ADAMA SCIENCE & TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

N-GRAM WORD PREDICTION FOR AFAN OROMO WORDS

By: Duressa Tamirat Gemeda

Advisor: Shimelis Assefa (PhD) ,Associate Professor

A Thesis Submitted to Adama Science & Technology University School of Electrical Engineering and Computing in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information System

Adama, Ethiopia
Septemper, 2016

ADAMA SCIENCE & TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

CORPUS BASED AUTO-COMPLETION OF WORD PREDICTION FOR AFAN OROMO WORDS

Adama, Ethiopia

Septemper, 2016

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

CORPUS BASED AUTO-COMPLETION OF WORD PREDICTION FOR AFAN OROMO WORDS

By: DURESSA TAMIRAT GEMEDA

Name and signature of Members of the Examining Board:

Name	Title	Signature	Date
Shimelis Assefa (PhD)	Advisor	_____	_____
_____	Examiner	_____	_____
_____	Examiner	_____	_____

Adama, Ethiopia

Septemper, 2016

APPROVAL SHEET

This Research Thesis entitled "TOWARDS CORPUS BASED AUTO-COMPLETION OF WORD PREDICTION FOR AFAN OROMO WORDS)" has been read and approved as meeting the preliminary research requirements of the Department of Computing in partial fulfillment for the award of the degree of Master in Information System, Adama Science and Technology University, Adama, Ethiopia.

Shimelis Assefa (PhD) _____ _____

Advisor Signature Date

_____ _____ _____

External Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

_____ _____

Institute coordinator for Research, P, GS&C Signature Date

Abstract

This study presents word prediction for Afan Oromo words. Word prediction is an application of Natural language processing that is used to auto complete the words that the user types the first letter or letters of a word and the system provides one or more higher probable words. The main objective of this study is to develop word prediction that predict and auto complete the words in the corpus. The developed corpus collected from governmental medias, news, cultural documents, history of societies for the sake of this study only. In order to develop the model, we have used unsupervised machine learning due to lack of large training corpus for this language. The idea behind the approach is to overcome the problem of a bottleneck, while unsupervised approach can be suitable when there is scarcity of training data. This makes our approach suitable for prediction when there is lack of resources. The algorithm that used in this study was N-grams algorithms (Unigram, Bigram and Trigram) for auto completing a word by predicting a correct word in a sentence which saves time and keystrokes of typing and also reduces misspelling. We used small data corpus of Afan Oromo language of different word types to predict correct word with the accuracy as much as possible. The result and finding are promising. We hope that our work will impact current state of understanding for automated Afan Oromo typing. This work describes how we improve word entry information, through word prediction, as an assistive technology for people with motion impairment using the regular keyboard, to eliminate the overhead needed for the learning process. We also present evaluation metrics to compare different models being used in our study. The result argued that WP yields an accuracy of 90% in Unsupervised Machine Learning Approach. The achieved result was encouraging, despite it is less resource requirement. Yet; further experiments using different N-gram approaches that extend this work are needed for a better performance.

Key words: Word Prediction, Auto-Completion, Afan Oromo, N-gram, Unsupervised Word Prediction.

Declaration

I declare that this thesis is my original work that had never been submitted for any award of degree in any University.

_____

Duresa Tamirat Gemeda

September 2016

Acknowledgement

First of all, I would like to thank Jesus Christ who helped me to succeed in all my life long learns. Next, I would like to thank my Advisor Shimelis Assefa (PhD) for his valuable assistance in providing his genuine, professional advice and encouragement that go even beyond the accomplishment of this study. All noteworthy aspects of this study are entirely because of him. He initiated me to do this study by giving precious comments on necessary points. My thanks go to him again, since it is difficult to mention his contribution to my achievements in words, it is better to say my heart has recorded it forever. I would like to acknowledge my friend Workineh Tesema for all his guidance at every step of the thesis work, for patiently listening to me even during uncountable hours, for advising me to do my study, for imparting so much valuable knowledge and for all his encouragement and words of kindness. And also I would like to acknowledge my wife Amina Shafo for all financial support that she adjusts with me as I learn this program and her encouragement and support that she does every step in the all education of this master program. Next, Words cannot express my deepest thanks to my Father Mr. Tamirat Gemeda and mother Sanbate Ittiso and my lovely Sister Ayantu Tamirat for her moral and financial support I received from her throughout my life of learning. My honest thanks equally go to All my Brothers for their help in providing me concerning background information. I would like to thanks all my friends those help me on this program. I would like to thank to Oromia Radio and TV organization for they gave me the time to learn this program and for their permission to data collection

Table of Contents

Contents **Page**

List of Tables

List of Figures

Acronyms

AAC- Augmentative and alternative communication

AWT-Abstract Window Toolkit

CLI-Common Language Infrastructure

GUI-Graphic User Interface

IA- Inter annotator agreement

ILP-Integer linear programming

JVM-Java virtual machine

K-NN-K-Nearest Neighbor

LDA-Latent Dirichlet allocation

LSA-Latent semantic analysis

MSW-Microsoft word

NLP - Natural Language Processing

OBS- Oromia Broadcasting Service

OMN-Oromia Media Network

ORTVO- Oromia Radio and Television organization

PLSA-Probabilistic latent semantic analysis

POS- Part of Speech

SNoW- Sparse Network of Winnows

SVM- Support Vector Machine

VOA-Voice of America

WP- Word Prediction

Chapter One

1. Introduction

1.1. Background of the Study

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages (Gerald R. Gendron, 2015). As such, NLP is related to the area of human computer interaction. As advances in technology grow fast, Natural Language Processing (NLP) plays a great role in our day to day activity especially in relation to processing natural language which includes word prediction, auto completing the words. In Ethiopia there are more than 80 languages that are used as regional and federal communication. Many people's are using their hand held devices or computers to create different files by their own mother tongue. While writing the text especially using local languages it takes time and resource hence most of word or text editors use English language. Especially regional languages like Afan Oromo where there is long vowels and short vowels. In case of long vowels which is vowel repetition it needs to prediction and completion to save time and resources of users particularly for people with disabilities, poor command of spelling (Yarowsky, 2007)

Prediction is an application of NLP which helps users by suggesting words and reduce spelling errors. Predication suggests words for users to read and use to help them identify and locate misspelling words. Word prediction is an important function that supportst user's strengths to assist writing (Roberto, 2009). In addition Word completion programs help monitors user input, and predicts words as the user type's letters.

Auto completion applications are programs that completes words depending on the first few character(s) of words as popup menu for the writers as option so that the user can select the exact word(s) to be written. If the word that the user wants to write is not present in popup menu as prediction, the writer must enter the next letter of the word (Sutherland, 2004). Applications like mobile input interfaces, for instance, include word completion in which a partially completed word can be completed to the full word intended by the user pretty like the once used by the smart phone and others devices. Since text prediction system can play a great role in assisting individuals with disabilities increasing the performance of typing speed

and for poor spelling. Predictive text inputting is the process of adoption of an intelligent algorithm that predicts words (Gudisa, 2013).

The absence of these applications in the local languages present a challenging problem in the society. In rural and urban areas where Hotels, Private Companies and Governmental Organizations are facing a problem while making their identity name, trade mark or advertising their product. For example, instead of saying *Daabboo*[Bread], they are saying *Daboo*[Cooperation] with the absence of the letter 'b' which makes completely different spelling, sense, speech and form of the language.

Therefore, the main idea of this study was to develop word prediction prototype for Afan Oromo specifically at word level. Afan Oromo uses Latin based script called Qubee and it has 26 basic characters. The method that used in this study was unsupervised machine learning hence there is no standardized annotated corpus for training the machines. Generally, guessing the next character or word (or word prediction) is an essential subtask of NLP application, hand-writing recognition, augmentative communication for the disabled, and spelling error detection. The motivation behind WP is Afan Oromo to allow the users to make sample use of the available technologies because word prediction present in any language provide great difficulty in the use of information technology as words in human language.

1.2. Statement of the Problem

Afan Oromo is an official language of Oromia Regional State (which is the largest regional State among the current Federal States in Ethiopia) (Keno, 2002). It is one of the major languages that are widely used as a working language of the Regional State of Oromia that is used in Ethiopia and Africa (Girma, 2014). Afan Oromo has a large number of ethnic groups when compared with the rest of Ethiopian langua (DINEGDE, June, 2012)ges (Bekele, 2011). There are large numbers of Afan Oromo speakers in East and Central Africa as well as other neighboring countries like Kenya.

One of the existing problems with Afan Oromo language is lack of auto word completion services. Some speakers of the language may not be familiar with how to spell some words in Afan Oromo because of pronunciation. The problem is more severe for peoples where Afan Oromo is their second language due to different pronunciation in different zones of the

regions hence they are writing as they are pronouncing which may not correct The lack of auto word completion, creates multiple problems with Afan Oromo texts literary works such as journals, Books, fictions and some newspapers. This may undermine the wide usage of the Oromo language, limits the amount of linguistic research in the region and impacts image of Afan Oromo for new generation due to misspelling the words. This has a potential to restrict readers to enjoy books published using Afan Oromo language due to wide range of inconsistencies in spelling. As a result, it can be safely assumed that books and other knowledge produced about the Oromo people may lead to varying interpretations and hence the future generation may have little knowledge of culture, custom, religion, etc. The problem of misspelling is magnified when non-Oromo language history Speakers try to create document using text editing software.

In addition the speed of typing could be greatly hampered when people write Afan oromo texts and at the same time misspelled word that create miscommunication between Authors and readers. Single letter may change the meaning of the word if misspelled. Furthermore, lack of Afan Oromo word auto completion impacts non-native speakers from learning Afan Oromo language in its proper form. Due to misspelling and low speed of typing, the new Afan Oromo speakers could develop low-esteem that prevents them from practicing the language. Therefore, this study is undertaken to solve the above listed problems by providing Afan Oromo word auto completion.

The purpose of this study is to design and develop a system that plays a significant role in prediction and completion to enhancing the word production. However, this study limited to develop prototype that helps to auto complete while writing words in Afan Oromo language.

1.3. Objective of the Study

The general objective and specific objectives of this study are as follow:

1.3.1. General Objective

The main objective of this study is to prepare corpus based Afan Oromo word prediction prototype particularly at word level.

1.3.2. Specific Objective

In line with the general objective, the research is aimed at addressing the following specific objectives:

- To review related works so as to have a conceptual understanding of the state-of-the-art in Afan Oromo word prediction .
- To acquire and prepare the required corpus from different sources to use it for training and contribute as the language resource;
- To develop word prediction prototype for Afan Oromo;
- To evaluate the performance of the model;
- To draw conclusion and recommendation;

1.4. Research Question

In this study, to attain the objectives of the study effectively attempt will be made to answer the following basic research questions.

- What are the technical applications that are appropriate for Auto completion of word prediction for Afan Oromo word
- What is the current status of Afan Oromo word prediction and auto completion?
- To what extent Auto completion of word prediction for Afan Oromo increases the consistent use of the language?

1.5. Scope and limitation of the study

The scope of the study is to investigating word prediction for Afan Oromo words particularly at word level which study the actual prediction of the word given in corpus. The technique used was an unsupervised machine learning approach; this study is conceptually limited to developing prototype for Afan Oromo words that auto complete words. Due to the absence of standard train and test corpus, we have prepared small training and test sets for the experimentation which needs further development for other purposes.

1.6. Significance of the study

The importance of this study is to increases low speed typing and reduce miss spelling word in Afan Oromo Language, by providing word prediction and auto completion. So it creates the conducive environment for writers, speakers and readers to read and write books, magazines, journals, and newspapers that are published in Afan Oromo. Additionally the study finding has significance for public at large to benefit from the result of this study. Every speaker of Afan Oromo language as their mother tongue, and other people who want to use Afan Oromo in their work will benefit from the study.

In addition to the benefits stated above, , this study contributes towards the development of Afan Oromo by creating convenience, ease of use, and consistency for both popular culture and scientific and academic circles. Besides, the result of this study helps to produce experimental evidences that demonstrate different application areas of machine learning technique to word auto completion of the language. It also contributes to future research and development in the area of Natural Language Processing (NLP) specifically in misspelling, sense disambiguation, machine translation, text processing, Information Retrieval, grammatical analysis, content and thematic analysis as those areas require sense disambiguation and auto completion as complement.

1.7. Organization of the study

The thesis is organized into six chapters comprising Introduction, Literature review, overview of Afan Oromo, Methodology, Experimentation and Evaluation, Conclusion & Recommendations. The first chapter gives the general introduction of the study. The second chapter presents review made on different literatures regarding word prediction together with its approaches and different machine learning techniques. The third chapter discusses the overview of the grammatical categories in Afan Oromo. The fourth chapter discusses the methodology and presents the process of data preparation for the system to be developed for Afan Oromo. The fifth chapter discusses the experimentation and discusses of the findings. Chapter sixth presents the conclusion and the recommendations as well as some directions to future works.

