20]

3.2.2 Effects of N-gram Order and Training Text Size on Word Prediction

In article entitled "Effects of n-gram order and training text size on word prediction", Lesher [36] studied the impact of N-gram order and training corpus size on the performance of word prediction regarding keystroke saving. The purpose of the study was to quantify the impact of adopting higher-order n-gram model that rely upon increased word context. Additionally, it explored the dependence of performance on the size of the corpus used for statistical language modeling. Training corpus of the study was constructed by equally combining text blocks from the Brown corpus, the LOB corpus, and a collection of Time Magazine articles. All headings and formatting directives were removed from the training texts. As a result, comprehensive n-gram statistics were automatically generated and stored for each training set. Finally, twenty-one experimental conditions were established by combining three different N-gram orders (unigram, bigram, and trigram) with each of the 7-training set.

The performance of the models was evaluated over seven test sets: each test set having more than 2500 words. However, the content of the testing was independent from that of the training set, taken from the researcher's previous word prediction study corpus. Training and test data set had different genre and writing style. For each experimental condition, the seven testing sets were independently generated using a 54 key QWERTY keyboard supplemented by a 10-word prediction list accessed using the F1 through F10 keys. After that, keystroke savings were computed for each testing set based on the numbers of keystrokes used to produce that text with and without prediction enabled, as keystroke savings were averaged across testing texts to provide a single performance measure for each condition.

The result of experiment showed the average keystroke saving for unigram, bigram, and trigram models increase as the number of words in the training set increase. The Performance of the model increased when training text size increased, irrespective of the n-gram order. However, for a given

training text size, keystroke savings also increased steadily with higher n-gram orders. The increment was much more marked for trigrams with 7.5% keystroke saving while unigram limited to 4.5 % keystroke saving. A large jump in keystroke savings was realized while moving from unigram to bigram model with 6.4% points at 3 million words, reflecting the transformation from context-insensitivity to context-sensitivity. The performance gained in moving from bigram to trigram models was considerably less dramatic with 0.8% variation, although the difference grew for larger training set [36].

3.2.3 Advances in NLP applied to Word Prediction

Aliprandi and Carmignani [30] conducted a research on word prediction for inflected languages, particularly for Italian language. They presented the limitations of statistical techniques for inflected languages, languages that have a large dictionary of word forms with several morphological features, produced from a root or lemma and a set of inflection rules. Thus, inflected languages pose a harder challenge to prediction algorithms since they had to deal with a usually high number of inflected forms that dramatically decrease Keystroke Saving. To outclass this problem, they designed a word prediction system called FastType.

FastType was based on combining statistical method with Part-of-Speech (POS) and related morph-syntactic information to provide a one-step procedure. Thus, the model suggests a list of word by considering words whose gender, number, tense or mood that are consistent with the sentence context.

The user interface, predictive engine and linguistic resource were main components of the FastType system. The Prediction Engine was the kernel of the Predictive Module. That manages the communication with the User Interface, and it predicts a list of words by assuring the agreement between gender, number, person, tense and mood with the syntactic sentence context. In general, Predictive Module provides core functionalities, such as the morph-syntactic agreement and the lexicon coverage, efficiently accessing the Linguistic Resources.

Language model was developed from POS n-grams and Tagged Word n-grams. The prediction algorithm based on Linear Combination algorithm combined POS n-gram models with tagged word n-gram models. A word and POS bigram and trigram models had trained corpus created from newspapers, magazines, documents, commercial letters and emails.

Keystroke saving (KS), Keystroke until completion (KUC) and Word Type Saving (WTS) are three parameters used to evaluate the system. The researchers indicated that 40 texts disjointed from 35 training set were used for testing. However, the size or number of words available in the testing data was not clearly specified. The result shows 51% keystroke saving, which is comparable to what was achieved by word prediction methods for non-inflected languages. Moreover, on average 29% WTS, meaning at standard speed without any cognitive load saving in time and 2.5 KUC is observed [30].

3.3 Word prediction for Ethiopian languages

In our country's context, so far researches on word prediction were conducted for Amharic and Afaan Oromo languages. Nesredin Suleiman [2] and Tigist Tensou [9] have conducted research on Amharic word prediction. On the other hand, Gudisa Tesema [10] made the first attempt to design and develop Afaan Oromo word prediction on mobile phone

3.3.1 Word Prediction for Amharic Online Handwriting Recognition

Nesredin Suleiman [2] conducted a research on word prediction for Amharic online handwriting recognition. The researcher interested to explore the hypothesis that speed of data entry can be enhanced with integration of online handwriting recognition and word prediction mainly for handheld devices. The main purpose of his study was to complete a word currently being typed by a user. Here, characters are suggested to complete the word using statistical information like frequency of occurrence of words. A corpus of 131,399 Amharic words and 17, 137 names of persons and places were prepared. The prepared corpus was used to extract statistical information like to determine value of n for the n-gram model, average word length of Amharic language, and the most frequently used Amharic word length. Hence, n is set to be 2 based on statistical information. The research was done using bi-gram model, where the intended word is predicted by looking the first two characters. Finally, a prototype is developed to evaluate performance of the proposed model and 81.39% prediction accuracy is obtained according to the experiment [2].

3.3.2 Word Sequence Prediction for Amharic Language

Tigist Tensou [9] conducted research on word sequence prediction for Amharic language. The main target of the study was to design and develop word sequence prediction model for Amharic language with inclusion of context information, predict words that a user intends to type based on context information is the task of word sequence prediction. The researcher was motivated to the study to fill previous research gap that was conducted on Amharic word prediction research by [2], that has been done to complete a word a user is currently typing using dictionary of words with their frequency. She claims that it is impractical to capture all word forms using only statistic of word due to the language ‘s rich morphology. Moreover, she identified [2] model does not consider context information. Thus, the lack of incorporating context information produced syntactically wrong word predict that cause extra cognitive load to adjust suggested words to appropriate form as well as causing reduction in speed of text entry. In the study, the researcher identified Amharic language has very complex inflectional and derivational verb morphology with four and five possible prefixes and suffixes respectively. It is morphologically complex and makes use of both prefixing and suffixing to create inflectional and derivational word forms which also requires some degree of infixing and vowel elision. Hence, to address this problem, she cast the word sequence predictor will propose root or stem word and morphological features internally with the aim of offering appropriate word form to the user.

The corpus for study was collected from Walta Information center, and a training corpus containing 298,500 sentences was used for model development. In addition, POS tagged corpus containing 8067 sentences was used to extract representative sentences for testing by means of random sampling method. Prediction model was developed using statistical methods and linguistic rules. Statistical models were constructed for root or stem and morphological properties of words like aspect, voice, tense, and affixes using the training corpus. Consequently, morphological features like gender, number, and person were captured from a user ‘s input to ensure grammatical agreements among words. Initially, root or stem words were suggested using root or stem statistical models. Then, morphological features for the suggested root or stem words were predicted using voice, tense, aspect, affixes statistical information and grammatical agreement rules of the language. Finally, surface words were generated based on the proposed root or stem words and morphological features.

Hornmorph morphological analyzer and generator program was employed to analyze the corpus and to produce surface words. Furthermore, Python programming language was used for Prototype development to implement statistical language models of tri-gram, bi-gram, and hybrid. Finally, evaluation of the model was performed using keystroke savings (KSS) as a metrics. According to the experiment, prediction result using a hybrid of bi-gram and tri-gram model has higher KSS and it is better compared to bi-gram and tri-gram models. [9]

CHAPTER FOUR: AFAAN OROMO LANGUAGE

4.1 Historical and Demographic Background of Afaan Oromo

Afaan Oromo is one of the most widely spoken languages in Africa, surpassed only by Arabic and Hausa [1]. The language is termed as, '*Afaan Oromo'* (*the Language of Oromo*) because it is used by Oromo Society, the native ethnic group of Ethiopia that account for the largest population of the country. With regard to this, various scholars revealed that the Oromo People are the largest single ethno-nation in Eastern Africa [2], constituting at least 40% of the Ethiopian population [3]. According to Hussein [4], "The Oromo people speak Afaan Oromo (the language of Oromo), which belongs to the Eastern Cushitic family of Afro-Asiatic phylum." Studies reveal that Afaan Oromo is the most important language of Ethiopia where it is used not only as a national (official) language by the Oromo people but also as a *lingua franca* by several million speakers of other languages [5]. Outside Ethiopia, the language is spoken by thousands of other Oromo tribes in Kenya [4]. In line with this**,** Dejene [5] states as, **"**It is a language of a great people with national history going back at least to the 16th century that played a major political and cultural role in North-East Africa and whose cultural and social organization (e.g. the famous 'Gada' system) are among the most outstanding in Africa". Besides being the widely used language in Africa, Afaan Oromo has been included among the essential languages in the world. Justifying this, the report by the U.S Government and its Education Department (1985) has revealed that Afaan Oromo has been considered as one of the 169 critical languages of the world [6]. Based on the aforementioned historical and demographic issues of Afaan Oromo, researching the different aspects of the language is worth mentioning. Accordingly, the current research focuses on NLP for Afaan Oromo is among the basic issues that need to be studied in order to enhance the wide usage of the language during the current digital age when the use of technological resources is increasing dramatically. The study involves conceptualizing the writing system of the language including its alphabets and sound systems as well as describing its syllabification, morphological process and grammatical rules as these issues are the basics for studying the word sequence prediction systems.

4.2 The Basic Tenets of Afaan Oromo Writing System

4.2.1 An Overview of Afaan Oromo Writing System

Different scholars identified that the writing system Afaan Oromo relies on Latin Script; the alphabets and sounds of the language are modifications of Latin writing system. Thus, Afaan Oromo shares a lot of features with English writing system except some modifications, and the writing alphabet of the language is known as 'Qubee Afaan Oromoo' which is designed based on the Latin script. Thus, letters in English or Latin Alphabets are also found in Afaan Oromoo except the ways they are combined in phonetic alphabets and the styles in which they are uttered [7].

In order to get some insights into the writing system of Afaan Oromo, it is very important to look-through the alphabets and sound systems of the language; hence, the interplay between alphabets and sound systems of the language could be described below.

4.2.2 Description of Afaan Oromo Alphabets and Sound Systems

As it has been mentioned before, Afaan Oromo uses Latin character but with some modifications on sound of consonant and vowels. It has 28 letters called Qubee. However, later on a new letter" Z" was included in the alphabet as there are words which require the letter. For example: "Zelaya" (gold), Zeeytuuna (guava), Azoole (river in Arsii), Zeekkara (Opera), Zalmaaxaya (mess), Waziiza (fire place or fire work) and Zawii (insanity) are Afaan oromo words written using "Z" Additionally 'P and V 'are also added. 'P and V' letters are not Afaan Oromo letters because there is no Oromo word written by use of either of them. But they are included by considering the fact of handling borrowed terms from other languages like English. For example: "Police", "Piano", "Television", "video" and etc. To sum up there are 31 letters of Afaan Oromo including 'Z ', 'P ', and 'V '[8].

Vowels (Dubbachiiftuu)

There are five vowels in Afaan Oromo; these are '**a**', '**e**', '**o**', '**u**' and '**i**'. They are similar to that of English, but they are uttered differently. Each vowel is pronounced in a similar way throughout its usage in every Afaan Oromo literature [8]. In other words, there is no rule which could violatetheir pronouncing style in difference contexts. There is no need to deal with phonetic

transcription because the pronunciation is made just as the words are normally written in different texts.

Consonants (Sagaleewwan dubbifamtoota)

Most Afaan Oromo constants do not differ greatly from Italian, but there are some exceptions and few special combinations.

4.3 Morphological Issues of Afaan Oromo

4.3.1 Description of Afaan Oromo Morphology

Like in a number of other African and Ethiopian languages, Afaan Oromo has a very complex and rich morphology [9]. It has the basic features of agglutinative languages involving very extensive inflectional and derivational morphological processes. In agglutinative languages like Afaan Oromo, most of the grammatical information is conveyed through affixes, (that is, prefixes and suffixes) attached to the root or stem of words. Although Afaan Oromo words have some prefixes and infixes, suffixes are the predominant morphological features in the language. Almost all Afaan Oromo nouns in a given text have person, number, gender and possession markers which are concatenated and affixed to a stem or singular noun form. In addition, Afaan Oromo noun plural markers or forms can have several alternatives. For instance, in comparison to the English noun plural marker, *s (-es),* there are more than ten major and very common plural markers in Afaan Oromo including: ***-oota***, ***-oolii***, ***-wwan***, ***-lee***, ***-an***, ***een***, ***-eeyyii***, ***-oo,*** etc.).

As an example, the Afaan Oromo singular noun ***mana*** (house) can take the following different plural forms: ***manoota (mana*** + ***oota), manneen (mana*** + ***een), manawwan (mana*** + ***wwan)***. The construction and usages of such alternative affixes and attachments are governed by the morphological and syntactic rules of the language [10]. Afaan Oromo nouns have also a number of different cases and gender suffixes depending on the grammatical level and classification system used to analyze them. Frequent gender markers in Afaan Oromo include ***-eessa/-eettii***, ***-a/-ttii*** or **–** ***aa/tuu***.

For example:

Word	*Construction*	*Gender*	*English*
Obboleessa	*Obbol + eessa*	*Male*	*brother*
Obboleettii	*Obbol + eettii*	*Female*	*Sister*
Beekaa	*Beek + aa*	*Male*	*Knowledgeable*
Beektuu	*Beek + tuu*	*Female*	*Knowledgeable*

Likewise, Afaan Oromo adjectives have case, person, number, gender, and possession markers similar to Afaan Oromo nouns. Afaan Oromo verbs are also highly inflected for gender, person, number, tenses, voice, and transitivity. Furthermore, prepositions, postpositions and article markers are often indicated through affixes in Afaan Oromo [9] [10]. The extensive inflectional and derivational features of Afaan Oromo are presenting various challenges for a number of NLP tasks in the language [9] [10] [7].

4.3.2 Syllabification in Afaan Oromo

According to Abebe [11], *Syllabification* is language-dependent: each language has its own structure of syllables. For example, in English more than two consonants can come consecutively in a single word as in ‘screen’. But, in Afaan Oromoo more than two consonants cannot come together except in diagraphs. Hence, there are four types of syllable structure in the language. These structures include CV, CVV, CVC and CVVC. All of these can be found at word initial, medial and final positions. A valid word can be composed from the combination of one or more type(s) of these structures. The words like eelee, ooluu and etc. seem to start with vowels, but linguists argue that there is hidden glottal stop called hudhaa (‘) in front of any word that seems to start with vowel.

In light of this, we say that every syllable in Afaan Oromoo starts with consonant. The following are just few examples.

CVC	*Shan (five)*
CV	*Na- ma (man), tu-re (stay), Ku-ma (thousand)*
CVVC	*Deem- (go)*
CVV	*Boo-naa (personal name)*

Majority of words in Afaan Oromoo are disyllabic with considerable number of tri-syllabic ones. Monosyllabic and quadric-syllabic words are rare [11].

4.3.3 Morphological Processes/Word Formation in Afaan Oromo

Different studies identified that there are two productive ways to form words from morphemes: *inflection* and *derivation* [7] [11].

Inflectional Morphology: deals with the combination of a word with a grammatical morpheme, usually resulting in a word of the same class as the original stem, and serving some syntactic function, for example plurals of nouns. They do not change the part-of-speech category but the grammatical function (also called morpho syntactic information) is changed. The different forms of a word are produced by inflection. In English, the word ‘work’ is a verb, and inflectional forms like 'works', 'working’, and 'worked' are produced by adding the 3rd person singular maker /-s/, the present continuous marker /-ing/ and the perfective /-ed/ respectively. These four word forms of ‘work’, i.e. ‘work', 'works', 'working', and 'worked' are all verbs and there is no change in the part-of-speech category due to the affixation.

Derivational Morphology: creates new words (i.e., words with a different part-of-speech category) by adding a bound morpheme to a stem. Derivation can be applied recursively, i.e., words that are already the product of one derivation process can undergo the process again. The following is an example from English: large (adj.) = enlarge (en- + large) (v) = enlargement (enlarge + -ment) (noun), and from Afaan Oromoo: bar- ‘to know’ (v) _ barumsa ‘education’ (bar-+-umsa) (noun). Another technique of word-formation in Afaan Oromoo is *compounding*.

Compounding is a process of forming new words by combining different lexical categories [11]. However, it is not the case that every two words combine to form a compound form. Rather, every language follows certain rules by which it forms its compound. In Afaan Oromoo, the combination

of abbaa ‘father’ + lafa ‘land’ forms new word abbaa lafaa ‘landlord’. The rules of compound word formation in Afaan Oromoo is unpredictable, and thus needs further linguistic study in the language.

On the basis of structural changes of the stem and other morphemes during affixation, morphological processes can also be classified as linear or nonlinear. In linear morphology affixes are added to the stem without changing the internal structural of the stem, though some changes might take place at the boundary of stems and affixes. On the other hand, morphological systems where the internal structure of the morphemes changes during the addition of suffixes are classified as nonlinear morphology. English pluralization pertains to linear category; similarly, morphological processes in Afaan Oromoo are mostly linear in nature [11].

4.3.4 Morphophonemic Processes

One of the main occurrences of morphophonemic changes is the change that takes place between the boundary of stems and inflectional or derivational suffixes. In Afaan Oromoo the change may be assimilation, epenthesis, metathesis, deletion, reduplication and so on. In the following section, we briefly discuss each of them.

Assimilation

The phonemes that come next to each other at morpheme or word boundary may take the form of the previous or next. This produces the combinations of a variety of stem-final consonants followed by **t** (third person singular feminine, second person singular and second person plural), n (first person plural, neutral common), **s** (common, causative-common singular) and so on. The change can take place between prefix and stem or stem and suffix. Some of the changes are optional because they differ according to the dialect spoken. Table 4.1 summarizes the change.

Table 4:1Assimilation Processes

Combination of phonemes	**Result**	**Example**
d + s	*ch*	*duud - + sa = duucha*
dh + s	*ch*	*nyaadh - + sisa = nyaachisa*
dh + n	*N*	*fuudh - + na = fuuna*
d + n	*nn*	*did - + n = dinna*
t +n	*Nn*	*dhaloot - +ni = dhaalonni*
t + ch	*ch*	*hojjet - + chisna = hojjechisna*
x + s	*cc or ch*	*fix - + siise = ficcisiise / fichisiise*
t + dh	*dh*	*barat - + dhu = baradhu*
dh +t	*T*	*fuudh -+ tan = fuutan*
l + s	*Ch*	*awwaal - + sise = awwaalchise*
b + t	*Bd*	*waraab - + te = waraabde*
s + t	*Ft*	*baas - + te = baafte*
d +t	*Dd*	*yaad -+ te = yaadde*
l +n	*H*	*gal - + ne = galle*
g + t	*Gd*	*dhug - + te = dhugde*
x +t	*Xx*	*fix - + te = fixxe*
c + t	*Cc*	*boc - + te = bocce*
j +t	*Jj*	*ajaj - + te = ajajje*
r + n	*Rr*	*abaar - + ne = abaarre*
s +n	*fn*	*baas - + ne = baafne*

Deletion

In Afaan Oromoo for the convenience of speaking, phonemes at word or morpheme boundaries are deleted. This process usually occurs in noun derivations or inflections.

For example:

Mana 'man' + -oota	*= manoota 'men'*
Nama 'man' +-ummaa	*= namummaa 'personality'*

In verbs, deletion usually takes place in stems ending with 'h, dh, hudhaa (')'.

