Word Prediction for Amharic Online Handwriting Recognition

By: Nesredien Suleiman

A Thesis

Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Addis Ababa University, Faculty of Informatics

Department of Computer Science

July, 2008

Addis Ababa University
School of Graduate Studies
Faculty of Informatics
Department of Computer Science

Word Prediction for Amharic Online Handwriting Recognition

By: Nesredien Suleiman

Approved by:

Examining Board:

1. Solomon Atnafu (Ph. D.), Advisor _____
2. _____ _____
3. _____ _____
4. _____ _____
5. _____ _____

Acknowledgement

First of all, I would like to say "***Alhamdulilahi Rabilalemin***". Second of all, I would like to sincerely thank my advisor, Dr. Solomon Atnafu, for his supervision, advice, patience, and moral support throughout this thesis work. I also want to thank Sebsibe Hailemariam, Daniel Yacob and Abera Abebaw for giving me variable ideas and guidance from the very beginning. I would like to thank Mike Scott for giving me the tool I used in this thesis.

I would also like to give special thanks to my family. My family members are always there to support me in every situation. On top all, two persons, my lovely mom and eldest brother, deserve very grateful thanks because without them I would not be who I am today.

Though it is difficult to mention the name of persons who gave me their hand while doing this thesis, it is necessary to mention those who gave their precious time to read the thesis document, to share ideas, and gave me moral and material support. Ashenafi Kassahun, Solomon Asres, Solomon Gizaw, Yonas Hailu, Daniel Kefale, Workagegnehu Petros, Misrak Tarekegn, and Yohannes are few of them. I am very grateful to thank them for what they did.

Last but not least, I would like to sincerely thank all of friends, colleagues, classmates, and my students for their assistance, encouragement, and inspiration during this research.

Table of Contents

List of Tables

List of Figures

List of Acronyms

PDA – Personal Digital Assistant

KSPC – Key Stroke Per Character

LCD – Liquid Crystal Display

CIC – Computer Intelligence Corporation

CRS – Character Recognition System

OCR – Optical Character Recognition

OHWR - Online Handwriting Recognition

ASCII – American Standard Code for Information Interchange

T9 – Text on 9 keys

DTW – Dynamic Time Warping

ACP – Adaptive Context Processing

HTK – Hidden Markov Model Tool Kit

HMM – Hidden Markov Model

TVET – Technical and Vocational Educational Training

SDK – Software Development Kit

VM – Virtual Machine

Abstract

Online handwriting recognition, keypads, soft keys are some of the data entry techniques used to enter data into mobile devices, such as smart phone, PDA etc. Data entry in these devices could be either predictive or non-predictive. A word prediction method is a data entry technique in which the first few characters of the word is written and the remaining are predicted. Among the data entry techniques, online handwriting recognition is commonly used for handheld devices such as PDAs. When online handwriting recognition is combined with word prediction, the data entry process will be more efficient.

In this work, we have proposed a word prediction model for Amharic online handwriting recognition. To design the model: a corpus of 131,399 Amharic words is prepared to extract statistical information that is used to determine the value of N for the N-gram model, where the value two (2) is considered as a result of the analyses made a combination of an Amharic dictionary (lexicon) and a list of names of persons and places with a total size of 17,137 has been used.

To show the validity of the word prediction model and the algorithm designed, a prototype is developed. Experiment is also conducted to measure the accuracy of the word prediction engine and a prediction accuracy of 81.39% is achieved.

Keywords: Amharic Online Handwriting Recognition, Amharic word prediction model, N-gram model for Amharic word prediction, word prediction corpus

CHAPTER ONE

Introduction

1. Overview

One of the most prevalent and necessary techniques used as an interface between human and machine is data entry technique. This technique is helpful to enter different kinds of data such as text, voice, image, movie, etc. to the machine in order to get processed. Since word prediction facilitates the data entry technique, the most commonly used data entry techniques will be presented in this section.

There are a number of data entry techniques that include speech, chorded keyboards, handwriting recognition, various gloved techniques [15], scanner, microphone, and digital camera.

Keyboards and pointing devices are the most commonly used devices during human-computer interaction [16]. Because of its ease of implementation, higher speed, and less error rate, keyboard dominated text entry system [17]. However, one must master the computer keyboard in order to gain the advantage of a keyboard.

In addition to keyboard problems, emerging of new application domains, availability of desktop applications, and success of text messaging on mobile devices provide an opportunity to researchers to introduce new data entry techniques which are suitable to mobile devices.

Basically, there are two text entry paradigms on mobile devices. These include key- and pen-based inputs [18].