Chapter Two

2. Related Literature Review

2.1. Introduction

In this chapter, related literature review in the Word Prediction and completion is presented. This part of study sheds light briefly on some of the work done around the topic of investigation and situates this study in the body of knowledge in word completion/suggestion and natural language processing.

2.2. Basic Concepts of Word Prediction

Word prediction is often recommended for busy users as a means to improve typing speed for clients with physical limitations. Although literature suggests that word prediction does have an effect on writing proficiency, increased speed is not one of its benefits when used with a standard keyboard. One reason given for the failure of word prediction to accelerate typing is that the user must look away from any source document to scan the prediction list during typing. Looking away from the source document may slow the typist more than any acceleration offered by word prediction. For input methods that already require the typist to look away from the copy, this effect might be irrelevant. The focus of this research was to determine whether word completion or word prediction programs would increase typing speed when used with an input method (an on-screen keyboard) that also requires looking away from the source document. Overall, these results show that the use of word prediction and word completion may assist on-screen keyboard users to improve typing speed.

The word prediction and completion is also auto complete that the user types the first letter or letters of a word and the system provides one or more higher probable words. If the word he intends to type is included in the list he can select it, for example by using the number of keys. If the word that the user wants is not predicted, the user must type the next letter of the predicted word. At this time, the word choice(s) is altered so that the words provided begin with the same letters as those that have been selected or the word that the user wants appears it is selected (Damerau, 1964). The other one is misspelling corrections which occur due to typing errors and lack of knowledge of the correct spelling. Presumably, when not being certain and trying to reproduce the correct spelling of a word, second language learners might also need to rely on the pronunciation of that word.

2.2.1. Word Completion

Word completion is the task of predicting and automatically completing words that the user is in the process of typing. Such work can prevent misspellings, help develop writing skills, and accelerate typing speed by saving keystrokes. Word completion is the process in which words that include the entered letter sequence are suggested by displaying a list of the most relevant words. The process of word completion is as follows: as the writer types the first letter or letters of the word, the system produces one word or a list of words beginning with the entered letter sequence. Each time a letter is added, the list is updated. When the target word appears in the list, it can be selected and inserted into the ongoing text with a single keystroke. Typically, word lists are numbered and words can be selected by typing the corresponding number. If the word that the user seeks is not predicted, the writer must enter the next letter of the word.

Word Completion are applicable in several areas which include search Engines like Google, word processors like Ms Word, open office, Email composition where texts tend to be repetitive, programming. It is an important Natural Language Processing (NLP) task in which we want to predict (determine) the correct word in a given context. Word prediction task can be employed in many applications, for example, predictive text entry systems, word completion utilities and writing aids. In word completion utilities, the word prediction task can start after typing the first letter of the target word, so that, the prediction task can be limited to alternative words that start with that entered letter (Homayoon S. and Beigi, 1992).

2.2.2. Predictive Typing Aids

The Researcher have carefully separated the reactive keyboard's prediction mechanism from its user interface. As already been noted the same prediction mechanism can be accessed through quite different interface to give system with a completely different "Up, Down and complete" This separation is particularly appropriate when dealing with tools for the speed of the writing support.One of the first predictive typing aids was the Reactive Keyboard (Darragh, et al., 1990), which made use of text compression methods (Witten, et al., 1999) to suggest completions. The reactive keyboard seeks the best of both worlds by employing a variable length context matching technique, which was originally developed to estimate character probabilities for text compression

2.2.3. Proposed Approach

In this research we propose the n-gram statistical approach. Statistical methods generally suggest words based on:

1. Frequency, either in the relevant corpora or what the user has typed in the past; or
2. Recency, where suggested words are those the user has most recently typed. Such approaches reduce keystrokes and increase efficiency, but they make mistakes. Even with the best possible language models, these methods are limited by their ability to represent language statistically. In contrast, by using semantic knowledge to generate words that are semantically related to what is being typed, text can be accurately predicted where statistical methods fail.

Currently, there are many approaches, which have been developed for use in word prediction. These approaches can be classified into three groups:

1. Use of statistical calculations, which can be either dynamically adaptable/not adaptable to their user;
2. Use of syntactic information in order to improve the accuracy of the prediction;
3. Use of semantics of the sentence to help the prediction process (Matthew E, 1996).

Two different methods are used to produce the predictions. The first method sorts the whole lexicon according to the frequency order and offers the user some few words from the prediction list with the highest frequencies. If the prediction, which 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 (Matthew E, 1996).

The second method of using a fixed lexicon requires more detailed corpus data. “Words stored in the corpus are untagged with the frequencies with which they appear after other given words. Consequently, when a letter (character) is entered, the most frequently used word which follows it can be extracted from the corpus to produce a predication list. The advantage of this method over the previously discussed method is that the prediction list will

be much more on the current status of the word and therefore more likely to contain correct predictions. This makes it a far more common choice for designers of prediction systems. The process of selecting a word is the same as for the previous method, although the list, which is initially drawn up, is changed each time a new word is entered rather than being a fixed list as was seen before" (Matthew E, 1996). Now if any of these four characters are not in the training corpus then the probability of the word will be zero (mean that a word is absent in the corpus). So in this statistical method if we want to consider these words then we need a huge data corpus that must contain all the words of the language. So there may arise the problems data scarcity which is:

- Many entries in corpus are with low frequency
- Probability of a word sequence will be very low or zero

If an entry does not exist in corpus, then the probability of the word will become zero because of data sparse.

2.3. Approaches to Word Prediction Machine Learning

Many approaches have been used through the evolution of WP research. Based on the level of supervision the machine learning approaches are categorized as supervised machine learning, unsupervised machine learning and Bootstrapping machine learning approaches.

2.3.1. Supervised Word Prediction

Supervised Word Prediction use machine-learning techniques to learn a classifier from labeled training sets, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class) (Roberto, 2009). The systems in the supervised learning approach category are trained to develop a classifier that can be used to assign a yet unseen example to one of the predicted words. That means, there is train corpus, where the system learns to classify and a test corpus which the system must annotate. So, supervised learning can be considered as a classification task.

Supervised learning requires labeled training data where every instance in the training data is associated with an output value or label that can be thought of as a special attribute or feature for each instance. For WP, every instance in the training data should be assigned a label that corresponds to the correct prediction of the word that the instance contains or represents. Machine learning algorithms make use of the instance attributes or features of the training data and generate a model to predict the label of any given instance. This model can be

applied to unseen instances to predict their labels. Algorithms that can learn to predict discrete valued labels are classification algorithms or classifiers, whereas the algorithms that can learn to predict continuous valued labels are called regression algorithms.

2.3.2. Unsupervised Word prediction

Unsupervised Word Prediction methods are based on unlabeled corpora, and do not exploit any manually tagged corpus to provide a prediction choice for a word in context unlike supervised method (Roberto, 2009). Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck (Gale, 1993), that is, the need for large-scale resources manually annotated with word predict. They do not rely on labeled training text and, in their purest version, do not make use of any machine-readable resources like dictionaries, thesauri, ontology. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses (Doina Tatar, 2001). Most of the time, supervised approaches are superior to unsupervised in terms of accuracy of prediction when used on the same type of words that systems were trained on (Ide, 1998). This is especially evident in the creation of corpora that is manually annotated (tagged) with the senses, which are used for training machine learning classifiers in a supervised setting. There are two important problems during manual tagging of a corpus: low inter annotator agreement (IA) and high cost of the annotation process.

2.3.3. Bootstrapping Machine Learning Approach

The bootstrapping approach is situated between the supervised and unsupervised approach of Word Prediction. The aim of bootstrapping is to build a prediction with little training data, and thus overcome the main problems of supervision: the data scarcity problem specially lack of annotated data. The bootstrapping methods use a small number of contexts labeled with prediction having a high degree of confidence. This is then used to extract a larger training set from the remaining untagged prediction. Repeating this process, the number of training contexts grows and the number of untagged contexts reduces. In addition to the above approaches, another criterion for categorization of approaches is use of data, knowledge (dictionary) (Beg, 2013).

2.4. Corpus - Based Approaches

A major challenging face WP research is the ability to acquire a large number of words with their frequency. Corpus-based approaches came up with an alternate solution to the challenge by obtaining information necessary for WP directly from a corpus. A corpus provides a bank of samples which enable the development of numerical language models, and thus the use of corpora goes hand-in-hand with empirical methods.

2.5. N-gram Model Algorithm

An n-gram is a language model which is simply a sequence of units drawn from a longer sequence; in the case of text, the unit in question is usually a character or a word. N-grams are used frequently in natural language processing and are a basic tool text analysis. Their applications range from programs that correct spelling to creative visualizations to compression algorithms to generative text. n-gram uses the previous N-1 words in a sequence to predict the next word. It is a language model which contains unigrams, bigrams, and trigrams. The model will train on the very large corpora (Gregory W. et al., 2007).

For this study we recommend n-gram algorithm due to N-gram model for word prediction is a more statistical approach to predict upcoming word in a sentence with more accuracy and N-gram language models provide a natural approach to the construction of sentence completion systems. To measure the probabilities by statistical model like N-gram data is divided into training set and test set.

N-gram language model is a type of probabilistic language model where the approximate matching of next item is very high. Probability is based on counting things or word in most cases. The probability of a word depends on the previous word which is called Markov assumption. Unigram looks single item from a given sequence. Bigram is called first-order Markov model which looks one word into the past and trigram is second-order Markov model which looks two words into the past and similarly an N-gram language model is N-1 Markov model which looks N-1 words into the past (Daniel Jurafsky and James H. Martin, (2000)). Thus the general equation for this N-gram approximation to the conditional probability of the next word in a sequence, w1, w2... wn,

We compute the probabilities of a test sentence from trained corpus which is designed very carefully. Suppose a word "Oromiyaa" word occurs 400 times in a corpus of one bag of words. Then the estimated probability for the word "Oromiyaa" P ($w_n|w_1$, w2…w_{n-1}. There are different type of n-grams, these are Unigram, Bigram and Trigram. The brief descriptions of these algorithms discussed as follows:

2.6. Unigram, Bigram and Trigram

The most successful form of statistical prediction examines the probabilities of words given the previous word or two words in the sentence. The probabilities of word one given in the sentence word in the corpus is **Unigram**. A lexicon, which stores the probabilities of word-pairs, is known as a **bigram**. One, which uses word-triples, is known as a **trigram** although it drastically increases the required number of entries in the corpus, since the number of combinations of three words is much higher than that of just two. In fact, it is possible to extend the number of words in probable sequences to as many as one wish although the combinations will soon become unmanageable (M. Damashek. Gauging, 1995). The term used for this form of lexicon is an n-gram where n is the number of words used in the probability sequences. The use of n-grams is a methodology which is common in word prediction and is employed to narrow down the bag of words from which the system can make its choice. It is a successful method in both prediction and completion since it relies on the fact that most languages generally contain standard word orders, which are often repeated. In addition, the use of this technique embodies the syntax of a given topic of discussion without having to directly address any specific problems of either. However, the smaller *n* becomes the technique to represent the concepts accurately. One method, which has been experimented with in speech recognition, is the use of long-distance n-grams. This approach spots key words in the sentence, which usually give rise to certain grammatical constructs and therefore certain word sequences. Examples of such key words are relative pronouns before relative clauses or a preposition before a prepositional phrase in case of English language. Knowledge of the presence of such a key word allows the system to increase the probabilities of normal *n*-grams, which are relevant to the grammatical construct associated with the given key word. Thus in order to use an n-gram for prediction for disabled persons, it must have been trained on appropriate transcriptions of entered letters

preferably from the conversations of disabled persons and even more preferably, from the person for which the system is intended.

2.7. Related Literature Review

According to Al-Mubaid(2006) word prediction is an important NLP problem which predict the correct word prediction utilities. In his study (work) he identifies WP using context based features of correct predictions. The technique used was MI, and $X^{2\ \text{combination}}$ of a top performer in machine learning, SVM, selection techniques. The result achieved was impressive.

The work that done by Keith Trnka, (2005) on the keystroke savings limit in word prediction for AAC show that WP is enhancing the communication ability of persons with speech and language impairments. The approach used was statistical language modeling techniques to word prediction. The approach presented here a pure n-gram based method. Additionally, integrated syntactic knowledge into their language models which was found to improve prediction somewhat, added topic modeling onto an n-gram approach, which also had mixed results. In his work he demonstrates the measurements settings that enhance the evaluation of WP. The obtained result was considerable and accurately interpreted.