Hodh- +te	*= hoote*

Epenthesis

In Afaan Oromoo, more than two consecutive consonants cannot occur together. When more than two consonants occur consecutively /i/ or others will be inserted between them. This process is sometimes called insertion, and is triggered on the basis of phonological information.

For instance, Elm- + -na = Elmina, and Sirb- +ta = sirbita

Reduplication

Reduplication is formed by copying the first consonant and vowel of the verb stem and geminating the second occurrence of the initial consonant. The resulting word indicates the repetition or intensive performance of the action of the verb. Generally, if the stem starts with consonant, reduplication has the form of CV(C) + stem, where C=consonant and V= vowel. But, it has the form of V (') + stem if the stem starts with vowel. Though adjectives can undergo reduplication, we only discuss the reduplication process in verbs in this thesis.

For example: Deemuu=deddeemuu, ciruu =cicciruu, affeeluu=a'affeeluu

4.4 Categories of Afaan Oromo Words

The uses of words are identified based on their rules to be applied in different contexts and morphological categories. Identifying categories of words depend on different aspects. In this regard, Abdi [12] stated that the category of a particular word can generally be identified by looking at the semantic of that word, by looking at the form (morphology) of that word or by looking at the actual position (syntax) of that word. The grammatical categories of Afaan Oromo have undergone a series of improvement in terms of its word categories and other syntactic features. In line with the basic classifications of words in English language, Afaan Oromo words are categorized into eight grammatical categories (Noun, Verb, Adjective, Adverb, Adposition, Pronoun, Conjunction and Interjection); however, some researchers like Mandefro (2012) Legesse, Assefa (2005) and Getachew (2009) as cited in Abebe [12] revealed that Afaan Oromo has five grammatical categories of words such as: nouns, verbs, adverbs, adjectives and Adposition. According to these researchers' pronouns included under the noun category, and conjunctions and interjections under Adposition. From this point, one can understand that Afaan Oromo has five major grammatical categories serving as heads in phrase construction.

Each of these classes again can be divided into other sub-classes. For instance, noun class is categorized as proper noun, common noun and pronoun, and Preposition and postpositions are sub classes of ad- positions. The subclasses in turn can be divided into subclasses, and the subdivision process may continue iteratively depending on the level and aim of the investigation. The major Afaan Oromo word categories are: *noun, verb, adverbs, adjectives, pronouns and Adposition* [13].

4.4.1 Nouns

Nouns are names that are used to name or identify things, people, animals, places or abstract ideas. In Afaan Oromo most of the time a sentence begins with a noun which starts with capital letter and it uses a noun as a subject followed with subject markers. Direct object and indirect object also optionally follows the noun which is the subject of a sentence. For instance, the bolded words in the following sentences are all nouns:

Hoolaan marga dheeda. (The sheep grazes grass).

Bunni dinaggee keeynaa gudisa. (Coffee develops our economy).

Definiteness, Number, Gender and case Markers

Nouns are inflected to indicate different grammatical functions such as number, gender, definiteness and case. Inflectional suffixes are combined with stem usually resulting in a word of the same class as the original stem

Number: A singular is marked by zero morphemes where as a plural noun is marked morphologically by suffixing the morpheme like -oota, -oolii, -een, -lee, -wwan, -yyii, -eetii, -ii, -oo to the base as free alternates [11]. It is difficult to predict which suffix is for which noun, but there is a possibility of using all these suffixes as plural makers. Linguists agree that some groups of suffixes are most preferably applied to almost all nouns, and the others are used with only some words. According to Abebe [11], categorizing them according to universal usage, through identifying those attached to stems ending in specific consonantal phoneme and those that end in some group of phonemes.

Table 4:2: Categories of number indicator Suffix

Suffix	*Category*
-Oota, -oolee, -oolii	*Suffixes that delete the last vowel*
-wwan, -lee	*Suffixes that don't delete the last vowel*
-een –(a)n	*Suffixes that double last consonant*
-eeyyii	*Suffixes that drop –eessa/eensa*

Case: is a grammatical category of nouns that indicates the nature of their relationship to the verb in sentences [12]. The number of cases varies from language to language. In this regard, nouns in Afaan Oromoo are inflected for nominative, ablative, instrumental and locative cases. Nominative case is used for nouns that are the subjects of clauses whereas, instrumental is used for nouns that represent the instrument ("with"), the means ("by"), the agent ("by"), the reason, or the time of an event.

The locative is used for nouns that represent general locations of events or states, roughly at. For more specific locations, Afaan Oromoo uses prepositions or postpositions. Postpositions may also take the locative suffix. On the other hand, the ablative case is used to represent the source of an event; it corresponds closely to English from.

Table 4:3: Summary of Case makers

Cases	***formed with the suffix***
Nominative	*N*
Ablative	*dhaa, rraa*
Instrumental	*tiin, dhaan*
Locative	*-tti,*

Definiteness: demonstrative pronouns like *kun* (this), *sun* (that) are used to express definiteness. In some Afan Oromo dialects the suffix *-icha* for male and *-ittii(n)* for female and for undermining usually has a singularize function is used where other languages would use a definite article [7].

Afaan<u>icha</u>	*Afaan<u>ichi</u>*
Jaart<u>ittiin</u>	*Jaart<u>ittii</u>*
<u>Jaars</u>icha	*<u>Jaars</u>ichi*

4.4.2 Verbs

Verb is the most important part of a sentence that says something about the subject of a sentence, expresses an actions, events or states of being. In Afaan Oromo verb occurs in the final positions of a sentence. It is not the case that verbs constitute a distinct, open word class in all languages. In Afaan Oromo verbs are forms which occur in clause final positions and belong to a distinct category [1]. For instance, in each of the following sentences the verb is bolded:

Caalaan farda ***bite***. (Chala bought a horse)

Caaltuun ***barattuu*** dha. (Chaltu is a student) Leensaan ***dhufte***. (Lensa has come)

Verbs are morphologically the most complex POS in Afaan Oromoo, with many inflectional forms; numerous words with other POS are derived primarily from verbs. Generation of syntactically and semantically correct sentences requires appropriate choice among the different forms of verbs. There are two major criteria to identify verbs from other word categories: syntax and morphology. In the former case, verbs function as predicates in a simple sentence and they are found at the end of a sentence. In the latter case, the agreement of verb with the number, gender and/or person of the subject, proper case markers for the different nominal forms and expression

of the tense, aspect of the verb and number, specificity of the of nouns are some of the important morphological constraints governing correct generation.

An Afaan Oromoo verb consists minimally of a stem, representing the lexical meaning of the verb, and a suffix, representing tense or aspect and subject agreement. For example, in dhufne 'we came', dhuf- is the stem ('come') and -ne indicates that the tense is past and that the subject of the verb is first person plural. As in many other Afro-Asiatic languages, Afaan Oromoo makes a basic two-way distinction in its verb system between the two tensed forms, past (or "perfect") and present (or "imperfect" or "non-past") [4]. Each of these has its own set of tense/agreement suffixes. There is a third conjugation based on the present which has three functions: it is used in place of the present in subordinate clauses, for the jussive ('let me/us/him, etc. verb', together with the particle haa), and for the negative of the present (together with the particle hin). For example, deemne 'we went', deemna 'we go', akka deemnu 'that we go', haa deemnu 'let's go', hin deemnu 'we don't go'. There is also a separate imperative form: deemi 'go (singular) [11].

4.4.3 Adverbs

Adverbs are words which are used to modify a verb, an adjective, another adverb, or a clause. Adverbs usually precede the verbs they modify or describe. An adverb indicates time, manner, place, cause, or degree and answers questions such as „how? ‟, „when? ‟, „where? ‟, and „how much? ‟. In the following examples, each of the italic words is an adverb:

Oboleessi koo *boru* deema. (My brother will leave tomorrow.) Boru (tomorrow) is an adverb.

Namichi *tasa* du'e. (The man accidentally died.) Tasa(accidentally) is an adverb.

4.4.4 Adjectives

An adjective modifies a noun or a pronoun by describing, identifying, or quantifying words. In Afaan Oromo an adjective usually follows the noun or the pronoun which it modifies. Some of Afaan Oromo adjectives are: hedduu, mara, kam, adii, qalla, tokko, kee, etc. In the following examples, the italic words are adjectives:

Gammachuun *qallaa* dha. (Gemechu is thin).

Konkolaatan Cala *adii* dha. (Chala‟s car is white).

4.4.5 Pronouns

Alike English, in Afaan Oromo pronoun can replace a noun or another pronoun. Pronouns are marked for number and gender. For example, pronouns like "ishee/isii" which means “she” is feminine (singular), "isa" which means 'he' is masculine (singular), "isaan” which means 'they' is plural and can be masculine or feminine and "nuyi" which means “we” is plural and can be masculine or feminine. We use pronouns to make sentences less cumbersome and less repetitive. Grammarians classify pronouns based on their functions and meanings in the sentence into several types, including the personal pronoun, the demonstrative pronoun, the interrogative pronoun, the indefinite pronoun, the relative pronoun, the reflexive pronoun, and the intensive pronoun [1].
For example:

Tolan dhufe. (Tola came.) Inni dhufe. (He came.)

	1st	*2nd*	*3rd*
singular	*Ani (I)*	*Ati (you)*	*Isa /Inni (he)* *Isii/ Ishee(she)*
Plural	*Nuti(we)*	*Isin (you)*	*Isaan / Jarri(they)*

4.4.6 Adposition

Adposition are traditionally defined as words that link to other words, phrases, and clauses and express spatial or temporal relations. Adposition is almost universal part of speech. It is a cover term for prepositions and postpositions. It is a member of a closed set of items that occurs before or after a complement composed of a noun phrase, noun, pronoun, or clause that functions as a noun phrase, and form a single structure with the complement to express its grammatical and semantic relation to another unit within a clause [7].
Some languages have either prepositions or postpositions, others have both and yet others have quite neither. For example, English has prepositions but Japanese has postpositions [7]. Unlike English and Japanese, Afaan Oromo has both prepositions and postpositions. As grammatical tools, Adposition marks the relationship between two parts of a sentence: characteristically one element governs a noun or noun-like word or phrase while the other functions as a predicate. Qotebula-dhaaf (for farmers) Ambo-tti(at Ambo) Saartuu-n (Sartu is) Numaa-f(for us) Gaara-

rraa(from above) Adpositions are traditionally defined as words that “link to other words, phrases, and clauses” and that “express spatial or temporal relations.”

4.5 Afaan Oromo Syntax

Basically, every language has standardized word orders in sentences. For instance, Abdi [42] stated that English and French languages have ‘Subject-Verb-Object’ (SVO) orders of words in their sentences. However, Afaan Oromo is different from these languages in its syntactic structure; it uses subject-object-verb (SOV) form which is similar to Amharic and Japanese languages [14]. Subject-verb-object (SOV) is a sentence structure where the subject comes first, and the object and the verb are second and third elements of a sentence respectively. For instance, in the Afaan Oromo sentence Qananisaan atileeti dha (Kenenisa is an athlete), Qananiisa (kenenisa)is a subject, atileeti (athlete) is an object and dha (is) is a verb. Afaan Oromo adjectives follow a noun or pronoun; their normal position is close to the noun they modify while in English adjectives usually precede the noun. For instance, nama cima (strong man), the adjective cima (strong)follows the noun nama (man).

There are different rules to word order in Afaan Oromo sentence construction understanding of this syntactic structure of sentence can help us to know the relationship between words which in turn leads us to categorize them correctly. For our work, we mainly focus on Main clause word order and Noun phrase word order. The points discussed in this section are based on [7] Main clause word order as we discussed Afaan Oromo uses subject-object-verb (SOV) format sentence construction in the main clause. But, the resulting sentence structure may vary as shown in the following cases. I. Subject + verb Daraartuun dhufte, (Derartu came.) Anaa dhufu. (Welcome.) II. Subject + complement (object or adverbial) + verb Calaan konkolaataan dhufe, (Chela came by car.) Beekaan mana baruumsaati dhufe. (Beka came from school.) III. Complement + verb Mirga ishee falmatte. (She defined her right.) Tokkummaa qabaadhaa. (Have a unity). Noun phrase word order the noun usually precedes the qualifiers. The qualifiers are arranged in the sequence: noun – adjective – possessive/demonstrative pronoun. The subject marker is added to the noun itself and to the qualifiers of the noun as the base form is repeated in all qualifiers of the noun. Odaan guddaan kun yoom dhaabbate? (When this Sycamore tree is planted?) Mucaan qaloon xiqoon kun cimtuudha. (This small thin boy is dynamic.)

CHAPTER FIVE: WORD SEQUENCE PREDICTION MODEL FOR AFAAN OROMO

5.1 Overview

This Chapter presents details about Afaan Oromo Word Sequence Prediction Model. Specifically, the Architecture of the Afaan Oromo Word Sequence Prediction Model and its components with their respective algorithms are described. To build a language model, N-gram statistical model was employed. Accordingly, to prepare corpus for language modeling and identify observable artifacts which is useful to set expectations in the model development, a preliminary analysis was conducted. Finally, various N-gram models were applied to extract most probable stem words, and morphological features like tense, case, number, gender and person.

5.2 Preliminary Analysis of Corpus

A preliminary study of the corpus is necessary to prepare corpus for statistical modeling. Thus, preliminary analysis was conducted to recognize the relationships between corpus genre, vocabulary size, distributions of various N-Grams and other observable artifacts which is useful to set expectations in the model development. Moreover, to support selection and adoption of method for language model with justification. Thus, to explore the effect of morphological information on word sequence prediction, the preliminary analysis is conducted on training corpus. Initially, the preliminary analysis was conducted to explore the distribution of word in the documents. Thus, the summary of the analysis shows that the occurrence of a word in corpus extremely rely on corpus genre. Figure 5.1 below shows occurrence and frequency of sample word in corpus document. For instance, the word “nama” occur 497 times in the Bible (Wangella.txt), whereas it occurs only once in news document “Guraandhala.txt”. Consequently, if the language model is trained on Bible corpus and tested on news document, the model will favor the word “nama” to be the next word that users intended to type.

	Dramaa.txt	Guraandhala.txt	Qeeqa.txt	VOA.txt	Wangeela.txt	a.txt	abc.txt	abcdfe.txt
nama	3	1	3	58	497	46	31	6
yakka	1	0	0	1	12	1	0	0
hojjette	0	0	0	1	2	0	0	0
manni	0	0	0	10	6	0	1	0
murti	0	0	0	0	0	0	0	0
gaaffii	0	0	1	7	59	4	2	0
fi	2	27	93	289	0	0	0	0
deebii	3	0	2	15	116	9	5	2
qindeessinee	0	0	0	1	0	0	0	0

Figure 5-1:Frequency of sample word in corpus

On the other hand, the result of the analysis show that some words occur frequently in a particular document and might not occur at all in some other documents which have different genre. Figure 5.2 shows that the occurrence of “nama” and “seera” in diversified documents.

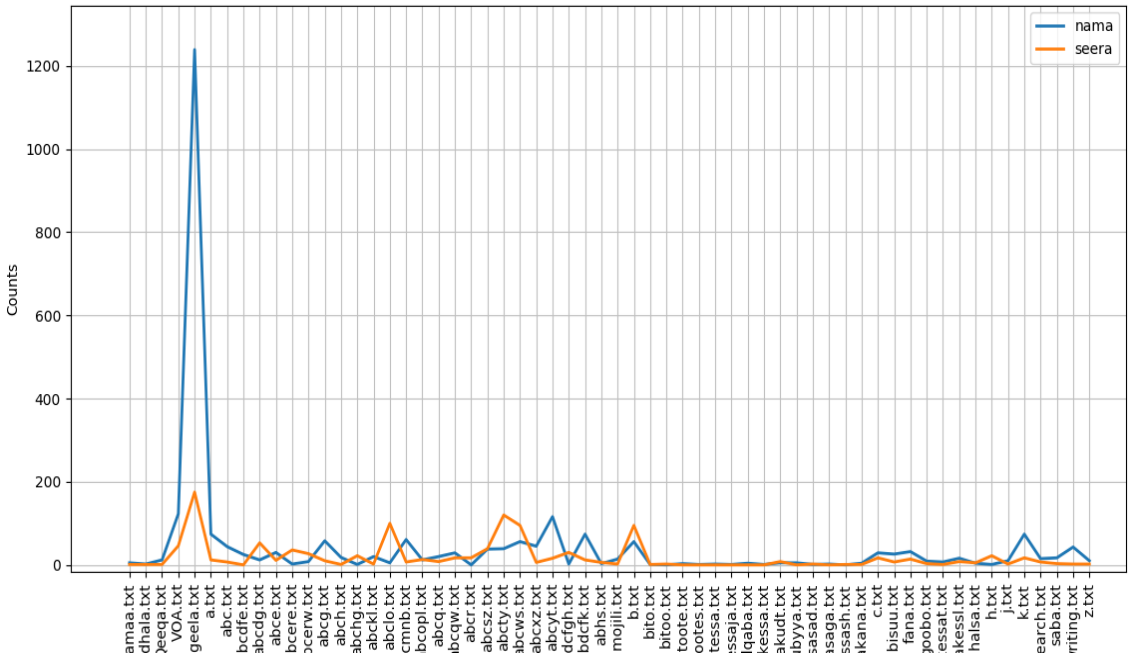


Figure 5-2: Occurrence of "nama" and "seera" in diversified documents

The other purpose of the preliminary analysis was to uncover the relationship between vocabulary size and distribution of various N-grams. Particularly, it helps to identify constraints to predict appropriate word in a given sequence regarding with providing the word that users intended to type and word that is grammatically acceptable in give a sequence. First, we looked into the *1*-grams, or unigrams models. After preprocessing and transformation the corpus has 69,750 distinct words representing total occurrences of 507,776 words in the whole corpus. Figure 5.3 shows the most frequently appearing words. The most commonly appearing word is "akka |as", appearing 10,000 times, or 1.97% of the total word count in **word forms.** This means, given any word, the unigram language model will predict the word "akka" every 1.967% times. The unigram counts give us an indication of the probability of each word occurring irrespective of any other information. If we have no other information at all, we cannot do better than guessing "akka" to be the next word in every instance.