Pen-based computers offer an interesting alternative to paper. One can write directly on a Liquid Crystal Display (LCD) screen with a stylus or pen. The screen has an invisible sensitive matrix which records the position of the pen on the surface. The trajectory of the pen appears almost immediately on the screen giving the electronic ink [20].

The development of handheld devices such as Tablet-PC and PDA made handwriting recognition a natural alternative text entry technique which not only solves the problem of large alphabet size but also helps in extending the reach of information technology to a larger community [21]. Handwriting recognition is a task that transforms handwritten input into meaningful symbolic representation. This task is more suitable for mobile or handheld devices where keyboard is not an appropriate input. Handwriting recognition is important for editing, annotating, and other applications that are heavily interactive [10, 23, 24, 25, 26].

As to the way data is presented to the recognition system, handwriting recognition system is categorized into two. These are: Online Handwriting Recognition (OHWR) and Offline Handwriting Recognition or Optical Character Recognition (OCR). OHWR uses the number and order of strokes to recognize the characters whereas Offline Handwriting Recognition uses the character image to recognize them. In the case of OHWR system electronic tablet or similar touch sensitive devices are used to collect data by tracing the pen movement and sensing the pen-tip position in order to capture x and y coordinates whereas in the case of

OCR a scanned handwritten or printed text is fed to the system in the form of digital image using optical scanner [2, 9, 11, 12, 23, 27, 28, 29].

Predictive text is one of the data entry techniques used in mobile or handheld devices. In this technique the system predicts what the user most likely intends to write based on some frequencies or other information. Prediction can be either character or word prediction. In the case of character prediction the next character is predicted based on the previous character(s) whereas in the case of word prediction the word is predicted based on the some of the characters [51].

Using predictive text input technique as a text entry method will reduce the input burden and increases the efficiency of the system [18]. The intent of predictive text is to simplify the writing of text messages, emails, and entries into an address book or calendar, makes the text input faster and easier and reduces the number of errors in applications such as handwriting recognition. Moreover, it provides suitable interfaces for the elderly and disabled persons and enables the projection of a wide range of characters (East Asian countries language characters) with a limited number of keys [8, 17, 41, 51].

Word prediction is a natural language processing problem that tries to predict the correct or most appropriate candidate word in a given context. The conventional prediction systems use word frequency lists to complete intended words which the user wants to write [38, 39]. Word completion utilities, writing aids, and language translation are among the most common applications of word prediction [39].

2. Motivation

Since most computers work in English and other few languages, people who do not speak such languages are either forced to access computers in those languages or will not use them at all. This has its own impact on the socio-economic development of that country.

In order to increase the usability of handheld devices and let people express their ideas using their native languages, these devices need to be localized into native languages. Hence, it is necessary to do research to alleviate these problems. One of the research areas which demand the researchers' attention is handwriting recognition in particular online handwriting recognition. Although character recognition is language dependent, handwriting is used everywhere without boundary limit to let people express, grow and clarify their ideas easily. Moreover, doing researches on handwriting recognition will not satisfy all people who are interested to use handheld devices because characters of some languages are represented by number of strokes. This takes much time to write a full word or sentence. Hence, in order to reduce writing time and improve correctness it is necessary to extend the options of writing text by adding additional feature such as word predicting text writing. This option is critically

important for text writing using handwriting recognition, since system that will improve the quality, speed of text writing, correct spelling errors, etc becomes an issue.

This issue becomes more critical for Amharic text entry on PDA like devices using handwriting recognition. Because the large number of characters of Amharic language that is needed to be recognized for each word entry requires high resource and accuracy. Without options such as word based predictive text entry, writing text would be more challenging.

With all the above reasons discussed we are motivated to do a research on prediction techniques so that it is possible to contribute our share to the technology advancement.

3. Research Problem

With the widespread use of handheld devices, handwriting recognition becomes an important data entry technique. Different researches have been conducted on online handwriting recognition for different language characters. Online handwriting recognition addresses the problem of interpreting the pen movement which follows a sequential pattern over time.

Online handwriting recognition is a challenging process by itself and when it comes to Amharic language it is more challenging as Amharic language characters are: represented by different number and order of strokes, formed by different structures, and composed of large number of characters which have different structures.

Unlike other languages, few works have been done to address the problem of online handwriting character recognition for Amharic language characters. However, as far as our

knowledge is concerned no work has been done to address word prediction for online handwriting recognition.

Word prediction for online handwriting recognition is challenging to develop as it is necessary to study the neighboring characters, the context or meaning of the word, and the structures and occurrences of words. However, its usage can highly improve the efficiency of online handwriting recognition.

Thus, this is an issue that needs to be addressed in order to explore the means for effective Amharic text writing particularly on handheld devices, reducing the input burden of the system, helping people with language impairment and increasing the usability of handheld devices for Amharic language users.