The research conducted by Guidsa, (2013) on word prediction text entry for Afan Oromo on mobile phone which is limited to only hand devices. His work presents a word prediction approach based on context features and machine learning. As the result it shows that the accuracy performance of his system 56.8%. The technique used was n-gram (unigram, bigram, trigram, four gram) which has a problem when there is lack of trained data like Afan Oromo especially, four gram needs huge size of corpus.

As (Md. Masudul Haque, 2015), Bangla languages also need to have word prediction as he has done research on Automated word prediction in Bangla language using stochastic language model. The researcher used Java program as tool in order to perform their experiments using unigram, bigram, trigram, back off and deleted interpolation language models. Their work starts with a training corpus of size 0.25 million words. The corpus has been constructed from the popular Bangla newspaper the daily “Prothom Alo”. The corpus contains 14,872 word forms. Researcher has taken some test data in a file from the training

corpus. After that comprehensive unigram, bi-gram and tri-gram statistics have been automatically generated and stored the sentences with predicted words in a file. Thus researcher has constructed *N*-gram models of word prediction by counting frequencies of words in a very large corpus, i.e. database and determines probabilities using *N*-gram (Md. Masudul Haque, 2015).

(Arora, 2007) Conduct another research on Context Based Word Prediction for Texting Language**.** The researcher uses machine learning algorithms tools to predict the current word given its code and previous word's Part of Speech (POS). The code, word and its POS are three random variables in the model. The dependency relationship between these variables can be modeled in different ways and the study analyzes and presents a discussion of pros and cons of each modeling approach (Arora, 2007).

(Ananthakrishnan, 2008) Presents a corpus based word prediction system, and based on a lazy learning algorithm, K-Nearest Neighbors Classifier On Word Prediction using a Memory based Learner. The algorithm, given a corpus and a sequence of words, tries to predict the succeeding word. The features are the preceding 'N' words and the distance measure is weighted by the proximity to the word to be predicted. Which means that distance for the first preceding word is N, the second preceding word is N-1 and so on. This can be found out using a K-Nearest Neighbor (K-NN) classifier and an overlap metric for calculating the distance between the test sample features and the features for the training samples.

Japanese word prediction, the entire implementation is written using the .net framework developed by Microsoft based on the common language infrastructure (CLI) in Windows is also the best review developed by (Lindh, 2009). The framework was chosen due to its automatic memory management capabilities which enable more agile software development and less test time. The framework also has an excellent library of stock data structures and GUI creation tool sets which saves a lot of time developing an interface to the main prediction engine.

The researcher investigates the use of topic models, such as probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA), for word completion tasks (Elisabeth Wolf, 2006) On the Use of Topic Models for Word Completion. The advantage of using these models for such an application is twofold. On the one hand, they allow us to exploit

semantic or contextual information when predicting candidate words for completion. On the other hand, these probabilistic models have been found to outperform classical latent semantic analysis (LSA) for modeling text documents. Researchers describe a word completion algorithm that takes into account the semantic context of the word being typed and also present evaluation metrics to compare different models being used in the study.

Additionally, (Roth, 2003) has use SNoW learning system for the task of word prediction on a classification approach to word prediction, a representation is learned for each word of interest, and these compete at evaluation time to determine the prediction. The SNoW architecture is a sparse network of linear units over a common pre-defined or incrementally learned feature space. It is specifically tailored for learning in domains in which the potential number of features might be very large but only a small subset of them is actually relevant to the decision made.

In addition to the above, (Sun, 2014) has done another word prediction on Predicting Chinese Abbreviations with Minimum Semantic Unit and Global Constraints, the researchers use an integer linear programming (ILP) formulation with various constraints to globally decode the final abbreviation from the generated candidates. Experiments show that method outperforms the state-of-the-art systems, without using any extra resource and the researcher use a labeling method which uses four tags, “KSFL”. “K” stands for “Keep the whole unit”, “S” stands for “Skip the whole unit”, “F” stands for “keep the First character of the unit”, and Label “L” stands for “keep the Last character of the unit”.

Generally, the purpose of this study specifically is to develop word prediction and completion prototype for local language which is desktop oriented and reduce the problem of misspelling while writing advertisement, help disables, saves time and any Afan Oromo word productions. In order to develop the model n-gram technique was used which is corpus based and unsupervised machine learning due to lack of trained data for training.

Chapter Three

3. Overview of Afan Oromo Language

3.1. Introduction

Afan Oromo, also called Oromiffaa is one member of the Cushitic branch of the Afro-Asiatic language family (Tullu Guya, 2003). It is the third most widely spoken language in Africa, after Hausa and Arabic. Its original homeland is an area that includes much of what is today Ethiopia, Somalia, Sudan and northern Kenya and some parts of other East African countries (Abera.N, 1988). Currently, it is an official language of Oromia Regional State (which is the largest region among the current Federal States in Ethiopia). It is used by Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 50 % of the total population in 2007 (Girma Debele, 2014). To the writing system Qubee (a Latin-based alphabet) has been adopt and become the official script of Afan Oromo from 1991 (Girma, 2014). Among the major languages that are widely spoken and used in Ethiopia, Afan Oromo has the largest speakers (Abera.N, 1988). It is considered to be one of the five most widely spoken languages among the roughly one thousand languages of Africa (Gragg and Gene. B, 1996).

Afan Oromo, although relatively widely distributed within Ethiopia and some neighboring countries like Kenya, Tanzania and Somalia, is one of the most resource scarce languages (Tilahun Gamta, 1995). It is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic phylum (Abera.N, 1988), unlike Amharic (an official language of Ethiopia) which belongs to Semitic language family.

Although it is difficult to identify the actual number of Afan Oromo speaking societies (as a mother tongue), due to lack of appropriate and current information sources, according to the census taken in 2007 it was estimated that 50 % of Ethiopians are ethnic Oromo (Kula Kekeba Tune, 2007). It is widely used as both written and spoken language in Ethiopia and neighboring countries (Kula Kekeba Tune, 2007). Currently it is used as an instructional media for primary, junior secondary schools and Ethiopian Universities in the region and out of the region like Mekele University. It is also given as a subject starting from grade one to high school throughout the schools in Oromia region. It is also broadcasted via televisions

and radio stations locally and internationally. Furthermore, few literature works, newspapers (Bariisaa, Kallacha Oromiyaa and Oromiyaa), magazines, educational resources, official documents and religious writings are written and published in this language (Girma Debele, 2014)

3.2. Afan Oromo Language Structure

Afan Oromo has a very rich morphology like other African and Ethiopian languages (Tilahun Gamta, 1995). With regard to the writing system, Qubee (Latin-based alphabet) has been adopted and became the official script of Afan Oromo since 1842 (Tilahun Gamta, 1995). The writing system of the language is straightforward, which is designed based on the Latin script. Thus letters in the English language are also in Afan Oromo except the way it is spelled. A detailed description of Afan Oromo Writing System can be found in any text related to the language, but (Mekonnen, 2000) discussed writing system of the language.

3.3. Alphabet of Afan Oromo (Qubee Afaan Oromoo)

Afan Oromo uses Qubee (Latin based alphabet) that consists of thirty-three basic letters, of which five are vowels, twenty-four are consonants, out of which seven are paired letters and fall together (a combination of two consonant characters such as 'CH', 'DH', 'NY', PH, etc). The Afan Oromo alphabet characterized by capital and small letters as in the case of the English alphabet. In Afan Oromo language, as in English language, the vowels are sound makers and are sound by themselves. Vowels in Afan Oromo are characterized as short and long vowels. The complete list of the Afan Oromo alphabets is found on the manuscript by (Tullu Guya, 2003) and (Tilahun Gamta, 1995). The basic alphabet in Afan Oromo does not contain 'p', 'v' and 'z', because there are no native words in Afan Oromo that formed from these characters. However, in writing Afan Oromo language, they are used to refer to foreign words such as "polisii" ("police").

3.4. Afan Oromo Sentence Structure

Afan Oromo and English are different in sentence structuring. Afan Oromo uses subject-object-verb (SOV) language. SOV is the type of language in which the subject, object and verb appear in that order. Subject-verb-object (SVO) is a sentence structure where the subject comes first, the verb second and the third object. For instance, in the Afan Oromo sentence

“Mooneeraan bilisa bahe”. “Mooneeraa “is a subject, “bilisa” is an object and “bahe” is a verb. Therefore, it has SOV structure. The translation of the sentence in English is “Mooneeraa has got freedom” which has SVO structure. There is also a difference in the formation of adjectives in Afan Oromo and English. In Afan 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, namicha gaarii (good man), gaarii (adj.) follows namicha (noun).

3.5. Articles

Afan Oromo does not require articles that appeared before nouns, unlike that of English. In English there are three main semantic choices for article insertion: definite article (the), indefinite article (a, an, some, any) and no article. In Afan Oromo, however, the last vowel of the noun is dropped and suffixes (-icha, -ittii, -attii) are added to show definiteness instead of using definite article. Forexample, “the man” is “namticha” to indicate certainty (Tullu Guya,2003).

3.6. Punctuation Marks

Punctuation marks used in both Afan Oromo and English languages are the same and used for the same purpose with the exception of the apostrophe. Apostrophe mark (‘) in English shows possession, but in Afan Oromo it is used in writing to represent a glitch (called hudhaa) sound. It plays an important role in Afan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels appeared together like “ba’e” to mean (“get out”) with the exception of some words like “ja’a” to mean “six” which is identified from the sound created. Sometimes an apostrophe mark (‘) in Afan Oromo interchangeable with the spelling “h”. For instance, “ba’e”, “ja’a” can be interchanged by the spelling “h” like “bahe”, “jaha” respectively still the senses of the words are not changed.

3.7. Conjunctions

Conjunctions are used to connect words, phrases or clauses. In Afan Oromo there are different words that are used as a conjunction. Conjunctions in Afan Oromo are “fi”, “haa ta’u malee”, “garuu”, “akkasumas”. Example: Amboo, Finfinnee fi Jimmaan magaalota Oromiyaati.(Ambo, Finfine and Jimma are Oromian cities.).So these conjunctions are also one type of words so in our study we used as word and give to our system to predict them.

3.8. Word and Sentence Boundaries

In Afan Oromo, like in other languages, the blank character (space) shows the end of one word. Moreover, parenthesis, brackets, quotes are being used to show a word boundary. Furthermore, sentence boundaries punctuations are almost similar to English language i.e. a sentence may end with a period (.), a question mark (?), or an exclamation point (!) (Getachew Rabirra, 2014).

3.9. Word Segmentation

The word, in Afan Oromo “jecha” is the smallest unit of a language. There are different methods for separating words from each other. This method might vary from one language to another. In some languages, the written or textual script does not have whitespace characters between the words. However, in most Latin languages a word is separated from other words, by white space characters (Atelach Alemu Argaw and Lars Asker, 2010). Afan Oromo is one of Cushitic family that uses Latin script for textual purpose and it uses white space character to separate words from each other’s. For example, “Bilisummaan Finfinnee deeme”. In this sentence the word “Bilisummaan”,” Finfinnee” and” deeme” are separated from each other by white space character. Therefore, the task of taking an input sentence and inserting legitimate word boundaries, called word segmentation, is performed using the white space characters.

Chapter Four

4. Methodology

4.1. Introduction

This chapter presents the proposed method employed in this study. In order to develop word prediction model for Afan Oromo we followed various steps which involve: (a) text preprocessing which take input and corpus, tokenize to remove stop words and perform normalization. To preprocess the developed corpus we have used Java Net Beans 8.0.2 tools. The reason why we used this tool is because Java is free and help to develop user interface. The other one is to (b) extract words to providing the clue about the predicted term using two techniques (Frequency and Recency). Based on the frequency of co-occurrences of the words the most frequent will be listed to select the candidate one. On the other hand, the word will be listed which are recently accessed, then select if it is the candidate words and select to complete words. In order to guess the next word or term the users should press one or more of the starting letter of the term. Hence we have used the most frequently occurring words in the corpus to predict. The detailed description of the algorithm is discussed in Section 4.5.1.