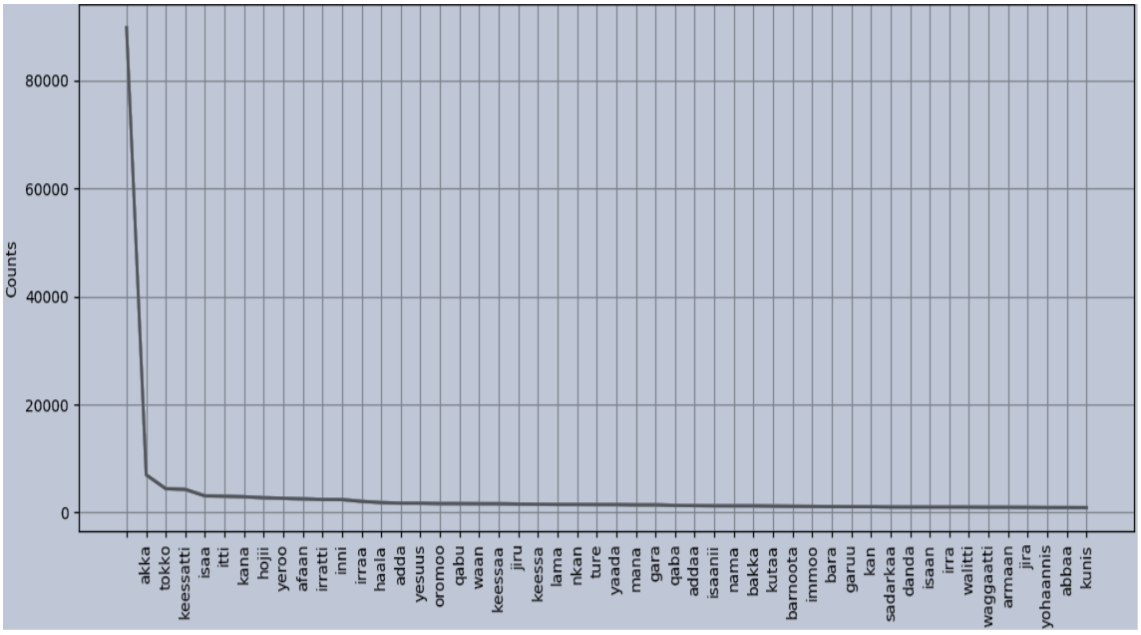


Figure 5-3:Most frequently appearing words

As a result, unigram model predicts inappropriate word for a given word sequence. This means, for any sequence of words the model predicts that the next word is the most frequent word in corpus. Consequently, the competence of the language model is limited to the existing corpus genre since the occurrence and frequency of words extremely rely on corpus genre. On the other hand, the result of above analysis shows that unigram model is inadequate to consider context information such as preceding word.

Secondly, the analysis is conducted on bigram model to examine the competence of the language model. Figure 5.4 shows the top 50 bigrams. The bigram "Afaan oromo" occurs 800 times out distinct 313,303 bigrams count. That means, given the word 'Afaan' the model predicts the next word as "oromo" with 1.14 %. Accordingly, bigrams count that were observed but which did not meet the minimum frequency threshold, were removed from prediction list. However, using Maximum Likelihood Estimator (MLE) the relative frequency of the bigram (bigram frequency/total bigram frequency beginning with current word) would reduce this partiality.

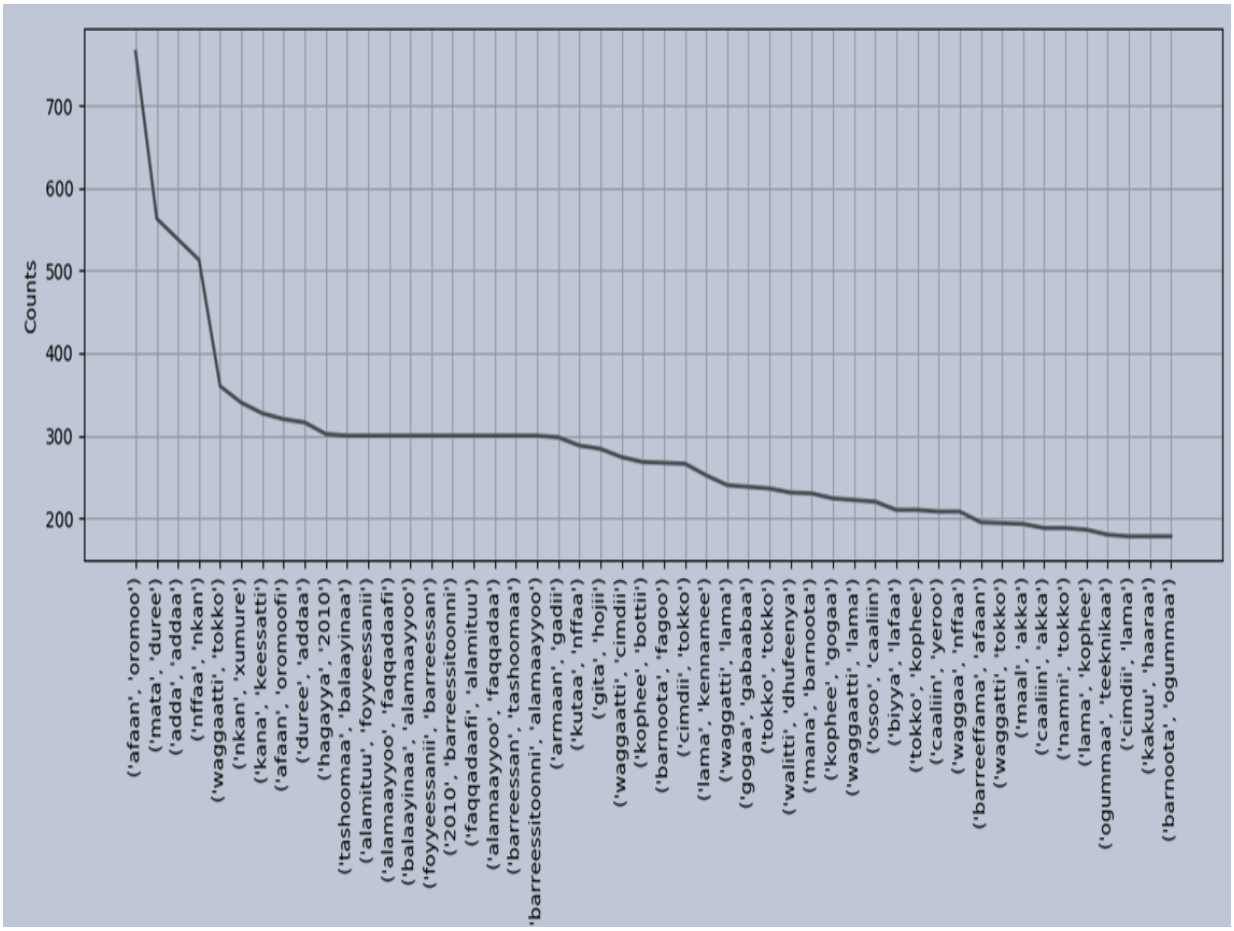


Figure 5-4: The top 50 Bigrams

Thus, Bigram model compute a bigram probability of a word *w2* given a previous word *w1*, by computing the count of the bigram *C* (*w1 w2*) and normalize by the sum of all the bigrams that share the same first word *w1*. since the sum of all bigram counts that start with a given word *w*1 must be equal to the unigram count for that word *w*1.

$$P(w2|w1) = \frac{c(w1, w2)}{c(w1)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5.1)$$

For example: Probability of the word “Oromo” given previous word is “Afaan” is calculated as shown below:

$c(\text{Afaan}, \text{oromo}) = 789, c(\text{Afaan}) = 2535$

$$P(\text{oromo}|\ \text{afaan}) = \frac{\text{C(afaan oromo)}}{\text{C(oromo)}}$$

$$P(\text{oromo}|\ \text{afaan}) = \frac{\text{C}(789)}{\text{C}(2535)} \quad = \quad 0.311$$

In fact, though, the language model predicts the word “oromo” given word “Afaan” which is grammatical acceptable, but it is not very interesting, since every time the word “Afaan” is given probably 31% it is the word oromo that as to be predicted. This means if user requires typing the word “Afaan” as other context as “mouth” the language model predict next word “oromo” that describe about language, unfortunately the context it intended for is to describe mouth. Thus, the language model is not context sensitive. It doesn’t consider on which context to predict next word rather it only relies on frequency count of words in corpus. In other word, the model favors the word “oromo” to be predicted and disfavor other words even if they sound well according to the context. For instance, the full space of possibilities is the given word with the size of the dictionary 69,750 words, they model only accommodate and assign probability out of 69% for all possible and impossible words sequence. The dictionary matrix below depicts sample of word count that assign with zero probability though they are acceptable or possible sequence of word they form when bounded.

Table 5:1: Dictionary matrix of sample word count

	Oromoo	Saba	Hoji	Mana	Murti	Isaa	Qamaa
Afaan	789	0	68	0	0	7	0
Oromoo	0	0	0	0	0	0	0
Saba	13	0	0	0	0	0	0

To address the problem of Zero probability, it is essential to understand the structure (syntax) and context of words in sentence. Since Zero probability could be ether structural zero (unacceptable sequence of word) or contingency zero. Hence, identifying zero probability whether it is structural zero or contingency zero is appropriate. N-gram language model would assign zero probability for two words which have not structural dependency. For example, the word “saba” and “murti” have no structural dependency. In other way of expression, in Afaan oromo there is no probability where the word “saba” is followed by word “murti”. Thus, we cannot find a sequence of “saba murti” in the corpus. Hence, the language model assigns zero probability for “saba murti” sequence. Even though, structural zero is assigned for grammatically not accepted word sequence, it is difficult to find all possible sequence in corpus. The major cause of zero probability is that corpus vocabulary is limited, so some perfectly acceptable word sequence may be missed from it. Thus, it is

contingency zero. This means that the *N*-gram matrix for any given training corpus is certain to have a very large number of zero probability that should certainly have some non-zero probability. For example, “Afaan isaa” is possible sequence. In Addition, a list of unique words that found on the test set may not exist or recognized in training set or unknown words. Thus, out -of -vocabulary is other cause of zero probability. In fact, literatures recommend handle unknown word in the test by adding a pseudo-word called <UNK> in training set and by training the model to compute the probability of <UNK> as any other regular word.

As a result, bigrams rely on frequency of words rather than context and grammatical rules that impose a heavy cognition load on the user to choose appropriate word. Besides, it also decreases the writing rate.

In fact, when the size of n or number of previous word that the model considers in predicting the next word increase the model accuracy would increase. That means, the language model would consider the context when number of previous word model consider increase. Figure 5.5 below shows the top 50 trigram sequence. The trigram “ffaa kan xumuree” occurs 350 times, which is given “ffaa kan” next word that the model predict would “xumuree “with the probability of the 0.25 trigram count. The phrase “5 ffaa kan xumuree” sound well than phrase predicted by bigram. In Addition, the word “Xumuree” is appropriate in the context of the previous two words. However, like bigrams, there are number possible word sequence which are not observed in corpus. As a result, it is a very sparse matrix.

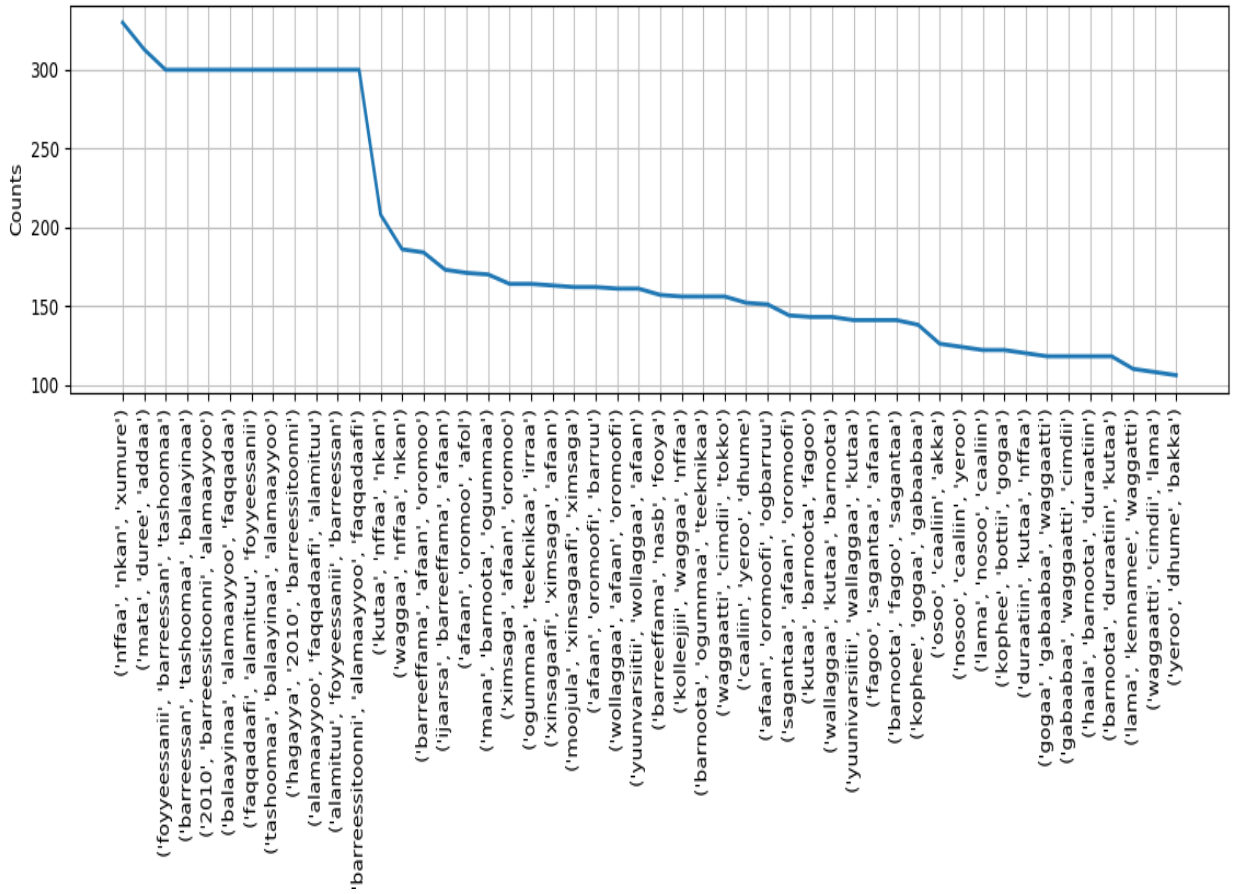


Figure 5-5:The top 50 Trigram sequence

Generally, the preliminary analysis indicated that N-gram language model alone is insufficient for Afaan oromo word sequence prediction since linguistic information and proficiencies has long distance syntactic dependencies. Accordingly, Afaan oromo cannot be modeled by a Markov chain because of data sparsity affect the n-gram model. On the other hand, the probabilities of an *N*-gram come from the corpus it is trained on, and training corpus is not expected always to ensure that each valid word sequence appears in corpus relevant number of times.

5.3 Morphological Analysis of corpus

As it has been frequently described in the previous chapters, Afaan oromo language has large dictionary of word forms with several morphological features produced from a stem and a set of inflection rules. To this end, in dealing with word sequence prediction for Afaan oromo language, a module that analyzes words in training data is essential to identify stem form and component morphemes. Thus, to build a tagged corpus that used for constructing statistical language models, Hornmorph was used for this study. Stem form and others morphological information that are useful for annotating each word in the corpus are extracted by Hornmorph morphological analyzer to build a tagged corpus. Consequently, out of 507,776 words of the corpus presumably 69% is analyzed by Hornmorph.

On the other hand, in Afan Oromo for a single verb root form over hundred valid word forms can be formed [12]. For instance, from a single verb stem ***—beek-*** ‖ over 216 valid word forms are generated by morphological generator developed by Abebe. Therefore, it is impractical to store all word forms for language model. For instance, if we have 10,000 verb and noun forms are identified and extracted from corpus, over 2 million words will be formed. Thus, the training corpus is pre-processed to hold only the stem form and selected morphological features of words. The morphological features and word formation method identified in chapter four was analyzed using Hornmorph. Thus, morphologically analyzed training corpus or tagged corpus consisting only stem form, cases, tense, number, person and gender is constructed. Accordingly, morphological information is tagged for words which their part of speech either verb or noun. Thus, tagged training corpus also contain words with no morphological information. Words with morphological information stored in tagged training corpus in respective with stem form, case, tense, number and. person. However, some words which are neither verb nor noun and Hornmorph unable to analyze are taken as it is, to keep consistency of stem word sequences. Figure 5.6 below shows the sample tagged training text analyzed using Hornmorph and constructed using python code.

Figure 5-6:The sample tagged training text constructed using python code.

'waashingitan', 'diisii', 'aanaa', 'kolfee', 'qaraaniyoo', 'naannoo^noun^0^0^0^sig^0^0', 'raphii', 'kan', 'jir^verb^0^0^0^pru^0^3', 'iddoo^noun^dat^0^0^sig^0^3', 'kosii^noun^abl^0^0^sig^0^3', 'gat^verb^abl^0^0^pru^0^3', 'gatam^verb^abl^0^0^pru^0^3', 'gat^verb^abl^0^0^pru^0^3', 'qoshee', 'jedham^verb^0^0^0^pru^0^3', 'voa-f', 'kaleessa', 'ibs^verb^0^0^0^pru^0^3', 'jiru', 'mana', 'isaanii', 'lafa^noun^bs^0^0^sig^0^0', 'kosii^noun^abl^0^0^sig^0^0', 'guutume', 'irratti', 'ijaaram^verb^0^0^0^0^0^3', 'ijaar^verb^0^0^0^0^0^3', 'jedhu', 'dura', 'sagalee^noun^abl^0^0^sig^0^0', 'dhoinsaa', 'dhagaamee', 'namee^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0', 'hundi', 'biyyoo^noun^bs^0^0^sig^0^0', 'dhidhimuu', 'dubbadh^verb^0^0^0^0^0^3', 'tuullaa', 'kosii^noun^abl^0^0^sig^0^0', 'jig^verb^abl^0^0^sig^0^3', 'illee^noun^bs^0^0^sig^0^3', 'reeffa^noun^bs^0^0^sig^0^3', 'baafam^verb^bs^impre^0^pru^0^2', 'jir^verb^bs^impre^0^pru^0^3', 'ministarri', 'dhimma^noun^bs^0^0^sig^0^0', 'koomiyunikeeshinii', 'mootummaa', 'itiyoophiyaa', 'obbo', 'negerii', 'leencoo', 'kaleessa', 'voaf', 'ibs^verb^0^0^0^pru^0^3', 'ibsa^noun^bs^0^0^sig^0^3', 'kennaaniin', 'lakkoobsa^noun^sb^0^0^sig^0^0', 'namee^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0', 'dhum^verb^sb^0^0^pru^0^3', 'gauu', 'dubbadh^verb^0^0^0^0^0^3', 'tur^verb^0^0^0^pru^0^3', 'maddi-oduu', 'rooyiters', 'immoo', 'poolisii^noun^bs^0^0^sig^0^0', 'irraa', 'odeeffannoo^noun^abl^0^0^sig^0^0', 'isaa', 'namee^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0', 'dhum^verb^sb^0^0^pru^0^3', 'dhuma^noun^bs^0^0^sig^0^3', 'gauu', 'gabaaseee', 'namoota', 'dhum^verb^0^0^0^pru^0^3', 'dhuma^noun^bs^0^0^sig^0^3', 'kanneen', 'kurna', 'hed^verb^0^impre^0^pru^0^2', 'hedduu^noun^dat^impre^0^sig^0^2', 'dh^verb^dat^impre^0^sig^0^3', 'dubartii^noun^abl^impre^0^sig^0^3', 'ijoolee', 'dha', 'oduu', 'dhaqqabeen', 'immoo', 'sabaa-himaan', 'mootummaa', 'gabaasetti', 'lakkoobsa^noun^sb^0^0^sig^0^0', 'namee^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0', 'duanii', 'gaee', 'jira', 'kan', 'reeffa^noun^0^0^0^sig^0^0', 'ijoollee^noun^0^0^0^sig^0^0', 'ijoo^noun^dat^0^0^sig^0^0', 'ijoollee^noun^dat^0^0^sig^0^0', 'ijoo^noun^dat^0^0^sig^0^0', 'isaanii', 'jaaa', 'baafadh^verb^0^0^0^0^0^2', 'guddaa^noun^bs^0^0^sig^0^0',

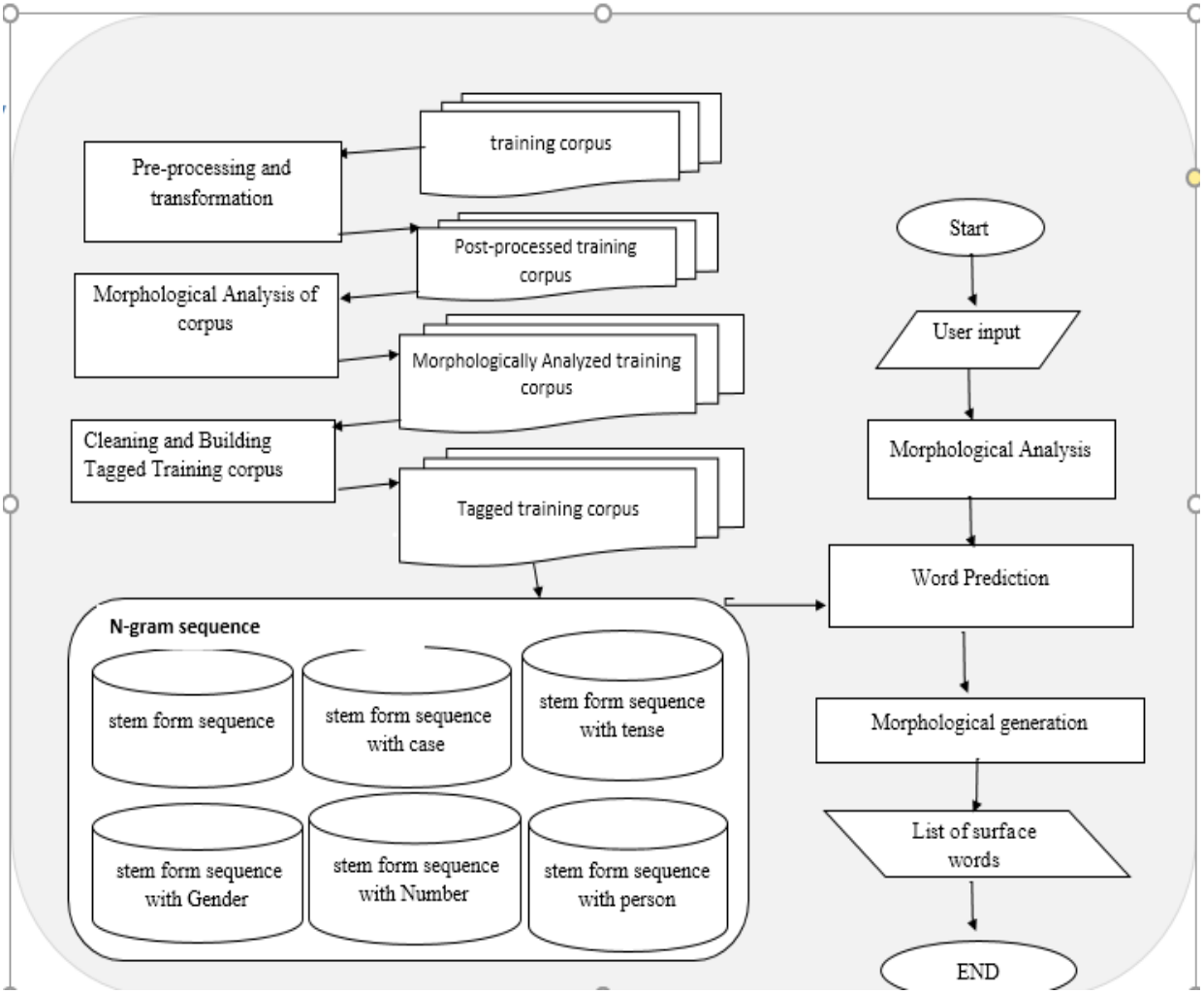
Figure 5-7: Algorithm to Build a Tagged Corpus

```
BEGIN
INPUT training-corpus
ANALYZE training-corpus using Hornmorph and WRITE in analyzed-corpus
INITIALIZE Dictionary with keys stem, POS case, tense, number, Person, gender,
INITIALIZE LIST morphological feature, New-Word, new-Word2 to FALSE

FOR line in training corpus:
        Add line in to list
        FOR each word in the list
        IF word is in new-Word keyword and new-Word2 is FALSE
                SET new-Word to TRUE
        ELSE IF new-Word is TRUE
                New-Word=FALSE
                New-Word2=TRUE
                root=word
        ELSE IF new-Word is TRUE and word is in case Key:
                case=word
        ELSE IF new-Word is TRUE and word is in Tense Key:
                tense=word
        ELSE IF new-Word is TRUE and word is in POS Key:
                POS=word
        ELSE IF new-Word is TRUE and word is in Number Key:
                number=word
        ELSE IF new-Word is TRUE and word is in person Key:
                person=word
        ELSE IF new-Word is TRUE and word is in gender Key:
                Gender=word
        ELSE IF word in new-Word key and new-Word2 is TRUE
              WRITE (root+'^'+POs'+'^'case +'^'+tense +'^'+number+'^'+ gender+'^'+person)
              on tagged-training-corpus
              SET new-Word2 to FALSE and new-Word to TRUE
OUTPUT tagged-training-corpus
END
```

5.4 Architecture of Afaan Oromo Word Sequence Prediction Model

Figure 5-8:Architecture of Afaan oromo word sequence prediction model



The Afaan oromo word sequence prediction model has two major components; N-gram sequence and prediction engine. N-gram sequence provides statistical information captured from tagged training corpus to prediction engine. Accordingly, prediction engine uses this statistical information to predict most probable stem form of words and morphological features of proposed root or stem words. Finally, the morphological generator produces surface words.

The prediction process is initiated after users enter one word. Accordingly, a user's input is accepted and analyzed using Hornmorph. Subsequently, stem and morphological features of words are extracted so that the word prediction engine uses this information to propose probable stems forms by interacting with the language model. Finally, the morphological generator produces surface words for proposed stem forms according to probable features predicted by language model. Figure 5.7 shows the Architecture of Afaan oromo word sequence prediction model.

5.5 Language Models

As described in preliminary analysis, it is the language model that gives a word sequence predictor a competence to predicted the next word. The Language model empowers the prediction engine by providing statistical information captured from tagged training corpus. Accordingly, to predict appropriate word for given sequence the statistical information of word sequence and their morphological features extracted and modeled using n-gram models.

The word sequence prediction task is accomplished in three phases. In phase one, stem form of words is suggested using stem n-gram models. In the next phase, morphological features of proposed stem words are predicted using statistical methods as well as linguistic rules to ensure word formation. Finally, the proposed stem word and morphological features are used by morphological synthesizer to generate appropriate surface words. Thus, phase one and two are two major tasks accomplished by language model that constructed from stem word sequences and morphological features using n-gram model trained on tagged corpus. Furthermore, as it is indicated at the beginning of this chapter, the accuracy of word predictor improved as n in the n-gram model increases due to suggesting words with more context information. However, as n in the n-gram increased the complexity and data size will simultaneously grew causing exponential response time. Consequently, we decided to use bigram, trigram and hybrid of bigram and trigram models using back-off algorithm.

Stem Forms Sequence

Bigram and Trigram statistical models are constructed for stem words sequence using the training corpus. Each n-gram model is separately stored on its own repository and they hold stem word sequences for each value of n with their probability of occurrence in the corpus. Probabilities of all unique stem word sequences with this respective value of n is calculated by counting occurrence of n stem word sequences and n-1 stem word sequences in the corpus where n is 2 for bi-gram and 3 for tri-gram models, and then calculating their ratio using MLE. Bi-gram and tri-gram probabilities are computed using equation (5.1) and (12) respectively.

$$P(w2|w1) = \frac{c(w1\,w2)}{c(w1)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

where, w1, w2 are stem words, P(w2|w1) is probability of a word w2 given w1, c(w2w1) is frequency of word sequence w2w1 in a corpus, c(w1) is frequency of w1 in a corpus.

$$P(w3|w2,w1) = \frac{c(w1w2w3)}{c(w1w2)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.2)$$

where, w1, w2, w3 are words, P(w3|w2w1) is probability of a word w3 given w2w1 previous words, c(w3w2w1) is frequency of word sequence w3w2w1 in a corpus, c(w2w1) is frequency of w2w1 in a corpus. Figure 5.9 shows the Algorithm to Construct n-gram Models. Sample trigram stem sequence extracted by python code is attached on Annex 6.

Figure 5-9: The algorithm for constructing Bigram and Trigram model

BEGIN
INPUT tagged training corpus
INITIALIZE LIST
INITIALIZE DICTIONARY
READ value of N
FOR each item in a tagged training corpus:
    Extract each sentence by <EOS>
    FOR each item in a sentence:
        SPLIT item by "^" and Access the value of key stem from dictionary
        EXTRACT stem form and ADD to LIST
    FOR each item in LIST:
        EXTRACT N sequence from LIST using index of item
        ADD each sequence to LIST two
FOR each item in LIST two:
    COUNT number of its occurrence, and ASSIGN value to frequency
    ADD the frequency with their respective item to DICTIONARY
FOR each item in DICTIONARY:
    CALCULATE probability of N sequence of words by taking ratio of frequency - of N sequence words with N-1 sequence words
    WRITE probability with their respective sequences in stem form sequence file
OUTPUT -stem sequence n-gram probabilistic model
END

Stem Forms with Case

To extract grammatical information regarding with grammatical category of nouns to identify the relationship between noun and verb in sentence, the bigram model of stem words with their respective case is constructed. Bi-gram model of stem words with their respective case is constructed by extracting and counting occurrence of unique stem word with its case sequence. This model stores frequency of each stem word with its case. Case for noun can be dative, base, subject, ablative, instrumental and locative cases. The most frequent case for a stem word is used later when producing surface words. The figure below shows the Algorithm to constructing Bigram model form stem and case. Sample bigram sequence of stem with case extracted by python code is attached on Annex 5.

```
BEGIN
INPUT tagged-training-corpus
INITIALIZE LIST
INITIALIZE DICTIONARY
FOR each word in tagged-training-corpus:
        SPLIT each word by "^" and ADD each item to a LIST
        EXTRACT stem and case using the item having "0" and "1" index from the list,
        ADD to DICTIONARY
FOR each root-case-sequence in DICTIONARY:
        ASSIGN frequency=0
        IF stem-case-sequence is new
                COUNT stem-case-sequence and ASSIGN it to frequency
                WRITE stem -case-sequence and frequency in a file
OUTPUT case-with-case n-gram model
```

Figure 5-10: The algorithm for constructing bigram model of stem with case

Stem Words with Tense

In Afaan oromo verbs are morphologically the most complex POS, with many inflectional forms; numerous words with other POS are derived primarily from verbs. Generation of syntactically and semantically correct sentences requires appropriate choice among the different forms of verbs. On the other hand, verb has a long distance depends with subject. For instance, in simple sentence "innii kalessaa dhufee", here "innii" is subject that convey grammatical information such as second person and singular. "kalessaa" is adverb even with limited derivational form. "dhufee" is verb having number of inflectional forms and context information representing tense, case, number and gender that requires to agree with subject. Furthermore, most of the time in Afaan oromo, sentence is written in the form of SOV order, so to predict next word, specially verb, we need to consider the subject of the sentence. Thus, to extract the relationship between stem form and suffix that indicate tense. we constructed a bigram model of stem form with tense. Here, frequency of each stem word with its respective tense is constructed. Perfective, imperfective, gerundive, and imperative or jussive are possible tense categories. Based on this information, the most likely tense indicator suffix for a given stem will predicted. The figure below shows the Algorithm for constructing a bigram model of stem form with tense

```
BEGIN
INPUT tagged-training-corpus
INITIALIZE LIST
INITIALIZE DICTIONARY
FOR each word in tagged-training-corpus:
        SPLIT each word by "^" and ADD each item to a LIST
        EXTRACT stem and tense using the item having "0" and "2" index from the list,
        ADD to DICTIONARY
FOR each root-case-sequence in DICTIONARY:
        ASSIGN frequency=0
        IF stem-tense-sequence is new
                COUNT stem-tense-sequence and ASSIGN it to frequency
                WRITE stem -tense-sequence and frequency in a file
OUTPUT stem-with-tense n-gram model
```

Figure 5-11:The Algorithm for constructing a bigram model of stem form with tense.

Stem Words with other morphological information

As described previously, In Afaan oromo language there is a long-distance dependence between noun and verb. In addition, in Afaan Oromo verbs must agree with their subject. In other word, verbs take suffixes that indicate subject agreement, specifically one of the seven basics person/number/gender categories of the language: first person singular and plural, second person singular and plural, third person singular masculine and feminine, and third person plural. Accordingly, we constructed a trigram model stem form with morphological feature such as number, person, and gender. For stem forms with person, the trigram of stem forms with person constructed from tagged training corpus. First person, second person and third person are possible categories. Based on this information, the most likely person indicator suffix for a given stem will predicted and used later to generate appropriate word surface.

Finally, for number and gender the same model as stem forms with person is constructed and stored on their respective file. Masculine and feminine are possible categories for gender. Accordingly, the most probable gender indicator suffix for a given root or stem will stored to be used later by morphological generator. Similarly, the most likely number indicator suffix for a given stem will stored to be used later by morphological generator. Figure 5.12 below shows the algorithm to build bigram stem words with number.

```
BEGIN
INPUT tagged training corpus
INITIALIZE LIST
INITIALIZE DICTIONARY key root, Number
READ value of N
FOR each item in a tagged training corpus:
        Extract each sentence by splitting <EOS>
        FOR each item in a sentence:
                Access the value of root and Number key from dictionary
                EXTRACT root or stem form and Number
                ADD to DICTIONARY
        FOR each item in DICTIONARY
                If N==3:
                        EXTRACT items on index of item, item +1 and item+2
                        ADD sequence of item with root key value and last item Number key value to
                TRIGRAM DICTIONARY
                ELSE If N==2:
                        EXTRACT items on index of item and item +1
                        ADD sequence of item with root key value and last item Number key value to
                BIGRAM DICTIONARY
                ELSE If N==1:
                        EXTRACT item
                        ADD item with root and Number key value to UNIGRAM DICTIONARY
IF root-number-sequence in TRIGRAM DICTIONARY:
        COUNT number of its occurrence, and ASSIGN value to frequency
        ADD the frequency with their respective item to BACK OFF DICTIONARY
ELSE IF root-Number-sequence in BIGRAM DICTIONARY:
        COUNT number of its occurrence, and ASSIGN value to frequency
        ADD the frequency with their respective item to BACK OFF DICTIONARY
ELSE IF root-Number-sequence in UNIGRAM DICTIONARY:
        COUNT number of its occurrence, and ASSIGN value to frequency
        ADD the frequency with their respective item to BACK OFF DICTIONARY
FOR each root-Number-sequence in BACK OFF DICTIONARY:
        ASSIGN frequency=0
        IF root-number-sequence is new
                COUNT root-number-sequence and ASSIGN it to frequency
                WRITE root-number-sequence and frequency in a file
```

Figure 5-12:The algorithm for constructing Stem and Tense sequence

5.6 Morphological Analysis of User Input

This module analyzes words accepted from a user and extracts required morphological features. Context information like, number and person is captured from a user ‘s input to predict appropriate morphological features for the coming stem word. Hornmorph is used to analyze word inserted by the user. Thus, morphological features such as, gender, number, person, and stem form are stored on temporary list or dictionary.

5.7 Word sequence Prediction

Prediction module predicts the most probable stem words and their morphological features using language models. Thus, Prediction words has two components, the stem forms predictor and morphological feature predictor. The stem forms predictor predicts the most probable stem forms. This component will estimate the most probable stem forms by computing the probability of the user input word or words in stem words bi-gram, tri-gram and hybrid model. Bi-gram model predicts stem word based on previous single word from current position, whereas tri-gram predicts stem word based on preceding two words. Hybrid of bi-gram and tri-gram model predicts the next word by considering preceding one or two words. The hybrid model uses a back off algorithm.

Finally, the stem forms predicted produces a list of 20 most probable stem forms.

The second component, the morphological feature predictor will predict the probable morphological features for list of proposed stem forms produces by stem form predictor. This means, each stem forms in prediction list checked for the most frequent case, tense, number and person in the language model. Finally, the most frequent morphological features represented in a way that the morphological generator can understand it.

Stem Word Prediction

Morphologically analyzed user ‘s input and previously constructed stem words bi-gram, tri-gram and hybrid probabilistic models are used to propose suitable stem words. Here, n last root or stem words are fetched from analyzed user ‘s input, and then, top highly occurring 20 stem words following a given n stem words are extracted from the language model, where, n is 1 for bi-gram and 2 for tri-gram model. Figure 5.13 below shows the algorithm to predict stem word.

```
INPUT stem-word-model and user-input// bi-gram or tri-gram and user input
READ last n words from a user-input//n=1 for bi-gram and 2 for tri-gram
INITIALIZE stem-keyword
INITIALIZE stem-word to “”
ANALYSE the last n word using Hornmorph and WRITE it last-n-analyzed-input file READ last-n-analyzed-input file
FOR each word in the last-n-analyzed-input
        IF word is in stem-keyword
        CONCATINATE word with stem word
        READ stem words probability model
FOR each word-sequence in -stem-word-model
        SPLIT the word-sequence to n
         IF stem-word == (n-1)thword or CONCATINATE (n-2)th word with (n-1)th word
                IF size of proposed-stem-words list is <20
                        ADD the nth-word to proposed-stem-words list
        OUTPUT proposed stem-word list
```

Figure 5-13 :The algorithm to predict stem word

Morphological Feature Prediction

Proposed stem words and previously constructed stem words with case, tense, number, gender and person n-gram sequence are used to propose the most probable morphological feature. Here, each proposed stem word is checked for the most frequent case, tense, number, gender and person in the n-gram sequence. In addition, the proposed morphological information needs to be represented in a way that the morphological generator can understand it. We have used similar algorithm to predict case, tense, number, gender and person. Figure 5.14 show the algorithms to predict morphological features.

```
BEGIN
INPUT stem-with-case sequence and proposed-stem-word list
FOR each proposed-stem-word in the list
        FOR each stem-word in root-with-case sequence
                IF proposed-stem-word equals stem-word in the case sequence
                        READ case that comes with proposed-case-word
                                ADD case to proposed-case list
OUTPUT proposed-case list
END
```

Figure 5-14: The algorithms to predict morphological features

5.8 Morphological Synthesis

The proposed stem words with morphological features such as case, tense, number and person are used to produce surface word or word form. As it is frequently mentioned in different sections of this study, Hornmorph morphological synthesis is used to produce correct words based on the proposed stem and morphological features.

CHAPTER SIX: IMPLEMENTATION AND EXPERMENT

In this chapter, tools and data collection techniques that are used to develop prototype is presented. In addition, experiment is conducted to demonstrate the performance of developed word sequence prediction model presented.

6.1 Data collection and pre-processing

The corpus for constructing the language model is collected from web document using web scraping technique. A web scraper script to navigate and extract text data from html documents on web pages is written using python programming language. A python *urllib. request* and *bs4* libraries are used for navigating through web page links and to extract text data from html respectively. Finally, a 6MB text data extract from four websites; https://www.afanoromo.fanabc.com, www.voaafaanoromoo.com, www.gadaa.com and https://chilot.me/regional-laws/oromia-nrs-laws/ were stored on text file format. The sample python code for web scraping is attached on Annex 1.