The architecture (pipeline) of the system with the underlying steps is presented in figure 4.1 below:

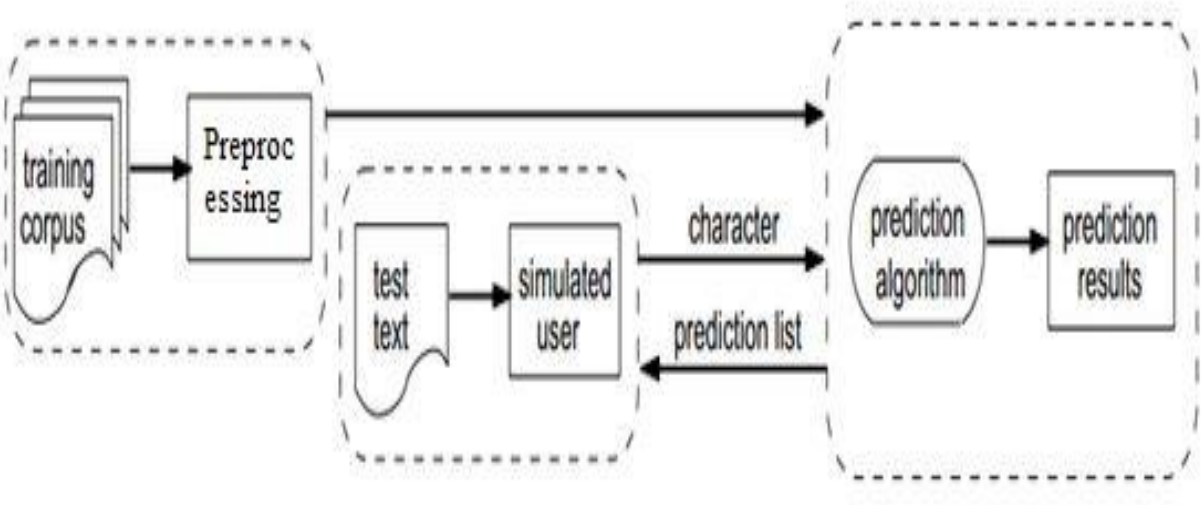

Figure: 4.1. System Architecture

4.2. Corpus

This section deals with the collection of data and preprocessing the collected data. For this study, we have prepared new corpus for the WP of Afan Oromo. The size of the corpus is an important factor in the quality of the predicting, with the general message that more data is better data (Rada , 2009). However, in such a small work, it is very difficult to obtain standardized dataset for under resourced language. The procedure for collecting and preprocessing corpus is described here below.

4.2.1. Corpus Acquisitions and Preparations

The corpus acquisition and preparation are set of techniques required for gathering and compiling data for training and testing algorithms. The lack of resource has led the researcher to explore the use of un annotated raw corpus to perform word prediction model. It should be noted that unsupervised prediction cannot actually label specific terms as a referring to a specific concept that would require more information than is available.

The mechanism to acquire resource is to use the data from various sources and hence it is not previously used in any research on Afan Oromo. The collected data is machine readable free text. It is collected from newspapers (such as the weekly newspaper Bariisaa, and Kallacha Oromiyaa and Oromiyaa. Bariisaa that are published once in two weeks), linguistics, news (ORTVO), government and official websites. Moreover, ORTVO found in Adama releases daily news through radio and television broadcast and on its official website .

To reduce the data sparsely we have used data from these sources since they are believed to represent texts addressing various issues of the language. Actually, the collected data was not directly used for the purpose. We have applied preprocessing (tokenize, stop word and normalize) techniques before it is used. Hence, the acquired corpus is a free sentence of the corpus; we have prepared and filtered to make it ready for the study in the following Sections.

4.2.2. Corpus Preprocessing

The preprocessing step includes tokenizing, punctuation removal by Java net beans 8.0.2 version. here we tokenize the whole document that composed of sentences to the words. In case of the punctuation to come with the word prediction we must remove punctuation from the whole documents , hence it has no effect on the meaning of the words.

A. Tokenization

The corpus which is a set of sentences first tokenized into words. Since, Afan Oromo uses Latin alphabet and hence the sentences can split using similar word boundary detection techniques like the use of white space in English. In this work tokenization is needed for some purposes that are to select words, stop word removal and for further steps. For instance, if we have the sentence *akkanatti hin haftu ni dabarti* in our corpus. We have tokenized into set of words on the white space, like *akkanatti, hin, haftu, ni* and *dabarti.*

B. Punctuations Removal

After tokenization takes place, we have removed Afan Oromo punctuation; hence it has no effect on meaning of the words. In this work, punctuation removal is used to remove punctuation from the selected words because the absence or presence of punctuation marks has no contribution to identify appropriate prediction. Not all tokenized words and punctuations are necessary for this work hence; the symbols have no effect on prediction and completion. For instance, Symbols such as (‘!’, ‘?’), as we it discussed in table 4.1 below. Since stop words do not have significant discriminating powers in the prediction of words; we filtered stop-words list to ensure only content bearing words are included. Few lists of stop words in Afan Oromo are shown in the table 4.1 below:

Table 4.1. Punctuation mark.

No	Punctuations	Description in English
1	; :	These symbols are helps to identify one word from other, so the absence or presences of these symbols are not much affect our system.
2	“ ”	This symbol is called quotation mark, according to the structure of this language it has on impact prediction.
3	?	This symbol is question mark which used to ask question but in case of this study the absence of this symbol have no effect.
4	&	This is conjunction, also no effect on prediction
5	()[]{}	These symbols are brackets which are less contribution for prediction
6	!	Exclamation mark
7	+-*/	These symbols mostly they has no contribution on the word autocomplete or prediction

4.3. Implementation Tools

To perform this experiment, the algorithm was implemented in Java NetBeans IDE 8.02 which runs on the prepared corpus. Java is a general purpose and open source programming language. Moreover, it is optimized for software quality, developer productivity, program portability, and component integration. Nowadays Java is commonly used around the world for Internet scripting, systems programming, user interfaces, product customization, numeric programming, and more. It is generally considered to be among the top four or five most widely-used programming languages in the world (T.Gudisa, 2015).

4.4. Testing

The WP system was trained on un annotated corpus, which constitutes predicted words. The work was tested by using most frequent 15 words collected from the public via questions (in an Appendix A) from random native speakers of the language. To this end, we collected the words from 10 individuals and selected the top 15 frequent terms. The selected predicted words are listed below:

No	Words	Total Frequency counted in aCorps
1	Afaan	135
2	Gaaffilee	11
3	Qorannoo	74
4	Oromoo	84
5	Dubartoota	27
6	Gumaa	41
7	Gadaa	130
8	Qulqulluu	81
9	hayyoota	4
10	Kabajaa	5
11	Uummata	49
12	Tokkummaa	58
13	Finfinnee	9
14	Oduu	5
15	Aannan	13

Table 4.2. Test set

4.5. Algorithm Approaches

In this study we used an approach to predict and auto complete words. To this end, the machine which is unsupervised approach was trained on the free text corpus. As discussed on Figure 4.1. System architecture, after the machine trained on the corpus, the system preprocessed the corpus as is discussed in section 4.1.2. Then, simply the users can press the first letter or letters of his candidate word. Assume that the user enter the first letter of a word then the system will count the frequency occurrence of words started by entered letter and make a rank at first. And also the system can show the recently predicted words as first predicted word. Finally, the system shows the list of predicted words. Then the user can select the words, to make auto complete and meaningful terms. The more description of this algorithm as follows:

4.5.1. N-gram Algorithm

As discussed on Figure 4.1, a useful part of the knowledge needed to allow word prediction can be captured using simple statistical techniques. In particular, we'll rely on the notion of the probability of a sequence (of characters, letters, and words). N-grams are token sequences of length *N* and a given knowledge of counts of N-grams such as these; we can guess likely next words in a sequence. More formally, we can use knowledge of the counts of N-grams to assess the conditional probability of candidate words as the next word in a sequence or we can use them to assess the probability of an entire sequence of words. It turns out that being able to predict the next word (or any linguistic unit) in a sequence is an extremely useful thing to be able to do. We can model the word prediction task as the ability to assess the conditional probability of a word given the previous words in the sequence $P(w_n|w_1,w_2 \ldots w_{n-1})$.

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned}$$

The probabilities in a statistical model like an N -gram come from the corpus it is trained on. This training corpus needs to be carefully designed. If the training corpus is too specific to the task or domain, the probabilities may be too narrow and not generalize well to new sentences. If the training corpus is too general, the probabilities may not do a sufficient job of

reflecting the task or domain. In the unigram sentences, there is no coherent relation between words, and in fact none of the sentences end in a period or other sentence-final punctuation. The bigram sentences can be seen to have very local word-to-word coherence (especially if we consider that punctuation counts as a word).

4.6. Evaluation Method

To our knowledge, there were no previous standard Afan Oromo word prediction dataset for evaluation as presented in Section 4.2. For this reason, we did not evaluate against the other systems. In this work, the evaluation was undertaken on the basis of precision and recall. Precision is defined as the percentage of correctly predicted words out the total of predicted words. Recall is defined as the percentage of correctly predicted words out of the total number of predicts words (Flickinger, 2015).

$$\text{Precision (\%)} = \frac{\text{\# correctly predicted Words}}{\text{\# predicted Words}}$$

$$\text{Recall (\%)} = \frac{\text{\# correctly predicted Words}}{\text{\# Total Number of entered Words}}$$

Chapter 5

5. Experimentations Results, Discussion and Evaluation

5.1. Introduction

This chapter deals about the implementation and the training data as well as the experimentation of the work. Accordingly, the researcher developed the implementation of the user interface prototype and the data used for its implementation and the results in detail. To this end, the evaluation of the study and the discussion used were discussed.

5.2. Data

In this section, we describe the dataset/corpus used in experiments, the manner in which the corpus was collected as well as the preprocessing techniques as it discussed in Section 4.2.

In this work, two types of data were used: (a) the small list of words to test the algorithms. Fifteen (15) highly frequent words were selected from the language speakers using the procedure described in Appendix A, (b) the big corpus containing thousands of sentences to extract words; hence, the compiled corpus for the experiment is a free text. It is composed of a set of 2,242 sentences with 37,272 token words. This corpus is prepared for the purpose of this study as there is no standard corpus for Afan Oromo language. However, it is unlabeled data and never used in any researcher and it needs to be developed further (as discussed in Section 4.2). Words occurring zero times in the corpus were used as irrelevant words. Punctuation marks were removed from the corpus, so they do not appear in the prediction words. The test data of this work is collected from the speakers of the languages in order to avoid bias we collect many possible words as it discussed in Appendix A.

5.3. Experimentations of the Word Prediction

In order to implement the word prediction and auto-completion system, the researcher used the Java NetBeans IDE 8.02 version. As described in chapter four, the tool used to develop the prototype and the development environment was presented in detail. The purpose of the developed interface was to enable the users to type the character of the needed word that the proposed system will run on the prepared corpus. The interface to perform this operation is

shown in Figure 5.1 below. The following figure shows the user interface of the word prediction used during the experimentation.