Additionally, 89 PDF files around 54 MB of size were collected from Internet via Google search engine and converted to text file using PYPDF python library. After transformation and pre-processing including removal of pictures, tables, charts, figures and unnecessary texts the document size is reduced to 9.8 MB. The detail python code for transforming and preprocessing pdf document attached on Annex 7.

Finally, 11.8 MB raw text file is used to construct a language model. Before setting up corpus for language modeling it is essential to undertake several transformations and text pre-processing since the documents are collected from Internet. In general, 84, 952 sentences are used for constructing language model.

Accordingly, the strategies and methods for transforming and pre-processing data for developing a model to predict sequence of words were identified and adopted based on Jurafsky and Martin's [7] [47] framework and recommendation. Thus, it is mandatory to customize this framework as it comfortable to this study. Overall, the following tasks are identified and adopted for the purpose transforming and cleaning the corpus. These are:

- Sentence tokenization: sentences in corpus is tokenized by end of sentence marker (.!?) and replaced by <EOS>
- Wordform: stemming is conducted to generate stem forms.
- Punctuation: Even though, Jurafsky and Martin treat punctuation as a word, in this study all punctuation except apostrophe has been removed. Apostrophe (hudhaa) is stored as they are since it has special function in Afaan Oromo writing.
- Numbers: numbers are removed from the corpus based on the Jurafsky and Martin [7] intuition that numbers will not have a great impact on a predication model.
- Whitespace: this was not discussed directly by Jurafsky and Martin. The intuition is that whitespace has little to do with context and excess whitespace in a text are removed.
- various ASCI code, all single letter, website URLS and e-mail address is removed

Test data set

To evaluate the proposed model, a test set having total of 456 sentences is collected www.voaafaanoromoo.com. We could not conduct the experiment with more test data due to low response time of the predictor. However, we believe the sentences used are representative.

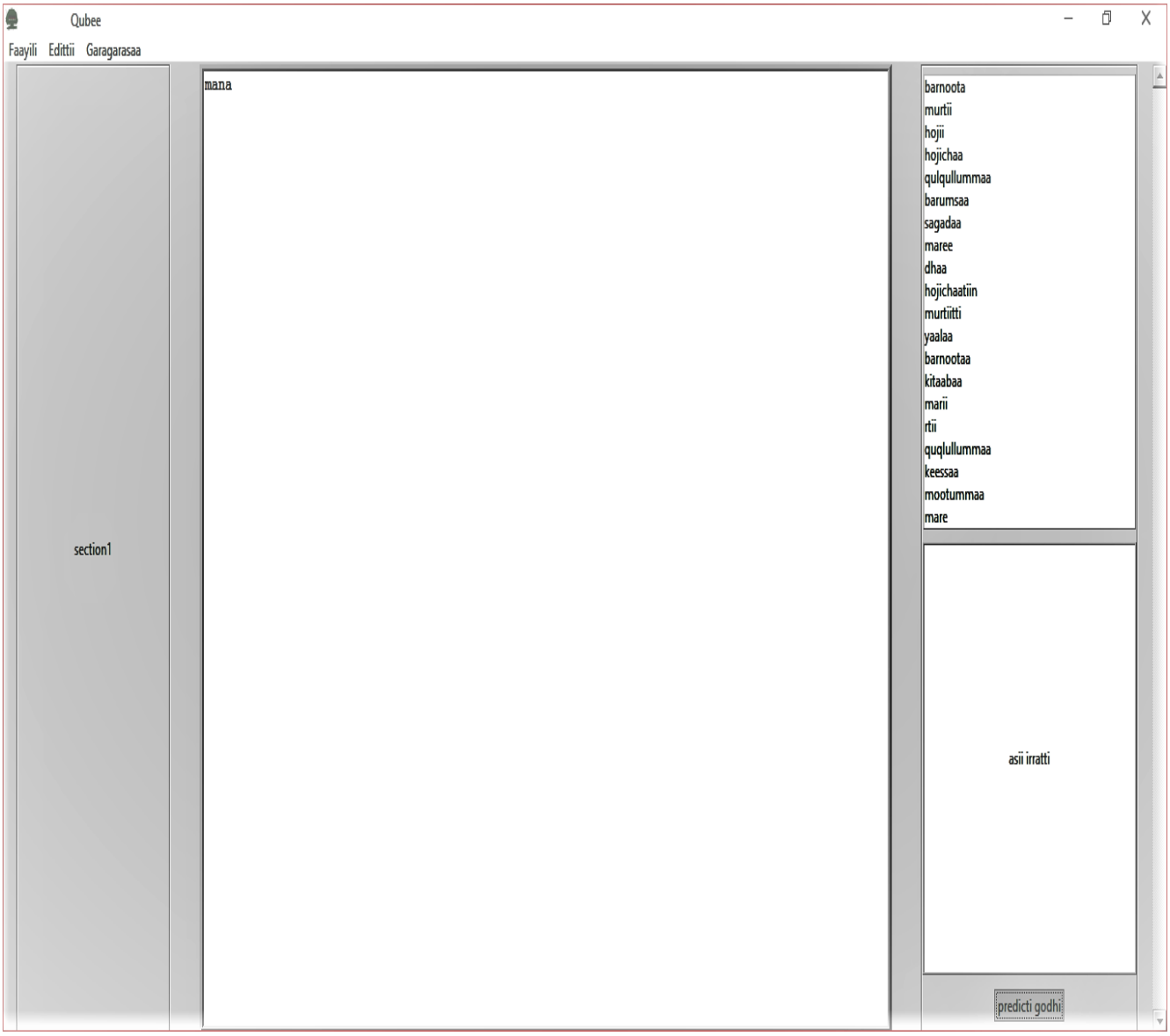
6.2 THE PROTOTYPE

In order to, evaluate the algorithms and make the necessary experiment on developed word sequence prediction models a prototype developed. The prototype has been developed using Python programming language. Figure 5.1 shows a user interface a protocol.

Figure 6-1:User interface for prototype

The prediction engine starts prediction task after users type one word in the text area and when space bar is pressed. Then the engine proposes the most probable twenty words and they displayed in a list box. Subsequently, a user clicks his or her preferred word from a given list of word options instead of typing each character. However, if the word that user require to type is not listed in a given option, then a user continues typing in normal way. Figure 6.2 shows a list of twenty most probable words generated by bigram model

Figure 6-2:List of twenty most probable words generated by Bigram model



6.3 Experiment

To evaluate the performance of the word sequence prediction models, the following experiments has been conducted. As the main purpose of this study is to examine the factor of morphological information on word sequence prediction accuracy, the experiment for this study was carried out on two models. Both, model one and model are trained and tested on the same training and test set. Thus, the first experiment is conducted on Model One, that has been developed on language model developed using N-gram model (bigram and trigram model) without considering morphological information. The other experiment had conducted on language model constructed from stem forms bigram and trigram model with morphological features such case, tense, person, gender, and number indictors.

Both models are evaluated based on prediction is accepted and appropriate if the proposed words are exactly as required by a user. Accordingly, for Model Two, even if, stem form of the proposed word is appropriate and the surface word formed by morphological generator is not exactly as user needed then prediction is assumed as inappropriate. Because, for this study we have no module to check the grammatical acceptance of word sequence. Hence, we assumed a perfect user who does not make typing mistake and picks the appropriate word right away when it is displayed in the list of word proposals. Accordingly, to simplify the evaluation task, we evaluate the competence of prediction model to write a sentence in test set. This means, give one or two words that sentence begin with the model will proposed list of twenty probable words, then if next word in sentence is in prediction list it is acceptable prediction.

Finally, the experiment conducted in this study evaluated based on obtained keystroke savings (KSS). KSS estimates saved effort percentage which is calculated based on (Eq.1.1) by comparing total number of keystrokes needed to type a text (KT) and effective number of keystrokes using word prediction (KE). Table 6.1 shows the summary of test result.

Table 6:1:Test result

Models		Total word predicted correctly	KT	KE	**KSS**
Language model without morphological features	Bigram	1,394	48,280	29,614	38.7%
	Trigram	1876	48,280	24,851	48.5%
	Hybrid of bi-gram and tri-gram	1975	48,280	18,986	60.7%
Language model with morphological features	Bigram	1919	48,280	19,347	60%
	Trigram	2129	48,280	15,593	67.7%
	Hybrid of bi-gram and tri-gram	2404	48,280	11,376	76.4%

6.4 Discussion

The experiment has revealed that Word sequence prediction using a Language model with morphological features optimize keystroke savings. As it can be seen from Table 6.2, the results of the experiments show the better keystroke savings attained using a Language model with morphological features. As a result, the result of the experiment shows that how morphological feature influence prediction task, even though it is difficult to draw a firm conclusion based on findings. In case of Model Two 60%, 67.7% and 76.4% keystroke savings is obtained using bigram, trigram and hybrid of bigram and trigram models. Even though, hybrid of bigram and trigram in model on perform better that bigram model of model two, the keystroke saving competence of Model Two dramatical increase for trigram and hybrid of bigram and trigram. In the case of model Two the keystroke saving obtained using bigram model is less compared with trigram and hybrid model. This is due to, even though the model predicts appropriate stem it is wrong wordform that it generates finally. Accordingly, we believe that the result in this work is promising and can be enhanced with addition of more linguistic resources in the language model.

CHAPTER SEVEN: CONCLUSION AND FUTURE WORK

7.1 Conclusion

The thrust of this study was to design Afaan Oromoo word sequence prediction model. Thus, we present a model that predict most probably based on statistical and morphological information of previous words. N-gram method was employed to construct a language model. In addition, morphological properties of Afaan Oromoo verb and noun have been extracted using Hornmorph morphological analyzer to develop language model from stem form with morphological feature such as tense, case, number, gender and person. Accordingly, the model set out to suggest the next word to be typed by a user in three phases. Firstly, most probable stem forms are predicted using language model. Secondly, morphological features are predicted for the proposed stem forms. Lastly, the proposed root or stem word and morphological features are used by morphological synthesizer to generate appropriate surface words.

To evaluate the performance of the word sequence model and support our conclusion with justification on how morphological features determine the accuracy of word prediction models we evaluate developed model by comparing it with a model that was developed without considering morphological feature. Accordingly, an experiment had been conducted based on Keystroke saving. Consequently, the result of the experiment indicated the better KSS is achieved with the model constructed from N-gram and morphological information. In conclusion, the developed model has potential advantages since an effective word prediction can be carried out using very large corpus size, statistically based techniques, and linguistic feature. We believe that application of this technology is ample. More importantly, it has capability to bring benefits of fast text typing to virtual keyboards to assisting people with disabilities.

7.2 FUTURE WORK

Word prediction system demands deep understanding of structural and semantic features of language under consideration. Hence, it seems that there is a plethora of gaps for improving and modifying Afaan oromo word sequence prediction. To this end, we strongly believe that this kind of study can be further investigated in numerous ways to optimize the task of Afaan Oromoo word sequence prediction. Accordingly, some directions for future work are suggested below.

- In this study, the language model built from stem forms, and morphological feature such as tense, case, person, gender and number are used. But, these morphological features are not sufficient. On the other hand, there is long-distance dependence between noun and verb. Thus, we recommend future studies on word sequence prediction to consider the long-distance dependence between words.
- In this work, when predicting features like, tense, gender and number for a given stem form, the first highly frequent feature is used, but it is not necessarily correct proposal. Therefore, we recommend considering other methods along with highest frequency to make more precise feature prediction in future research niche of this kind.
- Hornmorph program is a work in progress and it has some limitations. For instance, there are words that cannot be analyzed, wrongly processed or cannot be generated at all. In this work to keep the sequence of words, word which cannot processed by Hornmorph is taken as it is. Due to this, training done with wrong morphological analysis result brings erroneous prediction output. Therefore, it is suggested to employee other method for classifying words in their part of speech class is recommended to upgrade Afaan oromo word sequence prediction work.

- Even though, word sequence can be used as a component in spelling and grammar error correction system. Word sequence prediction requires a grammatical error detection and corrector modules. This resembles the adding, grammatically relation finder to capture grammatical relationship between words. On the other hand, In Afaan Oromoo spelling may result grammatical error. Accordingly, for future studies on word prediction system to reduce cognitive load and increase keystroke saving requires considering spelling and grammar issue.

- Word sequence prediction also requires the consideration of multi-words. Nevertheless, this study is limited to consider multi-words. Thus, it is recommended for future work of this kind to consider.

- In this work, Keystroke saving is used to evaluate the developed a word sequence model, However, there are also other evaluation metrics to be employed. Therefore, we suggest considering other evaluation metrics. For instance, grammatical acceptance of word in given sequence can be used as evaluation metrics.

REFERENCES

[1] P. Isokoski “Manual Text Input: Experiments, Models, and Systems” MSc thesis in Information science, University of Tampere: Tampere,2004

[2] Nesredien. S “word prediction for Amharic Online Handwriting Recognition” MSc thesis in Computer science, Addis Ababa University: Addis Ababa, Ethiopia July 2008

[3] P. Kristensson” Five Challenges for Intelligent Text Entry Methods,” journal of Association for the Advancement of Artificial Intelligence, 2009.

[4] Samuel and Johanna “Effect of dynamic keyboard and word-prediction systems on text input speed in persons with functional tetraplegia” JRRD, v 51 no3,2014 page467-480.

[5] Jacob, Wobbrock and Brad A. Myers” from Letters to Words: Efficient Stroke-based Word Completion for Trackball Text Entry” Information School, University of Washington Seattle, Washington USA,2012

[6] A. Sabbir Arif, and W. Stuerzlinger “Pseudo-Pressure Detection and Its Use in Predictive Text Entry on Touchscreens” York University, Toronto, Canada,2014

[7] Jurafsky and Martin, “Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition”, June, 2007

[8] Manning, “Foundation of Statistical natural language processing” MIT press, Cambridge

[9] Tigist Tesoun “Word Sequence Prediction for Amharic Language” MSc thesis in Computer science, Addis Ababa university, Addis Ababa,Ethiopia October 2014.

[10] Gudisa Tesema “Design and implementation of predictive text Entry method for Afaan oromo on mobile phone” MSc thesis in Electrical and computer Engineering, Addis Ababa university: Addis Ababa, February 2013.

[11] Debela Tesfaye “A rule-based Afan Oromo Grammar Checker” International Journal of Advanced Computer Science and Applications, Vol. 2, No. 8, 2011

[12] Abebe. A “Analysis of Rule Based Approach for Afan Oromo Automatic Morphological Synthesizer” Science, Technology and Arts Research Journal, 2013

[13] Getachew. M and Millon .M “Parts of Speech Tagging for Afaan Oromo”, International Journal of Advanced Computer Science and Applications Special Issue on Artificial Intelligence, 2012

[14] Charles A. MacArthur” using Technology to Enhance the Writing Processes of Students with Learning Disabilities” journal of learning Disabilities, volume 29, number 4, JULY 1996 Pages 344-354

[15] Keith. T, Debra. Y, J. McCaw “The Effects of Word Prediction on Communication Rate for AAC,” Department of Computer and Information Sciences University of Delaware Newark, DE 19716

[16] Hisham Al-Mubaid “Application of word prediction and disambiguation to improve text entry for people with physical disabilities,” Int. J. Social and Humanistic Computing, 2012

[17] Jack Hourcade, Elizabeth West and Phil Parette “A History of Augmentative and Alternative Communication for Individuals with Severe and Profound Disabilities,” focus on autism and other developmental disabilities volume 19, number 4, winter 2004

[18] Masood Ghayoomi *and* Saeedeh Momtazi” An Overview on the Existing Language Models for Prediction Systems as Writing Assistant Tools” Department of Computational Linguistics Saarland University, Saarbrucken, Germany,2009

[19] Michal Koutný “Word prediction using language models,” MSc thesis in computer science, Institute of Formal and Applied Linguistics, Prague 2012

[20] Sachin. A and Shilpa. A **“**Context based word prediction for texting or language”, school of computer science, Carnegie Mellon University, Pittsburgh 2014.

[21] Gaddisa Olani "Design and Implementation of Morphology Based Spell Checker", International Journal of Scientific & Technology Research Volume 3, Issue 12, December 2014.

[22] Keith Trnka and Kathleen McCoy, “Evaluating word prediction: framing keystroke savings”, In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 261-264, Association for Computational Linguistics, 2008.

[23] Steffen Bickel, Peter Haider, and Tobias Scheffer, “Predicting Sentences using N-Gram Language Models” Humboldt-University: Berlin

[24] Afsaneh Fazly, “The Use of Syntax in Word Completion Utilities” MSc thesis in Computer science, University of Toronto:2002

[25] Yangyang Shi, “Language Models with Meta Information”, MSc thesis in Mathematics: Southeast University, P. R. China Geboren te Yancheng, P. R. China.

[26] Stanley F. Chen and Joshua Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling” Aiken Computation Laboratory: Harvard University

[27] Ariya Rastrow, “Practical and efficient Incorporation of Syntactic Features into Statistical Language Models” PhD dissertation: Johns Hopkins University

[28] Carlo Aliprandi, Nicola Carmignani and Paolo Mancarella, “An Inflected-Sensitive Letter and Word Prediction System”, International Journal of Computing & Information Sciences Vol. 5, No. 2, August 2007, Pages 79 - 85

[29] Carlo Aliprandi and Nicola Carmignani “Advances in NLP applied to Word Prediction” Department of Computer Science – University of Pisa, Italy

[30] Hisham Al-Mubaid, “Learning-Classification Based Approach for Word Prediction” The International Arab Journal of Information Technology, Vol. 4, No. 3, July 2007

[31] Masudul Haque and Tarek Habib “Automated Word Prediction in Bangla Language Using Stochastic Language Models”, International Journal in Foundations of Computer Science & Technology

[32] Qaiser Abbas “A Stochastic Prediction Interface for Urdu” I.J. Intelligent Systems and Applications, 2015,

[33] Mahar and Memon “Probabilistic Analysis of Sindhi Word Prediction using N-Grams “Australian Journal of Basic and Applied Sciences · January 2011

[34] Kebede Hordofa, “Towards the Genetic Classification of the Afaan Oromoo Dialects”, University of Oslo: Department of Linguistics and Scandinavian Studies, 2009.

[35] Gregory W. Lesher and Bryan J. Moulto, “Effects of N-gram Order and Training Text Size on Word Prediction” Department of Communication Disorders and Sciences State University of New York at Buffalo, New York, U.S.A.

[36] Mohammed Hassen, 'A Brief Glance at the History of the Growth of Written Oromo Literature' in Cushitic and Omotic Languages 3rd, International Symposium, Berlin, 1996.

[37] Mandefro Legesse, Named Entity Recognition for Afaan Oromo, Master‟s Thesis, School of Graduate studies, Addis Ababa University, 2012.

[38] Tilahun Gamta, The Oromo language and the latin alphabet, Journal of Oromo Studies, 1992.

[39] Wakshum Mekonnen. 2000. Development of Stemming Algorithm for Oromo Texts. MA Thesis

[40] Mekonnen Hundie, “Thesis: Lexical Standardization in Oromo”, Department of Linguistics, AAU, 2002

[41] Diriba Megersa, “Thesis: An automatic sentence parser for Oromo language using Supervised learning technique”, Department of Information Science, AAU, 2002.