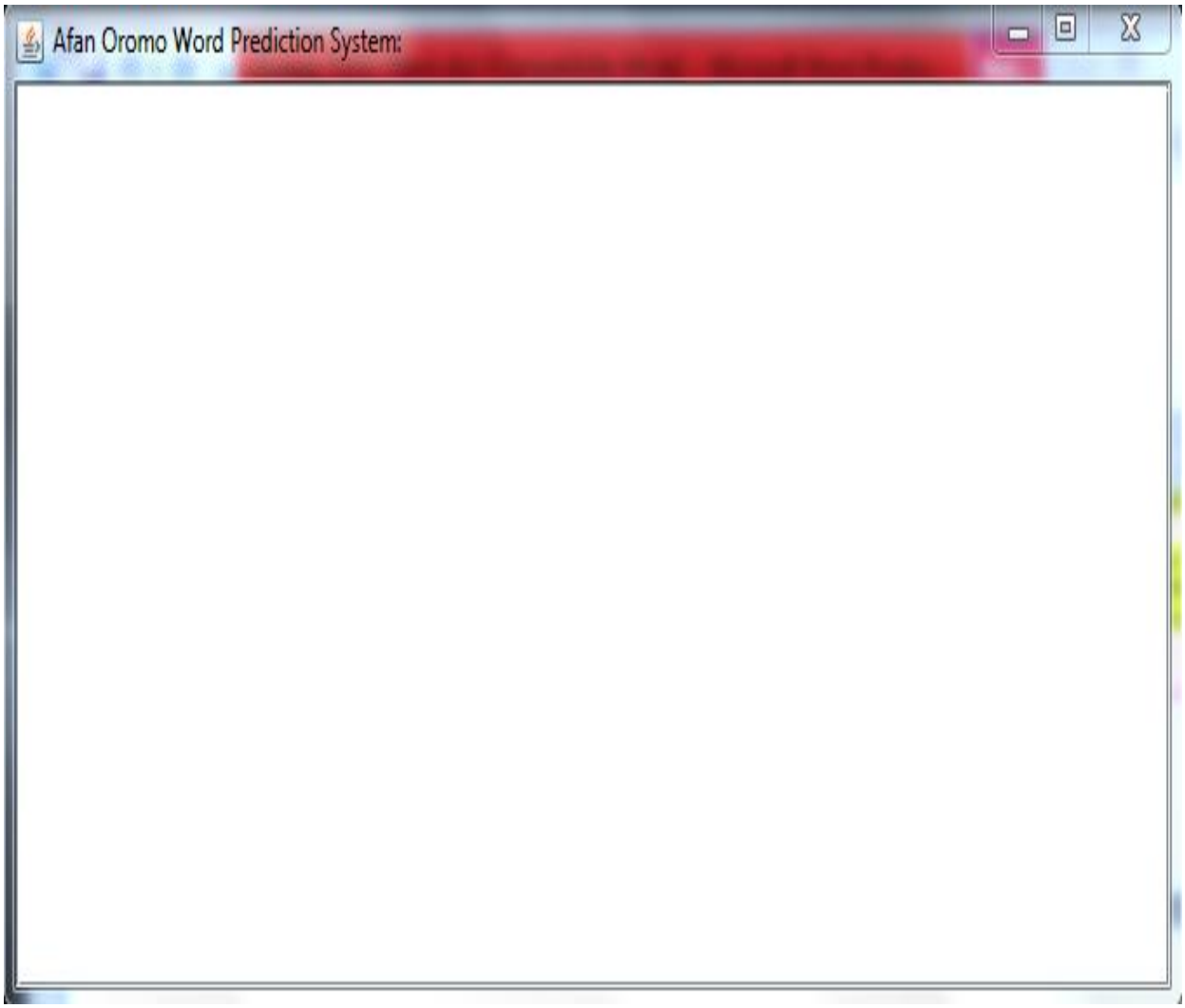

Figure 5.1. User Interface for Word Prediction

As the above figure 5.1 shows, the interface prepared for the place where users can type in their interest of the first character of word. Based on this when the user type the first character of the word the system will predict the list of words that are started by the typed character. If the predicted list does not contain the needed word then the user has to press the next character of that word. Unfortunately our system cannot support the words that are not available in our corpus. Hence, the frequency occurrence of the word in the corpus was zero mean that the word was not found in the corpus.

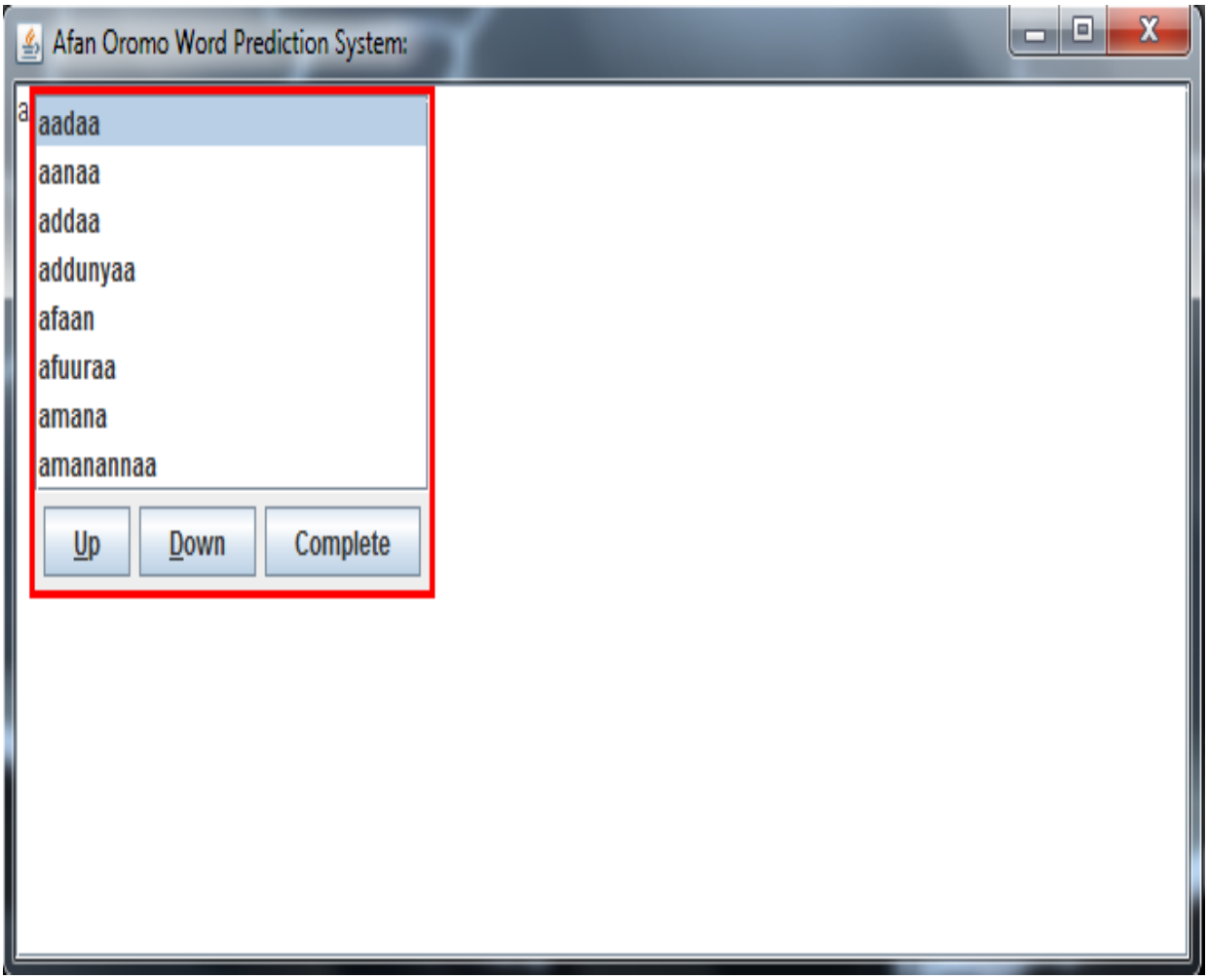

Figure 5.2. Auto-completion GUI

The above figure shows the visible graphical user interface for word prediction. A writer must write character of a word to get list of predict words. After inserting the first letter of the word, a box that contain the list of predict words will appear on the Screen Shot of word prediction User Interface button then the user will select by using Up and Down button to select the appropriate word and lastly by clicking Complete button action is applied to initiate the auto completion task. The algorithms will count the frequency or recency of a word to provide it. As mentioned above, based on the frequency value, the system predicts the best words.

5.4. Results and Discussion

This chapter presents the result and discussion of the work. As discussed above, the work essentially constitutes corpus based word prediction system that is based on machine learning algorithm, N-gram model. The algorithm, given a corpus and a sequence of words, tries to predict the succeeding word. In the current context, no grammar model or parse trees are used in order to gauge the morphology of the words to be predicted. For every word to be predicted, the algorithm needs to scan through the entire training corpus.

As discussed in previous chapters, word prediction is an interesting area with applications in main domains, speech recognition and Augmentative and Alternative Communication tools. It is often not the aim onto itself but a way of augmenting the ability to recognize speech and enhance understanding of either written or spoken language. Predictive word systems in place use the frequency-based method and predict the most commonly used words above other possible words (Fazly and Hirst, 2003).

As a technology on the way of growing, many users are tried to use different technology for different purposes. Word prediction is one of the technologies that plays a great role in our day to day activity especially in relation to processing natural language and that most of us used in our office to edit our files. The writer is intending to produce when creating text documents using varies text editors. Similarly, typing of Afan Oromo language with huge alphabets requires suitable and efficient methods to access all the letters with a QWERTY keyboard. In all application areas, various text entry and speech methods have been suggested to provide a more efficient input of texts with poor spelling and disability (Quiroga *et al.,* 2004).

Based on the experimental results the possible discussion has been given. Auto complete is a word completion tasks so that the user types the first letter or letters of a word and the program provides one or more higher probable words. If the word user intend to type is included in the list user can select it, for example by using the keys (complete button in our case). If the word that the user want is not predicted, the user must type the next letter of the predicting word. At this time, the word choice(s) is altered so that the words provided begin with the same letters as those that have been selected or the word that the user wants appears it is completed. Word prediction technique predicts word by analyzing previous word flow

for auto completing a word with more accuracy by saving maximum keystroke of any user or student and also reduces misspelling. N-gram language model is important technique for word prediction. We use large data corpus for training in N-gram language model for predicting correct word to complete Afan Oromo word with more accuracy (Masudul Haque, *et al.*, 2015).

For training of the model we used the corpus collected from different Medias to develop corpus for the sake of this study only. This corpus consists of the Wikipedia files, newspapers, governmental and non-governmental Medias and any markup is stripped off as discussed in Section 4.1.1 and 4.1.2. Specifically, hence the language was examined as under resources as discussed in section 4.2. Many researchers have also tried to incorporate linguistic knowledge by employing other grammatical information like the parts-of-speech tags, parsing trees, root and stem of the words for English language (Wu D *et al.*, 1999), and (Pereira F *et al.*, 1995) in order to boost performance of the N-gram prediction. However, in this study, such higher level linguistic information which requires annotated data is not used. The only input and features are the sequence of words from the given corpus. As the experiment shows that in most cases, the N-gram model, with N equal to 1 or 2, seem to work the best word prediction for Afan Oromo.

From the finding of this study, if the user enters three more letters the accuracy of prediction system is fewer lists of words hence in Afan Oromo three or more extra letters can make one word. Now if any of these three or four words are not in the training corpus then the frequency of the word will be zero that means, that word does not exist in our corpus. So in this statistical method if we want to consider these words then we need a huge data corpus that must contain all the words of the language. So there may arise the problems like many entries in corpus are with zero frequency and frequency of a word sequence will be very low (Haque, 2015).

As the experiment shows that a typical interface to a word prediction would be a list holding a number of relevant words picked out by the prediction. The list is then modified and pruned from incorrect alternatives as the user types more letters of the word. If the word prediction list contains the correct word the user can select it and go on to type the next word. For example, a user may have typed the word that has started by “a” and a word prediction presents the possible list shown in figure 5.3 below.

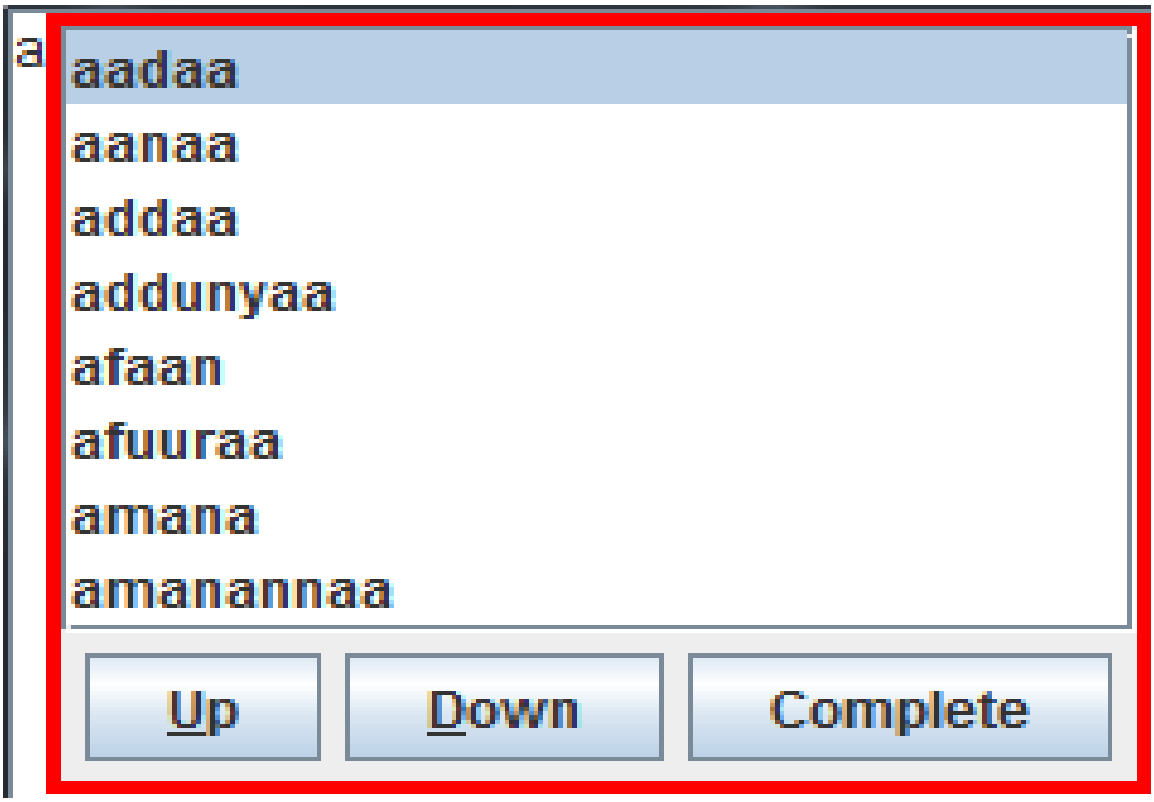

Figure 5.3. Sample Prediction List of Words.