[42] Abdi Sani, “Afaan Oromo Named Entity Recognition Using Hybrid Approach” MSc thesis in computer science Addis Ababa university: Addis Ababa 2015

[43] Tesfa Kebede, “Word Sense Disambiguation for Afaan Oromo Language”, MSc thesis in computer science Addis Ababa university: Addis Ababa 2013

[44] Michael Gasser, “Hornmorph: a system for morphological processing of Amharic, Oromo, and Tigrinya”, In Conference on Human Language Technology for Development, Alexandria, Egypt. 2011

[45] Michael Gasser, Hornmorph User's Guide, 2012.

[46] Steven Bird, Ewan Klein, and Edward Loper, “Natural Language Processing with Python” O’Reilly Media, United States of America 2009

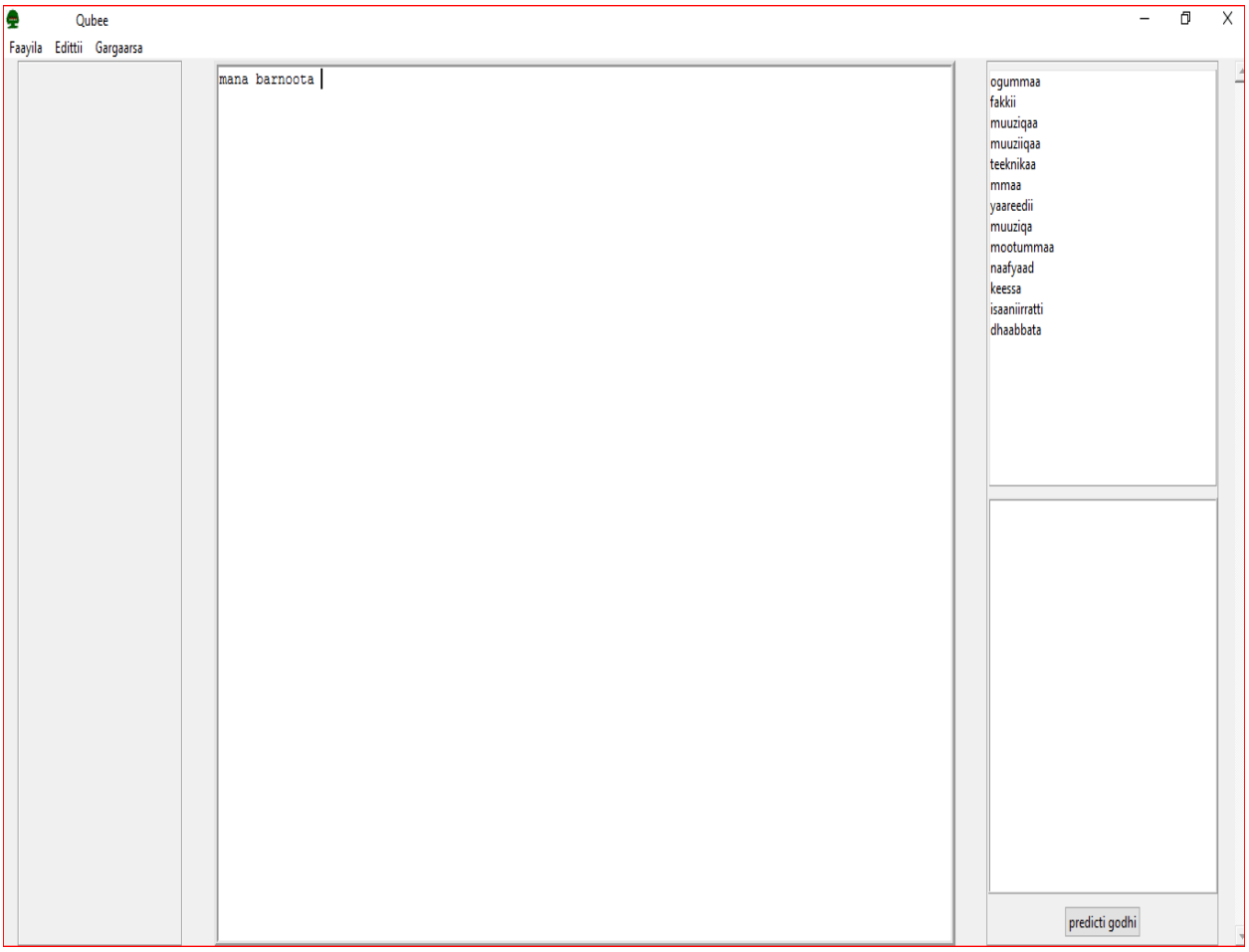
[47] Gerald R. and Gendron Jr. “Natural Language Processing: A Model to Predict a Sequence of Words”, Research Gate:2015

ANNEXES

ANNEX:1 A python code crawling particular web site

```
from urllib. request import urlopen
from bs4 import BeautifulSoup
import datetime
import random
import re
random.seed(datetime.datetime.now())
def getLinks(articleUrl):
    html = urlopen("http://www.voaafaanoromoo.com"+articleUrl)
    bsObj = BeautifulSoup(html)
    return bsObj.find("div", {"id":"bodyContent"}).findAll("a",
                href=re.compile("^(/voaafaanoromoo/)((?!:).)*$"))
links = getLinks("voaafaanoromoo.com/a")
while len(links) > 0:
    newArticle = links[random.randint(0, len(links)-1)].attrs["href"]
    print(newArticle)
    links = getLinks(newArticle)
```

Annex 2: A list of twenty most probable words generated by trigram model



Annex 3: A sample of word in tagged training corpus

voatagged - Notepad

File Edit Format View Help

```
['WAASHINGITAN', 'DIISII', 'Aanaa', 'Kolfee', 'Qaraaniyoo', 'naannoo^noun^bs^0^0^sig^0^0', 'Raphii', 'kan', '<^verb^0^0^0^0^0^1', 'iddoo^noun^ins^0^0^sig^0^1',
'kosii^noun^abl^0^0^sig^0^1', '<^verb^abl^0^0^sig^0^1', '<^verb^abl^0^0^sig^0^3', 'Qoshee', '<^verb^0^0^0^pru^0^3', 'VOA-f', 'kaleessa', '<^verb^0^0^0^pru^0^3',
'jiru', 'Mana', 'isaanii', 'lafa^noun^bs^0^0^sig^0^0', 'kosii^noun^abl^0^0^sig^0^0', 'guutume', 'irratti', '<^verb^0^0^0^0^0^3', '<^verb^0^0^0^0^0^1',
'<^verb^0^0^0^0^0^3', '<^verb^0^0^0^0^0^1', 'jedhu', 'Dura', 'sagalee^noun^ins^0^0^sig^0^0', 'dhoinsaa', 'dhagaamee', 'nama^noun^sb^0^0^sig^0^0',
'namee^noun^sb^0^0^sig^0^0', 'hundi', 'biyyoo^noun^bs^0^0^sig^0^0', 'dhidhimuu', '<^verb^0^0^0^0^0^3', 'Tuullaa', 'kosii^noun^abl^0^0^sig^0^0',
'<^verb^abl^0^0^sig^0^1', 'illee^noun^abl^0^0^sig^0^1', 'reeffa^noun^abl^0^0^sig^0^1', '<^verb^abl^0^0^sig^0^1', '<^verb^abl^0^0^sig^0^1', 'Ministarri',
'dhimma^noun^bs^0^0^sig^0^0', 'Koomiyunikeeshinii', 'Mootummaa', 'Itiyoophiyaa', 'Obbo', 'Negerii', 'Leencoo', 'kaleessa', 'VOAf', '<^verb^0^0^0^pru^0^3',
'ibsa^noun^bs^0^0^sig^0^3', 'kennaaniin', 'lakkoobsa^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0', 'namee^noun^sb^0^0^sig^0^0', '<^verb^sb^0^0^pru^0^3', 'gauu',
'<^verb^0^0^0^0^0^3', '<^verb^0^0^0^0^0^1', 'Maddi-oduu', 'Rooyiters', 'immoo', 'poolisii^noun^bs^0^0^sig^0^0', 'irraa', 'odeeffannoo^noun^bs^0^0^sig^0^0', 'isaa',
'nama^noun^sb^0^0^sig^0^0', 'namee^noun^sb^0^0^sig^0^0', '<^verb^sb^0^0^pru^0^3', 'dhuma^noun^bs^0^0^sig^0^3', 'gauu', 'gabaaseee', 'Namoota', '<^verb^0^0^0^pru^0^3',
'dhuma^noun^bs^0^0^sig^0^3', 'kanneen', 'kurna', '<^verb^0^0^0^0^0^2', 'hedduu^noun^bs^0^0^sig^0^2', '<^verb^bs^0^0^sig^0^3', 'dubartii^noun^bs^0^0^sig^0^3',
'ijoolee', 'dha', 'Oduu', 'dhaqqabeen', 'immoo', 'sabaa-himaan', 'Mootummaa', 'gabaasetti', 'lakkoobsa^noun^sb^0^0^sig^0^0', 'nama^noun^sb^0^0^sig^0^0',
'namee^noun^sb^0^0^sig^0^0', 'duanii', 'gaee', 'jira', 'Kan', 'reeffa^noun^0^0^0^sig^0^0', 'ijoo^noun^0^0^0^sig^0^0', 'ijoollee^noun^0^0^0^sig^0^0',
'ijoo^noun^0^0^0^sig^0^0', 'ijoollee^noun^0^0^0^sig^0^0', 'ijoo^noun^0^0^0^sig^0^0', 'isaanii', 'jaaa', '<^verb^0^0^0^0^0^3', '<^verb^0^0^0^0^0^1',
'<^verb^0^0^0^pru^0^3', 'isaa', 'qeeqan', 'Mootummaan', '<^verb^0^0^0^0^0^1', 'maashinii', 'biyyoo^noun^bs^0^0^sig^0^0', '<^verb^bs^0^0^sig^0^1',
'reeffa^noun^bs^0^0^sig^0^1', '<^verb^bs^0^0^sig^0^3', 'illee^noun^bs^0^0^sig^0^3', 'nuuf', 'hin', 'Maashinii', 'sanaaf', 'maallaqa^noun^bs^0^0^sig^0^0',
'kiisii^noun^bs^0^0^sig^0^0', 'kootii^noun^dat^0^0^sig^0^0', '<^verb^dat^0^0^sig^0^3', '<^verb^dat^0^0^sig^0^3', 'jedhu', 'Lafa', 'Aangaaas', '<^verb^0^0^0^0^0^3',
'kennaa^noun^dat^0^0^sig^0^3', '<^verb^dat^0^0^sig^0^1', 'jedhu', 'Ministarri', 'Dhimmoota', 'Koomiyunikeeshinii', 'Mootummaa', 'Itiyoophiyaa', 'Obbo', 'Nagarii',
'Leencoos', 'Amma', 'lubbuu^noun^bs^0^0^sig^0^0', 'carraaqqii^noun^bs^0^0^sig^0^0', 'guddaa^noun^dat^0^0^sig^0^0', 'gochaa^noun^dat^0^0^sig^0^0',
'gocha^noun^dat^0^0^sig^0^0', 'gochaa^noun^dat^0^0^sig^0^0', 'gocha^noun^dat^0^0^sig^0^0', 'gochaa^noun^dat^0^0^sig^0^0', '<^verb^dat^0^0^sig^0^1', 'Maal', 'jiginsa',
'lafa^noun^bs^0^0^sig^0^0', 'kosii^noun^abl^0^0^sig^0^0', 'sanaaf', 'sababaa', '<^verb^0^0^0^0^0^3', 'siachi', '<^verb^0^0^0^0^0^3', 'adda^noun^bs^0^0^sig^0^3',
'<^verb^bs^0^0^sig^0^1', 'Yeroo', 'beeksisna', 'jedhan', 'Namoonni', 'balaa^noun^bs^0^0^sig^0^0', 'irraa', 'lubbuu^noun^bs^0^0^sig^0^0', '<^verb^bs^0^0^pru^0^3',
'<^verb^bs^0^0^pru^0^2', 'hedduu^noun^bs^0^0^sig^0^2', 'mootummaa^noun^ins^0^0^sig^0^2', '<^verb^ins^pas^0^pru^0^1', 'tuullaa^noun^bs^pas^0^sig^0^1',
'kosii^noun^abl^pas^0^sig^0^1', 'gama-tokkotti', 'dhiibuu^noun^abl^0^0^sig^0^0', 'sii', 'sanaaf', 'sababaa', '<^verb^0^0^0^0^0^3', 'dubbatu', 'Dhaabbatni', 'Misoomaa',
'Tokkummaa', 'Mootummootaa', 'Yunivarsitii', 'Fininnee', '<^verb^0^impre^0^pru^0^2', 'jira', 'Sababaan', 'isaa', 'jira', 'Gabaasa', '<^verb^0^0^0^0^0^3',
'<^verb^0^0^0^pru^0^3', 'guutuu^noun^0^0^0^sig^0^3', 'dhaggeeffadhaa', 'Lakobsi', 'Namoota', 'Finfinnee', 'Keessatti', 'Jiginsa', 'Tuullaa', 'Kosiitiin', 'Dhumanii',
'Gauun', 'Beekame', 'Embed', 'code', 'Copy', 'and', 'paste', 'the', 'embed', 'code', 'below', 'The', 'code', 'has', 'been', 'copied', 'your', 'clipboard', 'Xurree',
'marsariitii', 'kbps', 'Taphachisi', 'WAASHINGITAN', 'DIISII', 'Prezidaantiin', 'yunaaytid', 'Isteetes', 'dura^noun^0^0^0^sig^0^0', 'duraa^noun^0^0^0^sig^0^0',
'Baraak', 'Obaamaa', 'filannoo^noun^ins^0^0^sig^0^0', 'prezidaantummaa', 'Amerikaa', '<^verb^0^0^0^0^0^3', 'baatii^noun^0^0^0^sig^0^3', 'Sadaasa',
'<^verb^0^0^0^0^0^3', 'ituu', 'hin', 'gaggeeffamiin', 'torbanneen', '<^verb^0^0^0^0^0^1', '<^verb^0^0^0^0^0^2', 'bilbila^noun^sb^0^0^sig^0^2', '<^verb^sb^0^0^sig^0^1',
'<^verb^sb^impre^0^pru^0^2', '<^verb^sb^impre^0^pru^0^1', '<^verb^sb^impre^0^pru^0^2', '<^verb^sb^impre^0^pru^0^1', 'iccitii^noun^sb^impre^0^sig^0^1',
'dhaggeeffatamu', '<^verb^0^0^0^pru^0^3', 'ajaja^noun^bs^0^0^sig^0^3', 'Prezidaant', 'Traamp', '<^verb^0^0^0^pru^0^3', '<^verb^0^0^0^pru^0^3', '<^verb^0^0^0^pru^0^3',
'<^verb^0^0^0^pru^0^3', 'himata^noun^bs^0^0^sig^0^3', 'suun', 'ragaa^noun^bs^0^0^sig^0^0', 'hin', '<^verb^0^0^0^0^0^1', 'hara^noun^bs^0^0^sig^0^1',
'<^verb^bs^0^0^sig^0^3', 'jira', 'Hogganaa', 'sadarkaa^noun^abl^0^0^sig^0^0', '<^verb^abl^0^0^sig^0^1', 'olaana^noun^ins^0^0^sig^0^1', 'olaanaa^noun^bs^0^0^sig^0^1',
'olaana^noun^ins^0^0^sig^0^1', 'olaanaa^noun^bs^0^0^sig^0^1', '<^verb^bs^0^0^sig^0^1', 'koree^noun^dat^0^0^sig^0^1', '<^verb^dat^0^0^sig^0^3',
'tika^noun^bs^0^0^sig^0^3', 'nagaa^noun^abl^0^0^sig^0^3', 'kan', 'mana^noun^bs^0^0^sig^0^0', '<^verb^bs^0^0^sig^0^2', 'marii^noun^bs^0^0^sig^0^2',
'bakka^noun^bs^0^0^sig^0^2', 'buootaa', 'ripaablikaantichii', 'Deeviin', 'Nuunes', '<^verb^0^0^0^0^0^1', 'hara^noun^bs^0^0^sig^0^1', 'jedhanti',
```

Annex 4: A web links scrawled by crawler script.

11. http://www.fanabc.com/afanoromo/index.php/news?start=33 - (0 records)

7. http://www.fanabc.com/afanoromo/index.php/news?start=22 - (1 records, 12 links)
    1. http://www.fanabc.com/afanoromo/index.php/news/item/15499-burjaajjiin-madaallii-meeshaalee-yaaddessaadha-jedhame - (0 records)
    2. http://www.fanabc.com/afanoromo/index.php/news/item/15498-finfinneetti-jiâ€™oottan-2n-itti-aananitti-dargaggoonni-kumni-20-hojiitti-ni-galu - (0 records)
    3. http://www.fanabc.com/afanoromo/index.php/news/item/15497-jiraattonni-magaalaa-nageellee-rakkoon-bulchiinsa-gaarii-akka-hiikkamuuf-gaafatan - (0 records)
    4. http://www.fanabc.com/afanoromo/index.php/news/item/15493-koorporeeshinichi-qajeelfama-kiraa-manaarratti-gatii-kaffaltii-sirreessuu-isa-dandeessisu-qopheesse - (0 records)
    5. http://www.fanabc.com/afanoromo/index.php/news/item/15489-somaaliyaan-balaan-bidiruu-yaman-lammilee-ishee-hedduu-itti-dhabde-akka-qoratamu-gaafatte - (0 records)
    6. http://www.fanabc.com/afanoromo/index.php/news/item/15487-kaampaanonni-hidhicha-irratti-qoâ€™annoo-geggeessan-akka-sagantaa-yeroo-baâ€™eefitti-qorannucha-geggeessaa-akka-jiran-ibsame - (0 records)
    7. http://www.fanabc.com/afanoromo/index.php/news/item/15486-qorannoon-magaalotatti-wiirtuuwwan-kilaastaraa-industirii-ijaaruu-dandeessisu-xumurame - (0 records)
    8. http://www.fanabc.com/afanoromo/index.php/news/item/15484-kiim-joong-uun-nama-isa-gorsu-barbaada-â€˜traamp - (0 records)
    9. http://www.fanabc.com/afanoromo/index.php/news/item/15479-biiruchi-qaamolee-raawwii-gaarii-agarsiisanii-fi-deggarsa-kennanii-beekamtii-fi-badhaasa-kenne - (0 records)
    10. http://www.fanabc.com/afanoromo/index.php/news/item/15478-dargaggoonni-miliyoona-3-hojiitti-seenuuf-galmaaâ€™aniiru-ministeericha - (0 records)
    11. http://www.fanabc.com/afanoromo/index.php/news/item/15477-pireezdaantiin-chaayinaa-ministira-dhimma-alaa-ameerikaa-waliin-mariâ€™atan - (0 records)
    12. http://www.fanabc.com/afanoromo/index.php/news?start=44 - (0 records)
8. http://www.fanabc.com/afanoromo/index.php/news?start=7623 - (1 records, 2 links)
    1. http://www.fanabc.com/afanoromo/index.php/news/item/422-daandiin-xiyaara-itiyoophiyaa-doolaara-miliyoona-200-caalu-buufachuun-durse - (0 records)
    2. http://www.fanabc.com/afanoromo/index.php/news?start=7612 - (0 records)

4. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii - (1 records, 13 links)
    1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15512-maanguddoonni-jaappaan-eeyyama-konkolaachisummaa-isaanii-yoo-deebisan-baasiin-sirna-awwaalchaa-ni-hirâ€™ifamaaf-jedhame - (1 records, 1 links)
        1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15512-maanguddoonni-jaappaan-eeyyama-konkolaachisummaa-isaanii-yoo-deebisan-baasiin-sirna-awwaalchaa-ni-hirï¿½%99ifamaaf-jedhame - (0 records)
    2. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15396-saqaa-haati-ilma-hangafaa-boranaa-ilmasheef-harkatti-keewwattu - (1 records)
    3. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15372-godina-arsii-lixaa-aanaa-shaashamanneetti-himatamaan-yakka-gudeeduu-raawwate-hidhaa-cimaa-waggaa-7n-adabame - (1 records)
    4. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15345-tiyaatirri-hulluuqqoo-booru-ebbifama - (1 records)
    5. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15326-ijaarsi-galma-abbaa-gadaa-odaa-bultum-xumurame - (1 records)
    6. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15073-akka-dhaha-oromootti-gurraandhala-tokkoo-hanga-saddeettii-booranni-jila-guddaa-qaba - (1 records)
    7. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15035-dargaggeessi-zayita-liitira-3-hate-hidhaa-waggaa-2-fi-jiâ€™a-6n-adabame - (1 records, 1 links)
        1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/15035-dargaggeessi-zayita-liitira-3-hate-hidhaa-waggaa-2-fi-jiï¿½%99a-6n-adabame - (0 records)
    8. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14987-roobart-mugaabee-ayyaana-dhalootaa-isaanii-93ffaa-nikabaju-sirna-kanaaf-sangaawwan-150-qalmaaf-qophaaâ€™aa-jiru - (1 records, 1 links)
        1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14987-roobart-mugaabee-ayyaana-dhalootaa-isaanii-93ffaa-nikabaju-sirna-kanaaf-sangaawwan-150-qalmaaf-qophaaï¿½%99aa-jiru - (0 records)
    9. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14976-ardii-8ffaan-argamuun-himame - (1 records)
    10. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14849-aadaa-sirbaa-aanaa-waacaalee - (1 records, 1 links)
        1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14627-sirna-wal-harkaa-fuudhinsa-baallii-gadaa-booranaaf-godaansi-ardaa-jilaatti-taasifamaa-jira - (0 records)
    11. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii?start=10 - (1 records, 11 links)
        1. http://www.fanabc.com/afanoromo/index.php/miiltoo-waarii/item/14562-sirna-gumaa-baasuu-aanaa-ilfataatti - (0 records)

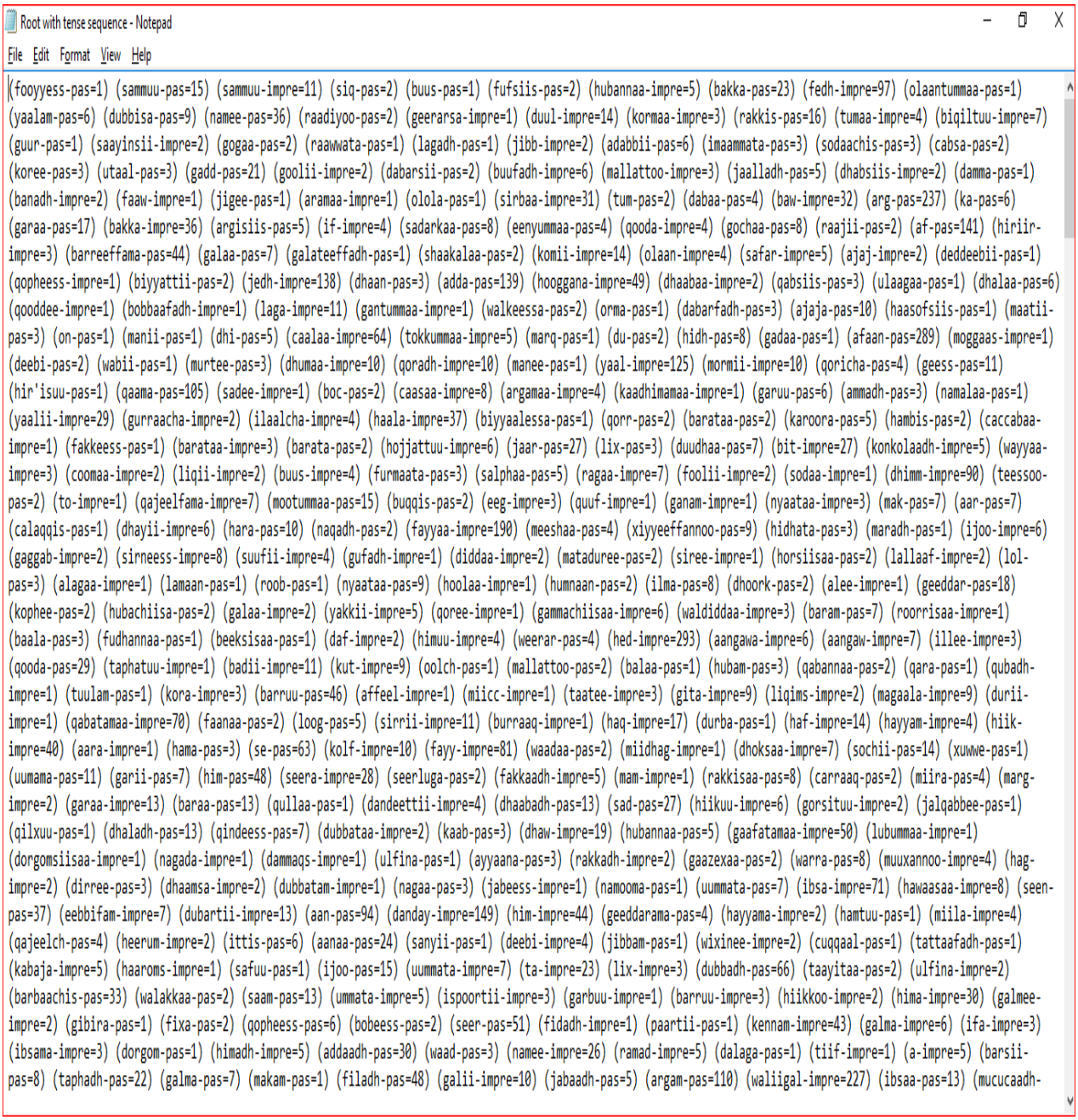
Annex 5: A bigram sequence of stem with number


Root with person2 sequence - Notepad

File Edit Format View Help

(harree-2=1) (guyyaa-2=15) (muraa-3=11) (dubbata-2=2) (loon-3=5) (guyyii-1=3) (qabeessa-1=28) (ilaalam-3=167) (walqixxee-3=3) (ilaalcha-1=10) (sukkaara-1=7) (deebii-3=26) (hama-3=13) (imaammata-3=3) (roorris-1=1) (qulqullina-2=1) (ogeessa-3=1) (xaxamaa-3=4) (qaraa-2=14) (qeerroo-3=1) (gonfadh-3=6) (faarfadh-1=1) (cimina-2=16) (angafa-1=1) (dhuuf-1=1) (hirtaa-2=1) (eegumsa-1=4) (giddugaleessa-2=1) (qabduu-2=3) (hawaasaa-1=33) (barbaad-2=43) (xiinxala-2=4) (shaakal-1=63) (qood-1=64) (olaanaa-2=3) (lag-2=9) (tuffatam-3=6) (dhiig-3=21) (gog-3=479) (biyya-2=40) (ibs-1=330) (guyyoota-1=2) (haguug-3=6) (guyyaa-1=22) (hubadh-2=220) (fajaj-3=1) (jibbisiis-1=6) (sharaf-3=2) (fuggis-3=1) (kenn-3=2144) (tokkee-2=8) (dhabsiis-3=9) (baaftuu-3=2) (dubartii-1=6) (lamaan-2=6) (naannoo-3=32) (onnachiiftuu-1=1) (haasaa-3=1) (amantoota-2=1) (mormaa-3=2) (ijoo-1=25) (daneess-3=1) (ardii-3=2) (hara-3=58) (maqs-1=1) (bareeda-3=2) (adda-2=226) (cuf-1=11) (afaan-1=20) (haadhadh-3=2) (maar-1=5) (hubannaa-3=14) (kormaa-2=9) (gulantaa-2=1) (caalmaa-1=1) (kuf-2=7) (afoola-1=10) (nafa-3=6) (guutadh-3=2) (ibsa-2=70) (fiigaa-2=2) (talaallii-1=1) (geerars-2=1) (qajeelch-1=30) (tiifuu-1=1) (baatii-3=39) (galaa-1=2) (qorraa-1=2) (ifa-1=41) (carraaq-2=7) (harma-1=2) (gat-1=74) (tapha-1=1) (waldoo-2=1) (gammachiisaa-3=1) (dalag-1=9) (cims-3=110) (dirqama-1=24) (qoricha-3=6) (mancaas-3=2) (gocha-1=15) (hidda-2=6) (yaabbadh-3=6) (maaram-2=1) (madaqs-1=1) (jiraadh-1=2) (mirkaneessa-3=2) (sim-3=71) (gocha-2=7) (tiifee-3=1) (garagara-3=4) (gama-2=76) (haam-2=1) (dhamaatii-3=6) (xalayaa-3=5) (akkaataa-1=18) (deebi-2=13) (siphadh-1=1) (naqam-3=13) (uraa-3=3) (mataduree-2=4) (adeem-1=71) (shallag-1=2) (qajeelchaa-2=8) (yoomessa-3=7) (paartii-3=2) (ilaalcha-2=17) (badhaasaa-1=1) (gugguuf-2=1) (hir'isuu-3=1) (abbal-3=2) (se-1=479) (idilee-2=6) (yuba-2=1) (himadh-2=3) (hiram-1=6) (ganam-3=20) (naaf-1=1) (kaasuu-3=23) (keessummeess-3=8) (ganda-1=1) (kaadh-3=18) (beekam-3=54) (aans-3=37) (laatam-3=3) (soor-1=6) (hodhisiis-1=6) (jaam-1=1) (qajeelch-3=5) (lammii-2=4) (amantoota-3=9) (hiikkoo-2=2) (argina-3=1) (danf-3=1) (garii-1=5) (degger-2=1) (labsii-3=14) (hubaa-3=4) (daree-2=4) (bulchiis-3=4) (himuu-1=32) (gufadh-1=2) (leenjisa-3=2) (labsa-3=6) (bakke-1=2) (mootummaa-3=37) (qaqqaba-3=2) (baas-3=310) (guut-1=56) (baraars-1=1) (kutam-1=1) (hag-3=3) (firoom-1=5) (maangoo-1=1) (geerarsa-3=1) (fageenya-3=6) (fufsiis-1=4) (bulch-3=14) (sim-1=11) (qoraan-3=1) (gomma-1=12) (soddaa-3=1) (facaas-1=4) (barataa-2=4) (boqota-1=1) (kophaa-3=6) (dhiiga-2=13) (illee-1=6) (loon-2=3) (sirna-2=3) (baw-2=104) (bara-3=429) (tapha-3=17) (raajaa-3=1) (birii-1=1) (oduu-2=4) (argam-3=145) (sodaa-3=1) (xiqqaa-2=2) (argamaa-1=3) (miidham-3=17) (ijaarsa-2=3) (qorannaa-3=19) (degger-3=13) (iccitii-3=4) (magaalaa-1=10) (miira-3=8) (qeesii-1=1) (olii-3=1) (tarkaanfadh-3=4) (maq-3=16) (maatii-1=8) (eebbis-1=2) (olaanaa-1=295) (jaam-2=31) (namee-1=43) (ajajam-2=2) (kireeffataa-3=1) (lakkaaww-3=2) (barataa-1=4) (walkeessa-3=2) (fiig-1=1) (ijaa-1=6) (hub-1=48) (shanan-2=1) (lafa-1=21) (fal-1=28) (nuura-3=2) (gaafadh-3=149) (eelaa-1=6) (jabina-2=1) (dhiya-2=2) (sabee-3=1) (amansiis-2=45) (carraa-3=24) (funy-1=14) (akeekkachiisa-3=7) (sobaa-3=29) (karra-3=1) (sagada-2=1) (kuusaa-3=5) (raawwata-1=3) (saalaa-3=3) (bittaa-2=13) (mootii-3=37) (jechisiis-3=4) (gadhee-3=5) (nama-2=128) (obsa-1=15) (cufaa-2=1) (hodhuu-2=5) (supham-3=5) (himama-3=1) (hundee-2=5) (laafaa-1=10) (lixa-1=1) (kennii-1=2) (yeroo-3=411) (bad-2=82) (waldoo-3=1) (kakuu-2=5) (qophii-1=3) (badhaadh-3=2) (araada-3=3) (suuq-2=1) (taatee-3=7) (gubee-3=1) (mormii-3=2) (hawaasaa-2=18) (cim-1=15) (haar-2=221) (jettu-1=7) (dhiyeessaa-3=2) (mirkaneessa-1=4) (hundee-1=2) (durba-1=1) (walakkaa-3=2) (fixa-1=3) (guyyii-3=7) (haaroms-1=1) (horadh-2=5) (leenjii-1=3) (gaafatam-1=27) (hiriira-1=1) (na-3=50) (mataa-3=28) (kompuutara-3=1) (kadhataa-1=8) (dhaabbadh-1=1) (xiinxal-3=113) (buusii-3=3) (fuula-2=8) (turaa-2=1) (mal-2=7) (mancaasa-3=2) (jaja-2=1) (ban-3=26) (saayinsii-1=5) (hed-1=3) (finciltuu-2=1) (hambifam-3=1) (nagaa-1=5) (morma-1=13) (fedh-3=189) (sodaadh-2=21) (abaar-1=3) (qilxuu-1=1) (ilaalchis-1=20) (xiqqoo-1=4) (kunuunsa-3=3) (dhi-3=7) (dorgomsiisaa-2=1) (fe'am-2=4) (dhabamsiisa-1=1) (fedha-2=7) (gumaa-3=11) (naasis-1=1) (gaafii-1=1) (dhagna-3=7) (akkeessituu-1=3) (damoo-2=2) (geeddaramaa-3=6) (ilaala-2=7) (sirneeffamaa-3=4) (baas-2=77) (filatam-3=110) (danfis-3=1) (muraasa-2=3) (dhadhaa-2=1) (jabee-1=2) (dargaggoo-3=3) (mucoo-3=1) (uumam-3=193) (gammachuu-2=5) (warra-3=61) (qubata-3=5) (haala-3=648) (odeeffannaa-1=1) (lallab-3=10) (cich-3=3) (nama-1=257) (dhiqadh-1=2) (deddeem-1=5) (nubee-1=2) (akeekkachiisa-2=8) (ogeessa-1=1) (diriirs-3=1) (birmadh-1=2) (haaloo-1=2) (faana-1=2) (seeness-2=26) (uummatee-3=2) (qabeessa-3=4) (dhork-2=6) (maxxansa-3=3) (ifadh-3=51) (dargaggoo-2=1) (jig-3=6) (sodaadh-1=10) (jirbii-1=6) (goota-3=1) (nuff-1=3) (gay-3=310) (dhaga'am-3=4) (madda-2=5) (sirneeffama-3=9) (guyyaafadh-3=2) (warshaa-3=1) (teessis-3=6) (kab-3=2) (geggeess-1=19) (duwwaa-3=1) (yoomessa-2=1) (amantii-2=3) (dhadhaa-1=5) (farad-3=16) (dheeress-3=1) (macca-2=1) (naasis-3=8) (coom-2=1) (ool-3=258) (haf-3=110) (bakke-2=8) (qancar-1=1) (luba-1=1) (tiruu-3=2) (hariiroo-1=2) (cufam-1=3) (gorsituu-2=2) (dhabsiis-1=3) (dayeess-1=10) (seeruu-2=2) (duree-3=1) (dabarsaa-1=64) (aangoo-3=24) (kakuu-3=8) (liqii-2=2) (hordoftuu-2=1) (shallag-3=14) (faaw-2=6) (qabatamaa-2=109) (caama-1=10) (bubbul-3=3) (amantoota-1=5) (iyy-3=181) (hiika-1=70) (falma-2=10) (lo-2=4) (geeddaram-3=25) (kennam-1=278) (buusii-2=1) (dhiyeessaa-2=1) (mijaawaa-3=2) (dadhab-1=16) (barsiisa-3=98) (ulfa-1=5) (xuwwe-3=1) (sirbaa-3=1) (namummaa-2=1) (lixaa-2=6) (seerummaa-3=1) (gaachan-1=2) (waamicha-3=1) (nagaa-2=7) (fedh-1=113) (ramad-1=45) (intala-3=1) (lawwee-2=1) (maq-2=252) (rakkisaa-3=24) (dafqa-2=2) (shubbisaa-2=6) (dhimm-3=195) (uumam-2=5) (magaa-1=6) (goolii-2=19) (sadarkaa-1=25) (buutee-3=2) (dubbis-3=674) (goggogs-2=1) (goll-1=12) (goos-3=1) (got-1=28) (sodaatam-3=2) (tuq-1=12) (jila-2=1) (dhiyeenya-