From the figure 5.3 above shows the predicted word displayed on the provided graphical user interface. If the listed words are not listed in the predicted list, the user must know as the word is not present in the prepared corpus, the user can still write the next character to the next word.

Sometimes, hence the dataset collected from multi resources, there are spelling errors. Unfortunately, our system cannot recognize the problem of misspelling due it needs Afan Oromo spelling checker system. However, if the word is spelled correctly and available in the corpus the performance is good.

Based on figure 5.3, the result shows that the predicted words are predicted based on their frequency occurrences which are ranked according to frequency in the corpus as it shown in the Table 5.1 below:

Unigram	Frequency	Rank
Aadaa	103	1
Afaan	92	2
Addaa	83	3
Addunyaa	64	4
Afuura	55	5
Ammana	36	6
Amanannaa	27	7

Table 5.1. Example of word prediction Suggestion List

Assume that the user wanted to type the word "*karaa"*. Simply the user can press the first spelling of the word on provided interface of the users. After that the system will list all the words started by "*k".* The system inside will count the words started by this spelling and rank it at first if it has high frequency of occurrence in the corpus. The user can choose the word directly from the list, for example with the mouse or by pressing the correct spelling. If the word is not in the suggestion list the user can type next letter of the word, in the case "k" and then get a new list with suggestions. Often a space is inserted after the suggested word, which allows the user to continue typing immediately after the prediction. A possible and often used, choice when implementing a word prediction could be to make the insertion of an alternative automatic when there is enough information to make the decision.

Based on the experiment, the result shows that apart from being for people with physical disabilities word prediction can also assist individuals with poor spelling to use a greater variety of words (Hunt-Berg & Rankin, 1994; Laine & Follansbee, 1994). However, although this word prediction can develop writing skills regarding aspects such as word fluency, variety of words and motivation of writing, the tests completed so far have been inadequate in scope and design (DiGiovanni, 1995).

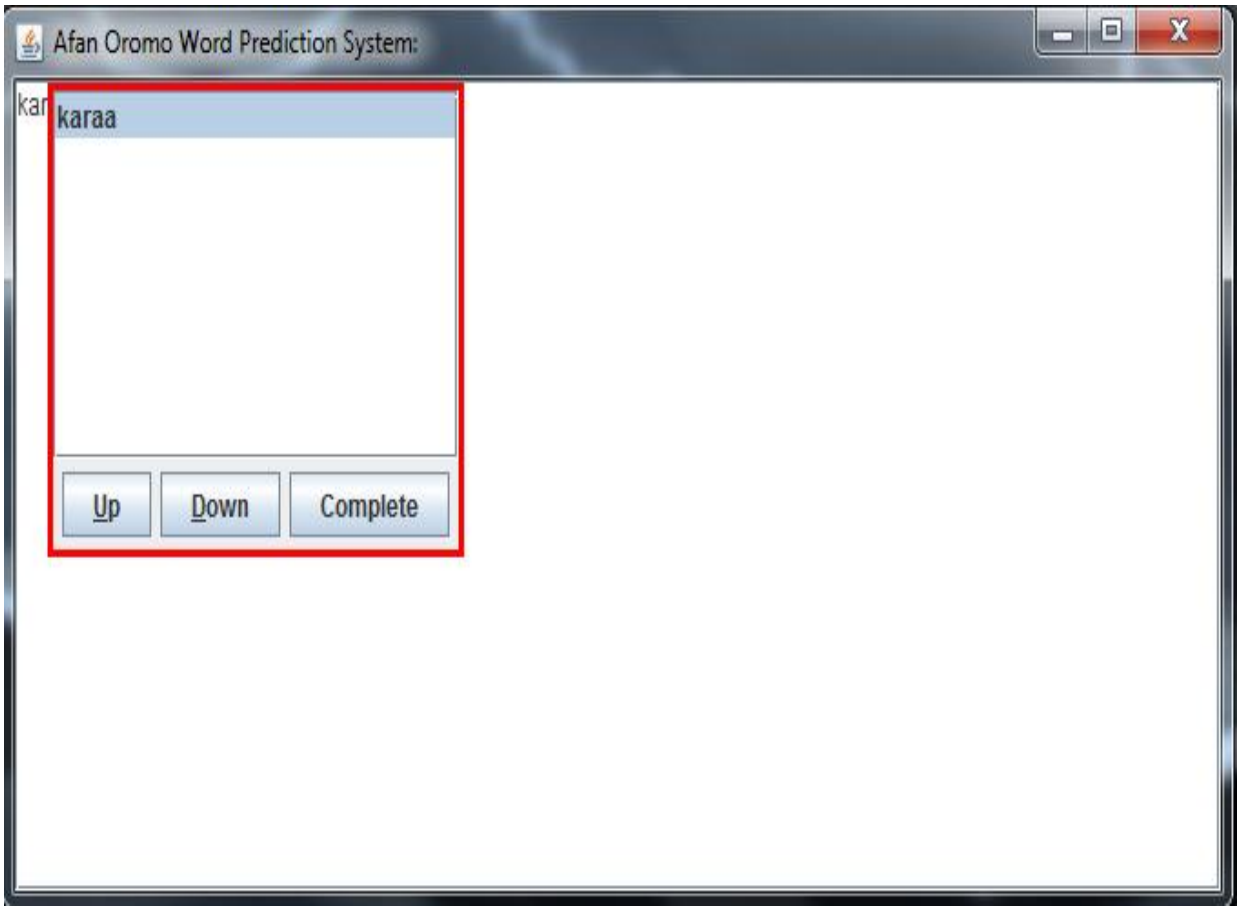

Figure 5.4. Sample Example After Three Spelling Pressed

Based on the above experiment, the system shows that in Afan Oromo correctly predicted after the users enter two letters. To be honest our system can work at one, two, three or four letters even it was work at sequence of words to make a sentence but the accurate performance of the system was not much surprising as the experiment argued, that is 83.84%, 61.4%, 54.8% of unigram, bigram and trigram respectively.

As a result, revealed that word prediction can be a sequential process over time with an input stream of characters. The task is to predict the next character given a string representing the input history. In this study the first character of a string represents oldest input and the last character represents the newest input. For example, given the string “abaaboo” a good guess for the next character would be ‘b’ since ‘b’ follows ‘a’ which is prefix of ‘b’ in the input history. It is well established that there is a close relationship between the tasks of prediction and auto-completion (Marton, Wu, and Hellerstein, 2005).

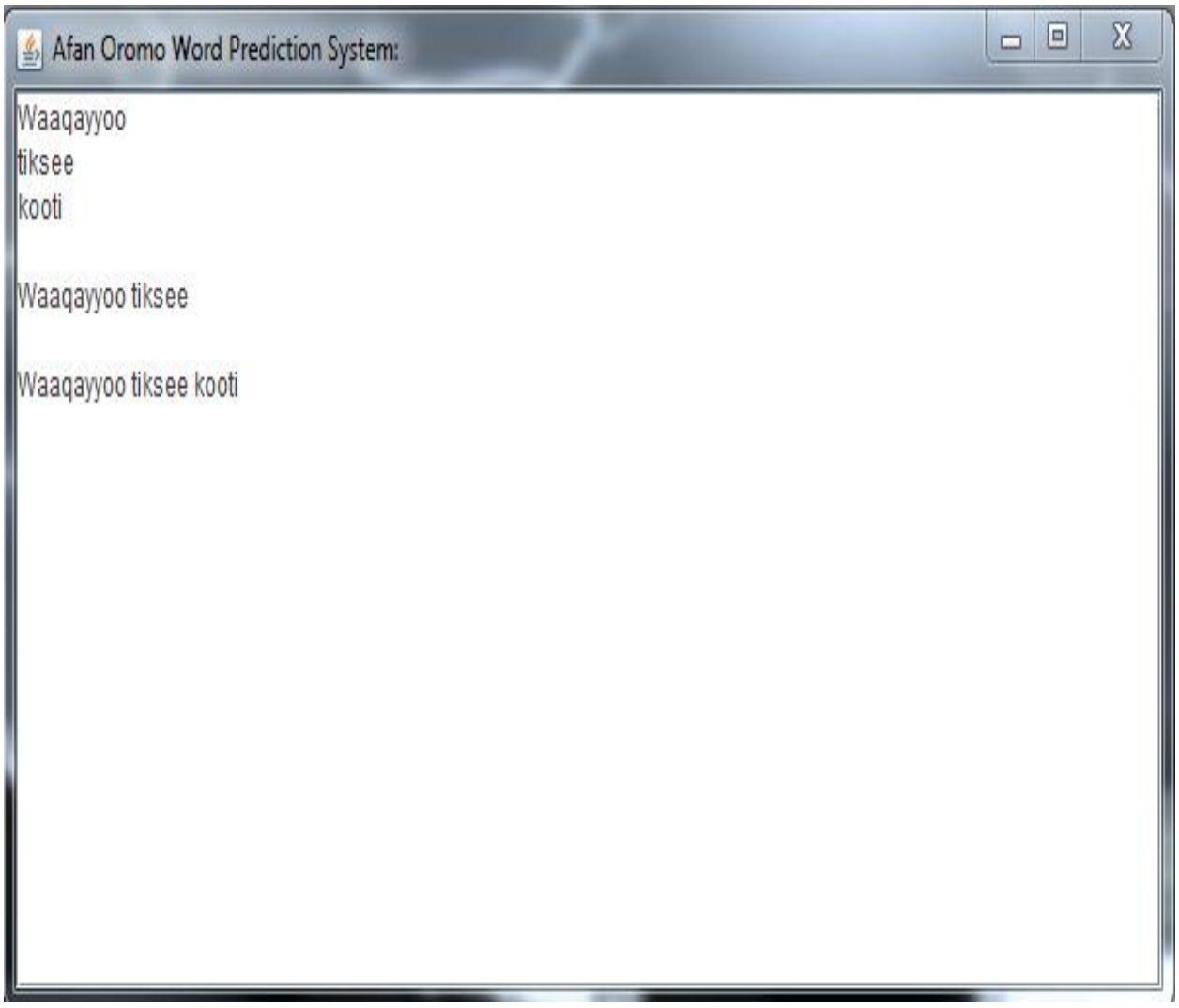

Figure 5.5. Unigram, Bigram and Trigram Sample Example of the Prediction

As the experiment shows that, we predict upcoming words in language especially for Afan Oromo language. So we apply some *N*-gram language model to predict Afan oromo words in a corpus, hence word prediction is a very important and complex task in natural language processing (NLP) to predict the correct word to complete in a very meaningful way (Masudul Haque, *et al* 2015). We use statistical prediction technique such as *N*-gram technique as for example unigram, bigram, and trigram. As it described in above Figure 5.5, the experiment shows that the first split of token word is unigram model like *"waaqayyoo, tiksee, kooti"* and bigram which is tokenized by two words like "*waaqayyoo tiksee*" and the third one is trigram which is tokenized by three sequence of words for example "*waaqayyo tikse kooti*".

There is different way of measuring the performance of the system. In our case the most intuitive performance measure is the accuracy of word prediction. It gives an idea about how many succeeding words can be predicted correctly knowing the previous words.

$$\text{Accuracy} = \frac{\text{Total number of words predicted correctly}}{\text{Total number of words}} * 100$$

Not only is it important to predict the succeeding word theoretically, but sometimes it is possible to calculate the accuracy of the system which is the probability of several possible words is also important. In our case, the frequency of the words helps us to compute the possible probability of the words. The probability of the succeeding word is used in different applications. This measure gives an indication about what is the probability assigned to the correct word as compared to what is the most likely word according to the algorithm.

Table 5.2. performance of the Model

Performance of Algorithms			
Type of words	**Unigram**	**Bigram**	**Trigram**
1-gram	83.84%	61.4%	54.8%
2-gram	75.4%	59.77%	40.7%
3-gram	70.1%	54.9%	31.8%
4-gram	67.9%	52.68%	29.53%
5-gram	61.8%	50.6%	26.89%

As the experiment shows that the computed conditional probability of words based n-gram models are bringing the detail results in above table (see details of table 5.2). As it described in Section 4.4, the nature of algorithms which are n-gram models (unigram, bigram, and trigram) are used as below conditional formula. More formally, we can use knowledge of the counts of N-grams to assess the conditional probability of candidate words as the next word in a sequence and use them to assess the probability of an entire sequence of words. Therefore, simply counting lies at the core of any probabilistic approach. Unfortunately, for most sequences and for most word corpus we won't get good estimates from this method hence the frequencies of most of words are zeros.