Annex 6: A bigram sequence of stem with person


Root with tense sequence - Notepad

File Edit Format View Help

(fooyyess-pas=1) (sammuu-pas=15) (sammuu-impre=11) (siq-pas=2) (buus-pas=1) (fufsiis-pas=2) (hubannaa-impre=5) (bakka-pas=23) (fedh-impre=97) (olaantummaa-pas=1) (yaalam-pas=6) (dubbisa-pas=9) (namee-pas=36) (raadiyoo-pas=2) (geerarsa-impre=1) (duul-impre=14) (kormaa-impre=3) (rakkis-pas=16) (tumaa-impre=4) (biqiltuu-impre=7) (guur-pas=1) (saayinsii-impre=2) (gogaa-pas=2) (raawwata-pas=1) (lagadh-pas=1) (jibb-impre=2) (adabbii-pas=6) (imaammata-pas=3) (sodaachis-pas=3) (cabsa-pas=2) (koree-pas=3) (utaal-pas=3) (gadd-pas=21) (goolii-impre=2) (dabarsii-pas=2) (buufadh-impre=6) (mallattoo-impre=3) (jaalladh-pas=5) (dhabsiis-impre=2) (damma-pas=1) (banadh-impre=2) (faaw-impre=1) (jigee-pas=1) (aramaa-impre=1) (olola-pas=1) (sirbaa-impre=31) (tum-pas=2) (dabaa-pas=4) (baw-impre=32) (arg-pas=237) (ka-pas=6) (garaa-pas=17) (bakka-impre=36) (argisiis-pas=5) (if-impre=4) (sadarkaa-pas=8) (eenyummaa-pas=4) (qooda-impre=4) (gochaa-pas=8) (raajii-pas=2) (af-pas=141) (hiriir-impre=3) (barreeffama-pas=44) (galaa-pas=7) (galateeffadh-pas=1) (shaakalaa-pas=2) (komii-impre=14) (olaan-impre=4) (safar-impre=5) (ajaj-impre=2) (deddeebii-pas=1) (qopheess-impre=1) (biyyattii-pas=2) (jedh-impre=138) (dhaan-pas=3) (adda-pas=139) (hooggana-impre=49) (dhaabaa-impre=2) (qabsiis-pas=3) (ulaagaa-pas=1) (dhalaa-pas=6) (qooddee-impre=1) (bobbaafadh-impre=1) (laga-impre=11) (gantummaa-impre=1) (walkeessa-pas=2) (orma-pas=1) (dabarfadh-pas=3) (ajaja-pas=10) (haasofsiis-pas=1) (maatii-pas=3) (on-pas=1) (manii-pas=1) (dhi-pas=5) (caalaa-impre=64) (tokkummaa-impre=5) (marq-pas=1) (du-pas=2) (hidh-pas=8) (gadaa-pas=1) (afaan-pas=289) (moggaas-impre=1) (deebi-pas=2) (wabii-pas=1) (muntee-pas=3) (dhumaa-impre=10) (qoradh-impre=10) (manee-pas=1) (yaal-impre=125) (mormii-impre=10) (qoricha-pas=4) (geess-pas=11) (hir'isuu-pas=1) (qaama-pas=105) (sadee-impre=1) (boc-pas=2) (caasaa-impre=8) (argamaa-impre=4) (kaadhimamaa-impre=1) (garuu-pas=6) (ammadh-pas=3) (namalaa-pas=1) (yaalii-impre=29) (gurraacha-impre=2) (ilaalcha-impre=4) (haala-impre=37) (biyyaalessa-pas=1) (qorr-pas=2) (barataa-pas=2) (karoora-pas=5) (hambis-pas=2) (caccabaa-impre=1) (fakkeess-pas=1) (barataa-impre=3) (barata-pas=2) (hojjattuu-impre=6) (jaar-pas=27) (lix-pas=3) (duudhaa-pas=7) (bit-impre=27) (konkolaadh-impre=5) (wayyaa-impre=3) (coomaa-impre=2) (liqii-impre=2) (buus-impre=4) (furmaata-pas=3) (salphaa-pas=5) (ragaa-impre=7) (foolii-impre=2) (sodaa-impre=1) (dhimm-impre=90) (teessoo-pas=2) (to-impre=1) (qajeelfama-impre=7) (mootummaa-pas=15) (buqqis-pas=2) (eeg-impre=3) (quuf-impre=1) (ganam-impre=1) (nyaataa-impre=3) (mak-pas=7) (aar-pas=7) (calaqqis-pas=1) (dhayii-impre=6) (hara-pas=10) (naqadh-pas=2) (fayyaa-impre=190) (meeshaa-pas=4) (xiyyeeffannoo-pas=9) (hidhata-pas=3) (maradh-pas=1) (ijoo-impre=6) (gaggab-impre=2) (sirneess-impre=8) (suufii-impre=4) (gufadh-impre=1) (diddaa-impre=2) (mataduree-pas=2) (siree-impre=1) (horsiisaa-pas=2) (lallaaf-impre=2) (lol-pas=3) (alagaa-impre=1) (lamaan-pas=1) (roob-pas=1) (nyaataa-pas=9) (hoolaa-impre=1) (humnaan-pas=2) (ilma-pas=8) (dhoork-pas=2) (alee-impre=1) (geeddar-pas=18) (kophee-pas=2) (hubachiisa-pas=2) (galaa-impre=2) (yakkii-impre=5) (qoree-impre=1) (gammachiisaa-impre=6) (waldiddaa-impre=3) (baram-pas=7) (roorrisaa-impre=1) (baala-pas=3) (fudhannaa-pas=1) (beeksisaa-pas=1) (daf-impre=2) (himuu-impre=4) (weerar-pas=4) (hed-impre=293) (aangawa-impre=6) (aangaw-impre=7) (illee-impre=3) (qooda-pas=29) (taphatuu-impre=1) (badii-impre=11) (kut-impre=9) (oolch-pas=1) (mallattoo-pas=2) (balaa-pas=1) (hubam-pas=3) (qabannaa-pas=2) (qara-pas=1) (qubadh-impre=1) (tuulam-pas=1) (kora-impre=3) (barruu-pas=46) (affeel-impre=1) (miicc-impre=1) (taatee-impre=3) (gita-impre=9) (liqims-impre=2) (magaala-impre=9) (durii-impre=1) (qabatamaa-impre=70) (faanaa-pas=2) (loog-pas=5) (sirrii-impre=11) (burraaq-impre=1) (haq-impre=17) (durba-pas=1) (haf-impre=14) (hayyam-impre=4) (hiik-impre=40) (aara-impre=1) (hama-pas=3) (se-pas=63) (kolf-impre=10) (fayy-impre=81) (waadaa-pas=2) (miidhag-impre=1) (dhoksaa-impre=7) (sochii-pas=14) (xuwwe-pas=1) (uumama-pas=11) (garii-pas=7) (him-pas=48) (seera-impre=28) (seerluga-pas=2) (fakkaadh-impre=5) (mam-impre=1) (rakkisaa-pas=8) (carraaq-pas=2) (miira-pas=4) (marg-impre=2) (garaa-impre=13) (baraa-pas=13) (qullaa-pas=1) (dandeettii-impre=4) (dhaabadh-pas=13) (sad-pas=27) (hiikuu-impre=6) (gorsituu-impre=2) (jalqabbee-pas=1) (qilxuu-pas=1) (dhaladh-pas=13) (qindeess-pas=7) (dubbataa-impre=2) (kaab-pas=3) (dhaw-impre=19) (hubannaa-pas=5) (gaafatamaa-impre=50) (lubummaa-impre=1) (dorgomsiisaa-impre=1) (nagada-impre=1) (dammaqs-impre=1) (ulfina-pas=1) (ayyaana-pas=3) (rakkadh-impre=2) (gaazexaa-pas=2) (warra-pas=8) (muuxannoo-impre=4) (hag-impre=2) (dirree-pas=3) (dhaamsa-impre=2) (dubbatam-impre=1) (nagaa-pas=3) (jabeess-impre=1) (namooma-pas=1) (uummata-pas=7) (ibsa-impre=71) (hawaasaa-impre=8) (seen-pas=37) (eebbifam-impre=7) (dubartii-impre=13) (aan-pas=94) (danday-impre=149) (him-impre=44) (geeddarama-pas=4) (hayyama-impre=2) (hamtuu-pas=1) (miila-impre=4) (qajeelch-pas=4) (heerum-impre=2) (ittis-pas=6) (aanaa-pas=24) (sanyii-pas=1) (deebi-impre=4) (jibbam-pas=1) (wixinee-impre=2) (cuqqaal-pas=1) (tattaafadh-pas=1) (kabaja-impre=5) (haaroms-impre=1) (safuu-pas=1) (ijoo-pas=15) (uummata-impre=7) (ta-impre=23) (lix-impre=3) (dubbadh-pas=66) (taayitaa-pas=2) (ulfina-impre=2) (barbaachis-pas=33) (walakkaa-pas=2) (saam-pas=13) (ummata-impre=5) (ispoortii-impre=3) (garbuu-impre=1) (barruu-impre=3) (hiikkoo-impre=2) (hima-impre=30) (galmee-impre=2) (gibira-pas=1) (fixa-pas=2) (qopheess-pas=6) (bobeess-pas=2) (seer-pas=51) (fidadh-impre=1) (paartii-pas=1) (kennam-impre=43) (galma-impre=6) (ifa-impre=3) (ibsama-impre=3) (dorgom-pas=1) (himadh-impre=5) (addaadh-pas=30) (waad-pas=3) (namee-impre=26) (ramad-impre=5) (dalaga-pas=1) (tiif-impre=1) (a-impre=5) (barsii-pas=8) (taphadh-pas=22) (galma-pas=7) (makam-pas=1) (filadh-pas=48) (galii-impre=10) (jabaadh-pas=5) (argam-pas=110) (waliigal-impre=227) (ibsaa-pas=13) (mucucaadh-

Annex 7: Trigram sequence of stem

Untitled - Notepad

File Edit Format View Help

(('jira', 'kitaabni', 'gabaabaa')=1) (('naannoo', 'oromiyaa', 'boqodh')=1) (('afaanonni', 'itoophiyaa', 'hund')=2) (('himuu', 'sillaaseefi', 'dargiiti')=1) (('waan', 'yeroo', 'dh')=1) (('olaantuu', 'ta', 'yunaayitid')=1) (('walfakkeenya', 'gosaa', 'gosa')=2) (('fottoksuu', 'lafee', 'ogeessaan')=1) (('dhaabbadh', 'dhaabbata', 'waliigal')=2) (('jijjiin', 'sirrii', 'dhug')=1) (('kenna', 'yesus', 'tokkicha')=2) (('ibsa', 'keyyattoonni', 'haf')=2) (('qulqullina', 'jalqab', 'jalqaba')=1) (('haala', 'boqonnaan', 'waggaa')=1) (('mallatto', 'cuuph', 'hafuura')=1) (('noofsa', 'kutaa', 'kutii')=2) (('baay', 'neen', 'isatti')=2) (('laas', 'tikii', 'epireen')=2) (('dhabummaa', 'adda', 'baas')=1) (('qab', 'lachanuu', 'guyyaa')=1) (('aman', 'gargaar', 'afaa')=2) (('qaba', 'mata', 'duree')=3) (('dubbata', 'iis', 'garaagarummaa')=1) (('maalfakkaataa', 'tur', 'barri')=2) (('dubartitti', 'argisiisu', 'baay')=1) (('hammatu', 'jechuudha', 'haaluma')=1) (('guurrattee', 'dhufteenwarrii', 'mirg')=1) (('calluma', 'jedh', 'nakka')=1) (('hidh', 'hidhadh', 'hidhata')=2) (('madda', 'bar', 'barruu')=1) (('harka', 'qabamuutiin', 'manaa')=1) (('jechaan', 'nata', 'duree')=1) (('jjetanirratti', 'tauu', 'qab')=1) (('gosoota', 'og-barruu', 'kaanirraa')=1) (('ilaalcha', 'hiikka', 'namee')=1) (('barnoota', 'walqixxe', 'barnoota')=1) (('jettuun', 'hawaasa', 'haal')=1) (('walitti', 'fid', 'aad')=1) (('immoo', 'naadaa', 'nnama')=1) (('aan', 'a', 'dhuf')=5) (('beektota', 'biyya', 'alaa')=2) (('morm', 'mormii', 'mana-murtii')=1) (('maqa', 'maqaa', 'isaatiin')=2) (('nkan', 'dh', 'turedha')=1) (('akka', 'ogbarruu', 'mormii')=1) (('kallatiidhaan', 'haal', 'haala')=2) (('ogbarruu', 'dubbii', 'jedhama')=1) (('danqaa', 'danqa', 'hiisaniiru')=1) (('tumaale', 'heera', 'mirgoota')=1) (('dandeenya', 'jechoonni', 'ngaaffiil')=1) (('neroo', 'taasis', 'kaayyoon')=1) (('keeyyata', 'amansiis', 'falm')=2) (('garaagaruummaa', 'jir', 'agarsiis')=1) (('qarshii', 'qab', 'qaba')=1) (('qubeessuu', 'afaa', 'oromoo')=1) (('jir', 'maallaqa', 'jijjiir')=2) (('nama', 'ntii', 'tilmaan')=1) (('tokko', 'qabadh', 'qabatamaa')=3) (('adeeman', 'waliin', 'adeemsa')=1) (('jirta', 'kutaa', 'ammo')=1) (('amala', 'keeyyata', 'gaarii')=1) (('xiyyeeffatu', 'taasis', 'qorataan')=1) (('qubee', 'laatiiniin', 'barreeffame')=2) (('garuu', 'kakuu', 'moofaadhaan')=1) (('deebii', 'mudde', 'barreesseen')=1) (('turerraan', 'afaan', 'oromoo')=1) (('mur', 'murtee', 'isheen')=1) (('ofii', 'isaa', 'barumsa')=1) (('kanas', 'banksfi', 'banks')=1) (('keewwata', 'jalatti', 'ibs')=1) (('asirratti', 'inni', 'yesuus')=1) (('gadaati', 'dubbii', 'himuaadaa')=2) (('dhugaa', 'jir', 'faalleessu')=1) (('john', '2005', 'business')=4) (('ilaalam', 'ilaal', 'osoo')=2) (('hojjadh', 'nakka', 'dorgom')=2) (('kenn', 'gargaar', 'qaba')=1) (('danbii', 'qajeelfama', 'hojjattuu')=2) (('hinturree', 'eeyyee', 'deebiin')=2) (('kakaas', 'kakaasa', 'akka')=1) (('kakuu', 'moofaani', 'ibsa')=1) (('dhumaa', 'egaree', 'nuumsa')=1) (('kitaaba', 'kitaabee', 'dubbisaa')=1) (('hojjachuu', 'baayyatee', 'isaaf')=2) (('irratti', 'hunduu', 'dubbachu')=2) (('naraa', 'garaa', 'irraa')=1) (('arg', 'ittigaafatamtoonni', 'sadarkaa')=2) (('hayyam', 'irratti', 'hundaawuudhaan')=2) (('idil-addunyaa', 'ibs', 'miidhamtoonni')=1) (('battala', 'bilcheessu', 'pressure')=1) (('barsiifni', 'yesuus', 'yeruma')=1) (('ijaar', 'yaal', 'isaa')=2) (('koree', 'qoradh', 'qor')=3) (('keeyyata', 'kaay', 'dhiyaatanii')=1) (('deemtotaaf', 'durs', 'kenn')=1) (('guddaan', 'nama', 'kaay')=1) (('mal', 'mala', 'karaa')=1) (('jalqaba', 'ykn', 'madda')=1) (('isaa', 'barreeffachuu', 'rakko')=1) (('kaay', 'hundinuu', 'daangeeffamaniidha')=1) (('kana', 'kaay', 'axareeraan')=1) (('qaba', 'shaakala', 'gaaffilee')=2) (('alaa', 'qab', 'qabam')=2) (('1986', 'folk', 'groups')=3) (('ngadiitiin', 'nhaa', 'nilaall')=1) (('guddaa', 'qab', 'qaba')=16) (('guddaadha', 'namni', 'daangaa')=1) (('maamiltoota', 'bilbil', 'bilbila')=1) (('nirratti', 'geggeess', 'geggeessaa')=1) (('dalloo-mannaa', 'ganditii', 'bona')=1) (('tokko', 'hindubbisiisiin', 'dandeettii')=1) (('barbaad', 'nama', 'jedh')=1) (('kallattii', 'nlaman', 'geggeess')=1) (('nraajii', 'nraga', 'nkan')=1) (('gat', 'yakka', 'yeroo')=1) (('fidanirratti', 'yaad', 'yaada')=1) (('dhala', 'nama', 'hundumaa')=1) (('olaantummaan', 'geggeess', 'abboo')=1) (('qabu', 'jedham', 'adda')=1) (('waan', 'jedhanii', 'adda')=1) (('caalaa', 'cims', 'bakka')=2) (('dhiiraafi', 'durba', 'ndhiiraafi')=1) (('nrruulee', 'nbiroo', 'kamuu')=1) (('qabaadh', 'qaba', 'hindagatiin')=1) (('harka', 'caalmaan', 'qajeelfama')=1) (('abbaa', 'mana', 'qab')=2) (('gors', 'barreeyse', 'akeekni')=1) (('aman', 'warra', 'giriikiitiin')=1) (('qabamaa', 'akkaataa', 'labsaa')=2) (('jiini', 'tokk', 'guyyoota')=1) (('waaqummaatti', 'hirmaannaa', 'waaqummaatti')=1) (('faayidaa', 'qorannoo', 'qab')=2) (('eerutti', 'gundoo', 'tokko')=1) (('seera', 'naan', 'raawwatame')=2) (('kaayyoo', 'aars', 'aarsaa')=1) (('karaa', 'gam', 'gama')=1) (('jiruu', 'hubaatuun', 'mata')=1) (('waaqa', 'warra', 'peershiyaa')=1) (('jiddugalli', 'dagatamtu', 'hojiilee')=1) (('1931', 'bakka', 'uummata')=1) (('ittiin', 'jedh', 'abbaa')=1) (('hiika', 'isa', 'nisaaf')=1) (('dhug', 'dhugaa', 'nbaateef')=2) (('nkennamaaf', 'mana', 'barumsaatti')=2) (('nyeroo', 'hunda', 'olkaas')=1) (('ture', 'kiraaf', 'itoophiyaa')=1) (('beekam', 'maxxa', 'nnsanii')=2) (('inni', 'beellada', 'qalma')=1) (('gurguddoo', 'dh', 'irraa')=1) (('amaloota', 'walquunnamtii', 'tokko')=2) (('manguddoota', 'lammii', 'dhata')=1) (('bara', 'baraa', 'natuun')=1) (('guutam', 'guutama', 'gaaffii')=1) (('jaakeeta', 'wandabo', 'gababa')=2) (('gilgaala', 'ragaa', 'xiinxal')=1) (('gadi', 'qotame', 'tur')=1) (('raawwatame', 'fakkaatu', 'garaagarummaa')=1) (('sadarkaalee', 'kuninis', 'kaneen')=1) (('naannessuf', 'qandilleessu', 'busaafi')=1) (('adda', 'irraa', 'maaltu')=1) (('uffata', 'dubartii', 'wuccuu')=2) (('garaa', 'garaa', 'kanneen')=1) (('nidandeessa', 'walquunnamtii', 'kaay')=4) (('yaadidetama', 'jedh', 'hiisa')=1) (('ogbarruu',

Annex 7: A python code preprocessing PDF document

```
import PyPDF2
import re
import sys
import os
import codecs
import string
import nltk
def create_dir(folder):
    corpusdir = folder
    if not os.path.isdir(corpusdir):
        os.mkdir(corpusdir)
def Read_pdf_file(file):
    p_file = file
    pdffile = open (p_file, 'rb')
    pdfreader = PyPDF2.PdfFileReader(pdffile)
    no_of_page = pdfreader.numPages
    lis = []
    for page_no in range (no_of_page):
        pageobj = pdfreader.getPage(page_no)
        content = pageobj.extractText()
        page = content
        lis.append(page)
    text = lis
    pdffile.close()
    return text
def cleanInput(filepath):
    cleInput = []
    text = Read_pdf_file(filepath)
    stringfi = str(text)
    strings = nltk.sent_tokenize(stringfi)
    count = 0
    for se in strings:
        sen = strings [count]
        for qu in sen:
            clen = re.sub('\^[\n+?][.*?]', " ", sen)
            clen = re.sub('\W', " ", clen)
            clen = re.sub('\s', " ", clen)
            clen = re.sub(' +', " ", clen)
            clen = re.sub('\[[0-9*]\]', " ", clen)
            clen = bytes(clen, "UTF-8")
            clen = clen.decode("ascii", "ignore")
```

```
        for item in clen:
            item = item.strip(string.punctuation)
            if len(item)> 3:
                if item.startswith('n')and item[1].isupper() :
                    item = item.strip('n')
                    cleaninput.append(item)
                else:
                    cleaninput.append(item)
        cleInput.append(cleaninput)
        count+=1
    return cleInput
def convertStri(list_stringf):
    stringf = cleanInput(list_stringf)
    count = 0
    tokens = 0
    corp = []
    for sent in stringf:
        sent = stringf[count]
        sent = ' '. join(sent)
        word = nltk.word_tokenize(sent)
        for i in word:
            tokens = tokens + 1
        if tokens > 4:
            senr = sent.strip(string.punctuation)
            senr = senr + '<EOS>'
            corp.append(senr)
        count +=1
    han = " ".join(corp)
    return han
def out_put_file(file,Nfile):
    pdf = file
    paths = r'C: \Users\Toshiba\Desktop\UPDF\corpusor/'
    create_dir(paths)
    filena = Nfile
    for t in pdf:
        with open(paths+str(filena)+ '.txt', 'w', encoding ='ascii', errors = 'replace') as fout:
            sys.stdout = fout
            print(pdf)
```