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

This can be used to make estimations of how probable of words is computed. The estimation is based on how frequency of the word is in a corpus. The formula used to compute by matching the count result of the multi-word. Language models are based on sequences of words and are the most commonly used in text input today. Just as some sequences of words are more common than others, some sequences of prediction words are more common than others. Gathering statistics on these sequences creates a sort of implicit grammar for the language.

In order to understand the merits of our work in predicting words in Afan Oromo, we need to delve into all results found. We see from Table 5.3. And Figure 5.6 that unigram model performs modestly, whereas bigram and trigram models performance is obviously very poor. The average accuracies of all models deployed are shown in Table 5.3. We see, from Figure 5.6 and Table 5.3, that the average accuracy of unigram and bigram model is close, where the accuracy (46.74%) of the trigram model is the minimum. Some results of predictions by all models deployed are given below:

Table 5.3. Average performance of Prediction Models

No	Models	Average performance of Algorithms
1	Unigram	71.8%
2	Bigram	65.87%
3	Trigram	46.74%

Hence our work is not concern with the context between words, the system was work only prediction particularly at word levels. In the unigram sentences, there is no coherent relation between words, and in fact none of the sentences end in a period or other sentence-final punctuation. The bigram sentences can be seen to have very local word-to-word coherence (especially if we consider that punctuation counts as a word). It means that split word from words by punctuation marks. Due to this reason the result at bigram and trigram model are

very poor hence no need to identify context for prediction in our case. The result of experiment was showed at graph 5.6 below:

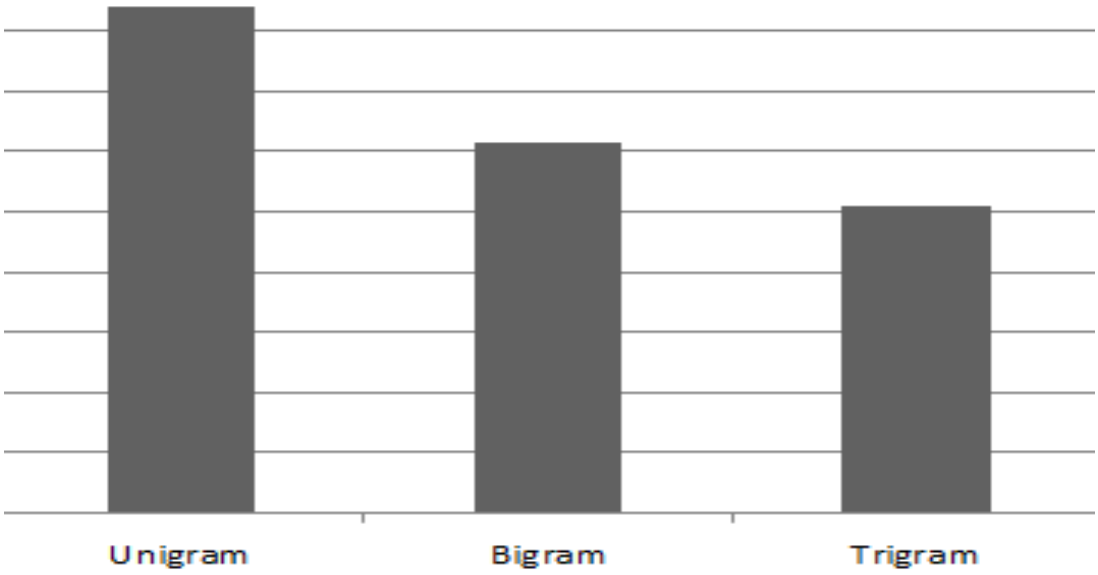

Figure 5.6. Prediction Models Graph

Performance of word predictor depends on some important component such as which language model is used, the average length of the sentence in the language, amount of trained dataset. In our work we use 15 to 20 words in a sentence of the corpus for each execution from 12,062 trained datasets.

As the result of study shows table 5.4 below represented the frequency of words, correctly predicted and incorrectly predicted. The correctly predicted and incorrectly predicted represents the number of words match with user entered and the system required words based on user input. Therefore, the table shows that the number of words that is predicted 11 from the 15 words. However, the system was predicted 10 words correctly. In general, the system result shows that some improvements want to make towards correct prediction.

Total Number of Test Words	Frequency of words in the Corpus	Correctly predicted	Wrongly Predicted
15	**726**	**90%**	**10%**

Table 5.4. Performance Measurement

5.5. Evaluation

Based on the formula that described in Section 4.7, the performance of the work was evaluated. We have used the two metrics of the evaluation method which are precision and recall metrics. The evaluation component exists on its own in order to have the possibility automatically evaluate the system. It allows testing the predictions under different metrics and formally, its interface is similar to that of the GUI component (therefore it could test what the real user would see). As it discussed in section 5.4, the system was evaluated as table 5.5 below:

No	Accuracy of Unigram Performance	
	Precision	**Recall**
Number of Correctly predicted	**90%**	**66.66%**

Table 5.5. Evaluation Method

Chapter 6

6. Conclusion and Recommendation

6.1. Conclusion

The study deals with Afan Oromo word prediction. As discussed in the previous chapters, in this research, unsupervised word prediction for Afan Oromo was used which is important for under resourced languages in general. To this end, we have forwarded the conclusion and recommendation as presented in the following sections.

The overall focus of this research is to investigate Word prediction which addresses the problem of low typing speed and poor spelling and completion. A word prediction guesses what word a user is typing, based on previously written words and prefix letters of the current word. The predictor can then either insert the word with the highest frequency into the text or let the user choose an alternative from some kind of list interface which is then completed. Ideally, this can speed up and ease the user's typing of words. The method used here for developing the word prediction is n-gram models that are unigram, bigram and trigram information from a training corpus of thousands of words to use in the prediction lookup process.

This work is improving and enhancing textual information entry for disabled users has been investigated, with Unsupervised approach and user interface proposed and implemented to facilitate and simplify text input for such people.

For under resourced Ethiopian language like Afan Oromo the unsupervised approach is recommended. Since there is no annotated corpus (even difficult to obtain electronic materials for such language), unsupervised approach plays a great role to predict simply without considering its contexts. The unsupervised approach which is not relies on hand-constructed rules that are acquired from language specialists rather than automatically trained from data.

N-gram based word prediction works well for English but we use for Afan Oromo language which is more challenging to get performance because it depends on training corpus of large data. We use more smoothed corpus data and increase corpus size in future to get higher

performance. Here in our work we use unigram word prediction but we can develop our process to predict a set of word to complete a sentence in a very meaningful way. In the beginning of the work, there were discussed existing solutions for word prediction task with the focus on n-gram language models. Consequently, theoretical background needed for the word prediction was introduced. That included methods for frequency estimates, in order to cope with sparse training data of the language models and approaches to combine more models together.

We also presented an evaluation methodology for measuring the precision and recall of discovered predictions. As the result shows that n-gram model like unigram and bigram are best result for Afan Oromo word prediction.

The experiment presents that for Afan Oromo, it works the best performance at *n* one or two spelling of the words. the study clearly put that the user can press only one or two more spellings to correct way of prediction in Afan Oromo. Hence Afan Oromo has multi vowels to make a sound sometimes it needs to press more spelling to quickly have the correct predictions.

In this work the unsupervised machine learning achieved an accuracy of 90%, 66.66% of precision and recall respectively. We found that better results were achieved using unsupervised machine learning features from the other type of machine learning like supervised machine learning commonly used for word prediction. The results obtained were encouraging as there is lack of resource of the language because of shortage of POS and labeled corpus.

Generally this study will Answer the following research question :-

1. What are the technical applications that are appropriate for Auto completion of word prediction for Afan Oromo word ?

As the researcher was study the technical applications to solve the main problem of the Afan Oromo language Word prediction and Auto completion the Appropriate technical application or implementations tools for the problem, For the implementation tools its better if using the Java Neat Beans software and Machine Learning from machine learning Unsupervised Machine learning is best for the Under resource language like Afan Oromo Unsupervised

Word Prediction methods are based on unlabeled corpora. They do not rely on labeled training text . Unsupervised Machine learning Was it can used on small training data and suitable for under resourced languages and also this approach depends on the available and more reliable training data.

2. What is the current status of Afan Oromo word prediction and auto completion?

Currently Afan Oromo language Status in Word prediction and Auto completion nothing is done previously so One of the existing problems with Afan Oromo language is lack of word prediction and auto word completion services. There is no application that helps user of Languages as they type speedily and write correct spelling. Being not having word prediction, creates multiple problems with Afan Oromo texts literary works such as journals, Books, fictions and some newspapers. In addition to this the speed of typing of many secretaries when they write Afan Oromo texts is very low and misspelled word that create miscommunication.

3. To what extent Auto completion of word prediction for Afan Oromo increases the consistent use of the language?

The main purpose of developing the word prediction and Auto completion prototype for Afan Oromo language is to overcome the problems that happens in the language user and new user of the language, The user of this language (Secretaries) due to absence of the word prediction when they write Afan Oromo texts their speed of typing is very low and misspelled word that create miscommunication. Therefore Auto completion of word prediction for Afaan Oromo will Increase or improve in terms of typing Speed, correcting the misspelled word and also make the language itself technology.

6.2. Recommendation

The study shows that this method of allowing some words in the context to differ from the training models may enhance the word prediction capability as compared to the traditional N-Gram approach.

The following recommendations are forwarded based on our findings with regard to the developments of resources and future research directions to WP for Afan Oromo:

It is better if it incorporates linguistic knowledge by employing other grammatical information like the parts-of-speech tags, parsing trees, root and stem of the words. Researches in WP for other languages use knowledge resources like POS, lexicon. In this study, we faced a significant challenge as Afan Oromo lacks those resources. Taking into account their contribution to WP and other research concerned institutions should develop these resources. For other language a standard prediction annotated data are available for WP research and for testing a standardized WP system. The researcher doesn't have such data for Afan Oromo language. So, there need to be an initiative to prepare the data for WP research. For the under resourced Ethiopian language like Afan Oromo, the linguistic (POS) was made better result, it needs linguistic background knowledge of the language. All the result showed that the techniques are fairly successful and effective in the prediction task. Thus, more research work should be exerted to carry out further improvements on the performance of these techniques. The size of the corpus used for this research was limited and doesn't has good quality. These limitations affected the accuracy of word prediction because if resources used are small, we may not be able to find all possible prediction. Therefore, some work should be done with large and high quality corpus to minimize these problems.

Future research directions for WP in Afan Oromo include:

- In addition to corpus based approach, there are also knowledge source based which are used for WP in another language and found a good result like POS, annotate data. These approaches need to be investigated for Afan Oromo as well.

7. References

Al-Mubaid H. (2003). Context-based word prediction and classification. *Social and Humanistic Computing*,, PP.385–388.

Al-Mubaid, H. (2003). Application of word prediction and disambiguation to improve text entry for people with physical disabilities (assistive technology). *Int. J. Social and Humanistic Computing*, PP.14-15.

Ananthakrishnan, G. (2008). Word Prediction using a Memory based Learner.PP. 1-6.

Arora, S. A. (2007). Context Based Word Prediction for Texting Language.PP. 2-8.

Boote, D. &. (2005). On the centrality of the dissertation literature review in research preparation. *Educational Researcher* ,PP. 3-15.

Cockburn, A. and Siresena, A. (2003) Evaluating Mobile Text Entry with the Fasta Keypad. People and Computers XVII (Volume 2): British Computer Society Conference on Human Computer Interaction, Bath, England, PP.77–80.

Elisabeth Wolf, S. V. (2006). On the Use of Topic Models for Word Completion. PP.1-12.

Ermias.A, D. T. (2011). Designing a Rule Based Stemmer for Afaan Oromo Text. *M.Sc. Thesis*.

Fazly, A. and Hirst, G. (2003) 'Testing the efficacy of part-of-speech information in word completion', Workshop on Language Modeling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics, Budapest.

Gudisa, T. (2013). Design and Implementation of Predictive Text Entry Method for Afan Oromo on Mobile Phone. *master thesis*, PP.62-65.

Girma, D. (2012). Afan Oromo news text summarizer. *master thesis*,

Keith Trnka, D. Y. (2005). The Keystroke Savings Limit in Word Prediction for AAC. 2-5.

Lindh, F. (2009). Japanese word prediction. *thesis*, PP.8-20.

Md. Masudul Haque, M. T. (2015). automated word prediction in bangla language using stochastic language models. *International Journal*, PP.4-9.

Muzeyn Kedir, A. (2012). ''Development of stemming algorithm for Silt'e language text. *MSc thesis faculty of informatics, Addis Ababa University, Addis Ababa*.

Marton, Y., Wu, N., and Hellerstein, L. 2005. On compression-based text classification. In Proceedings of the European Colloquium on IR Research (ECIR), PP. 300-314.

N. and Veronis, J. (1998). Word Sense Disambiguation. *The State of the Art. Computational Linguistics*.

Pereira, F. C., Singer, Y., Tishbi, N., 'BeyondN-Grams', Proc. Symposium of Foundations of Artificial Intelligence, 1995

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*,PP. 130–137.

Quiroga, L., Crosby, M.E. and Iding, M.K. (2004) 'Reducing cognitive load', Proceedings of the 37th Hawaii International Conference on System Sciences, PP.319–334. Richard .P, M. (2014). Python Software Foundation.

Roth, D. a. (2003). A classification of approach to word prediction.PP. 126-130.

Sun, L. Z. (2014). Predicting Chinese Abbreviations with Minimum Semantic Unit and Global Constraints. *Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1-10). Doha, Qatar: Ministry of Education, China.

Sutherland, B. M. (2004). Predictive Text Entry in Immersive Environments. *Proceedings of the IEEE Virtual Reality* , PP.241.

Workineh Tesema (2015) Towards The Sense Disambiguation Of Afan Oromo Words Using Hybrid Approach, Jimma University, Jimma, Ethiopia.

Wu, D., Sui, Z., Zhao, J., 'An information-based method for selecting feature types for word prediction', Proc. Eurospeech, Budapest, 1999.

Appendix A: 15 test words

The questions are prepared and converted to Afaan Oromoo language to make suitable to collect the words prediction test from public.

Saala: _____ Umurii:_____

Sadarka Barumsa:

kutaa 1-5 ☐ kutaa 6-12: ☐ Barataa Yuuniversitii:☐

Barataa Kolleejjii: ☐ Barsiisaa: ☐ Digirii: ☐ Maastersii ☐
Q/Bulaa: ☐ Kan biraa: ☐

1. Jechoota hiikka lamaa ol qaban (words) kan naannawwaa keessanitti fayyadamtu naaf tarreessi?

Lakk	**Jechoota**	**Baay'ina Hiikkaa**	**Hiikkaa**
1	Afaan	2	Qooqa,Qaama nama Afaan
2	Gaaffilee	1	gaaffii
3	Qorannoo	1	Qorqnnoo
4	Oromoo	1	Saba Oromoo
5	Dubartoota	1	Koorniyaa,Dubartoota
6	Gumaa	1	Gumaa
7	Gadaa	1	Gadaa
8	Qulqulluu	1	Qulqulluu
9	hayyoota	1	hayyoota
10	Kabajaa	1	Kabajaa
11	Uummata	1	Uummata
12	Tokkummaa	1	Tokkummaa
13	Finfinnee	1	Finfinnee
14	Oduu		Oduu

15	Aannan		Aannan

Appendix B: Code that are used in the study

```
//autocomplete/AutoTextComplete.java
package wp;
import javax.swing.*;
import javax.swing.text.*;
import java.awt.event.*;
import java.awt.*;
public class AutoTextComplete extends JWindow
        implements KeyListener, FocusListener, ActionListener {
    JTextComponent parent = null;
    JList lst = null;
    java.util.TreeSet<String> val = new java.util.TreeSet<String>();
    java.util.Vector<String> tempVector = new java.util.Vector<String>(1, 1);
//declarations for table
    JTable tableParent = null;
    int activeColumn = 0;
    StringBuffer typed = new StringBuffer();
    public AutoTextComplete() {
        lst = new JList();
        lst.addKeyListener(this);
        lst.addFocusListener(this);
        this.getContentPane().add(new JScrollPane(lst), BorderLayout.CENTER);
```

```
        setButtons();
        ((JPanel)
this.getContentPane()).setBorder(BorderFactory.createLineBorder(Color.red,
3));
        lst.setToolTipText("Press F5, F6 to navigate, F7 to Complete");
    }
    public AutoTextComplete(JComponent jc) {
        this();
        if (jc instanceof JTable) {
            tableParent = (JTable) jc;
        } else {
            parent = (JTextComponent) jc;
        }
        jc.addFocusListener(this);
        jc.addKeyListener(this);
    }
    public AutoTextComplete(JTable t, int col) {
        this(t);
        activeColumn = col;
    }
    public void setActiveColumn(int col) {
        activeColumn = col;
    }
    private void setButtons() {
```

```
        JButton b[] = {new JButton("Up"), new JButton("Down"), new
JButton("Complete")};
        char c[] = {'U', 'D', 'S'};
        String txt[] = {"Move up (F5)", "Move down (F6)", "Complete (F7)"};
        JPanel p = new JPanel(new FlowLayout());
        for (int i = 0; i < b.length; i++) {
           b[i].addActionListener(this);
           b[i].setMnemonic(c[i]);
           b[i].setToolTipText(txt[i]);
           p.add(b[i]);
        }
        this.getContentPane().add(p, BorderLayout.SOUTH);
    }
    public void addItems(String it[]) {
        val.addAll(java.util.Arrays.asList(it));
    }
    public void addItems(java.util.List<String> init) {
        val.addAll(init);
    }
    public void setItems(String it[]) {
        val.clear();
        val.addAll(java.util.Arrays.asList(it));
    }
    public void setItems(java.util.List<String> init) {
```

```
        val.clear();
        val.addAll(init);
    }
    private void moveUp() {
        if (!this.isVisible()) {
            return;
        }
        int index = lst.getSelectedIndex();
        lst.setSelectedIndex(index = index > 0 ? index - 1 : tempVector.size() - 1);
        lst.validate();
        lst.scrollRectToVisible(lst.getCellBounds(index, index - 1 < 0 ? 0 : index -
1));
    }
    private void moveDown() {
        if (!this.isVisible()) {
            return;
        }
        int index = lst.getSelectedIndex();
        lst.setSelectedIndex(index = index < tempVector.size() - 1 ? index + 1 : 0);
        lst.validate();
        if (index == 0) {
            lst.scrollRectToVisible(lst.getCellBounds(0, 1));
        } else {
```

```
            lst.scrollRectToVisible(lst.getCellBounds(index + 1 < tempVector.size()
- 1 ? index + 1 : index,
                    index + 1 < tempVector.size() - 1 ? index + 1 : index));
        }
    }
    private void select(boolean enterPressed) {
        if (!this.isVisible()) {
            return;
        }
        if (parent == null && tableParent != null) {
            tableSelection();
            return;
        }
        if (parent instanceof JTextArea) {
            String txt = parent.getText();
            int i = 0, n = parent.getCaretPosition() - 1, count = 0;
            char c;
            for (i = n; i >= 0; i--) {
                if (!(((c = txt.charAt(i)) >= 'a' && c <= 'z') || (c >= '0' && c <= '9'))
&& count != 0) {
                    break;
                } else {
                    count++;
                }
```

```
            }
            if (count > 0 && this.isVisible()) {
                parent.setSelectionStart(i + 1);
                parent.setSelectionEnd(n + 1);
                if (i < n && !lst.isSelectionEmpty()) {
                    parent.replaceSelection((String) lst.getSelectedValue());
                }
            }
        } else {
            parent.setText((String) lst.getSelectedValue());
        }
        this.setVisible(false);
    }

    private void tableSelection() {
        int r = tableParent.getSelectedRow(), c =
tableParent.getSelectedColumn();
        try {
            tableParent.getCellEditor(r, c).stopCellEditing();
        } catch (Exception ex) {
        }
        lst.validate();
        tableParent.setValueAt(lst.getSelectedValue(), r, c);
        typed.setLength(0);
```

```
    this.setVisible(false);
    tableParent.repaint();
    typed.setLength(0);
}
public void actionPerformed(ActionEvent ae) {
    String com = ae.getActionCommand().toLowerCase();
    if (com.equals("up")) {
        moveUp();
    } else if (com.equals("down")) {
        moveDown();
    } else if (com.equals("select")) {
        select(false);
    }
}
public void focusLost(FocusEvent fe) {
    this.setVisible(false);
    typed.setLength(0);
}
public void focusGained(FocusEvent fe) {
    if (fe.getSource() == lst) {
        parent.requestFocus();
    } else if (fe.getSource() == parent
            || (fe.getSource() == tableParent
```

```
            && tableParent.getSelectedColumn() == activeColumn)) {
            populateList();
        }
    }
    public void keyPressed(KeyEvent ke) {
        int kc = ke.getKeyCode();
        int index = lst.getSelectedIndex();
        if (kc == KeyEvent.VK_F5 || (parent instanceof JTextField && kc ==
KeyEvent.VK_UP)) {
            moveUp();
        } else if (kc == KeyEvent.VK_F6 || (parent instanceof JTextField && kc
== KeyEvent.VK_DOWN)) {
            moveDown();
        }
    }
    public void keyReleased(KeyEvent ke) {
        if (tableParent != null) {
            int r = tableParent.getSelectedRow(), c =
tableParent.getSelectedColumn();
            this.setVisible(c == activeColumn);
            if (r < 0 || c < 0 || c != activeColumn) {
                return;
            }
        }
```

```
        int kc = ke.getKeyCode();
        if (kc == KeyEvent.VK_F7 || (parent instanceof JTextField && kc ==
KeyEvent.VK_ENTER)) {
            select(true);
        } else if (parent != null || tableParent.getSelectedColumn() ==
activeColumn) {
            populateList();
        } else {
            this.setVisible(false);
            typed.setLength(0);
        }
    }
    public void keyTyped(KeyEvent ke) {
        char c = ke.getKeyChar();
        if (c != '\r' && c != '\n' && c != '\t' && c != '\b') {
            typed.append(ke.getKeyChar());
        } else {
            typed.setLength(0);
        }
        if (c == '\b') {
            typed.setLength(typed.length() > 0 ? typed.length() - 1 : 0);
        }
    }
    private void populateList() {
```

```
        String txt = "";
        int r = 0, c = 0;
        if (tableParent != null) {
            r = tableParent.getSelectedRow();
            c = tableParent.getSelectedColumn();
            if (r < 0 || c < 0 || c != activeColumn) {
                return;
            }
            txt = tableParent.getValueAt(r, c) != null ? (String)
tableParent.getValueAt(r, c) : "";
            txt += typed.toString();
            if (txt != null) {
                txt = txt.toLowerCase();
            } else {
                txt = "";
            }
        } else if (parent instanceof JTextArea) {
            txt = parent.getText().toLowerCase();
            int i = 0, n = parent.getCaretPosition() - 1, count = 0;
            char ch;
            for (i = n; i >= 0; i--) {
                if (!(((ch = txt.charAt(i)) >= 'a' && ch <= 'z') || (ch >= '0' && ch <=
'9')) && count != 0) {
                    break;
```

```
        } else {
            count++;
        }
    }
    txt = txt.substring(i + 1, n + 1);
} else {
    txt = parent.getText().toLowerCase();
}
tempVector.clear();
int index = lst.getSelectedIndex();
for (java.util.Iterator<String> i = val.iterator(); i.hasNext();) {
    String s = i.next();
    if (s != null && !s.equals("") && s.toLowerCase().startsWith(txt)) {
        tempVector.add(s);
    }
}
lst.setListData(tempVector);
lst.validate();
if (tempVector.size() == 0) {
    this.setVisible(false);
    return;
}
this.pack();
```

```
        Point p = new Point(0, 0);
        if (parent != null) {
            try {
                p = parent.modelToView(parent.getCaretPosition()).getLocation();
            } catch (Exception ex) {
            }
            Point loc = parent.getLocationOnScreen();
            p = new Point(p.x + loc.x, p.y + loc.y);
        } else {
            p = tableParent.getLocationOnScreen();
        }
        if (tableParent != null) {
            Rectangle rect = tableParent.getCellRect(r, c, true);
            p.x += rect.x;
            p.y += rect.y;
        }
        Rectangle bound =
GraphicsEnvironment.getLocalGraphicsEnvironment().getMaximumWindowB
ounds();
        if (p.y - this.getHeight() > 0) {
            this.setLocation(p.x, p.y - this.getHeight());
        } else if (p.y + this.getHeight() > bound.y + bound.height) {
            this.setLocation(p.x, bound.y + bound.height - this.getHeight());
        } else {
```

```
            this.setLocation(p.x, p.y);
        }
        if (!this.isVisible()) {
            this.setVisible(true);
        }
        lst.setSelectedIndex(index >= tempVector.size() || index < 0 ? 0 : index);
    }
}
```