4. Objective

1. General Objective

The main objective of this research is to design a model, develop algorithms for a word-based predictive Amharic text input for the method of online handwriting recognition, particularly for handheld devices.

2. Specific Objectives

To achieve the above mentioned general objective the following specific tasks have been performed:

- Review relevant works in other languages
- Explore the existing word-based predictive models
- Develop a corpus upon which the models can be built.
- Review different online handwriting recognition systems, for Amharic and other languages.
- Identify and adopt one of the appropriate online handwriting recognition systems
- Design Amharic word prediction model
- Develop the necessary algorithms for word prediction engine for Amharic language
- Develop a prototype to demonstrate the effectiveness of the designed model and the developed algorithms.
- Evaluate the performance of the word prediction system.

5. Scope and Limitation

Word prediction for online handwriting recognition is quite a complex process and demands efforts and resources. The scope of this study is confined to the analysis and development of appropriate model for word prediction in an online handwriting recognition system using one of the existing Amharic dictionaries (lexicon). This work focused only on achieving word prediction accuracy of the word prediction system for online handwriting recognition. The searching efficiency of the system has not been taken into consideration.

6. Methodology

To achieve the expected result of the study, different approaches are employed as follows:

- **Literature Review**: - Related literatures from different sources (books, journals, Internet, etc.) will be reviewed to understand handwriting recognition systems in particular online handwriting recognition systems related with Ethiopic scripts. Moreover, Character-based recognitions, word-based recognitions and predictive text recognitions which have been addressed using different approaches will be reviewed. Fairly all methods will be reviewed within the frame of reference of their behavior for Ethiopic character set.

- Select and use one of the existing Amharic dictionaries with our system.

- **Adoption of Character Recognition Engine**: - To demonstrate the ability of the method to capture features of Ethiopic script a recognition engine will be investigated and/or customized.

- **Select Implementation Tools**: - In this study, we will use number of tools in order to come up with the solution for the problem we need to address.

- **Experiment**: - experiments will be performed to evaluate the performance of the developed system.

7. Thesis Organization

The rest part of the thesis is organized in 4 chapters. Chapter 2 deals with Literature Review and Related Works which are investigated on different issues, in Chapter 3, the Word Prediction Model of the system is presented, Chapter 4 deals with Implementation and Experimentation in order to show the result of the developed system and finally Chapter 5 talks about Conclusion and Recommendation.

CHAPTER TWO

Literature Review and Related Works

1 Literature Review

1 Data Entry Techniques

As it has been discussed in chapter one keyboard is one of the most commonly used data entry techniques. There are different kinds of keyboard layouts. QWERTY, named after the first six leftmost letters on the top alphabetic row of the keyboard, is the most commonly used logical keyboard layout for Latin scripts. In France, AZERTY layout is used. In this layout A and Z replace Q and W of the QWERTY layout. Similarly, in Germany, Y is replaced by Z and gives the QWERTZ layout. Additionally, Half-QWERTY keyboard is used to type with only one hand. In this keyboard, both the left and right part of the QWERTY keyboard is mapped on to the same set of keys with a special shift key to move between choices. QWERTY keyboard is the primary text entry device on desktop systems [16].

Recently, cellular phones are the most widely used mobile devices in the world [4]. In these devices text entry is done using small buttons (keys) found on the keypad. There are only 8 to 12 keys in these devices. On the 12-key keypad 10 of the keys are labeled using numbers 0 to 9 and the rest two are labeled by symbols # and *. The 26 English language alphabets are spread over the number keys 0 to 9 and space character is represented by 0 or # in most of the devices. However, some companies represent space character using different keys [4, 18, 49].

Basically, there are three text entry techniques on mobile phones. These are: multi-tap, two-key and predictive text entry methods [4, 18].

The *multi-tap* method, currently, is the most commonly used text input method for mobile phones. In this technique, a user has to press each key one or more times to enter the character. For example, to enter a character "z" a user has to press the number key 9 four times and if he/she wants to enter a character "n" the user presses the number key 6 two times [18]. This method is the easiest method to learn [16]. In this method, *timeout* and/or *timeout kill* strategies are used to differentiate among characters found on the same key, for example, to write the word "ON" one of these strategies is used as both O and N are on the number key 6. However, because of these strategies this method suffers from segmentation problem. Some mobile phones (for example, Nokia Phones) incorporate both strategies in order to let their user to choose between these strategies [18].

The ***two-key*** method is a second type of text entry method which needs two key presses to write a single character. In this technique, the first key press is to select the group of the character (for example, the number 5 key for J, K, or L) and the second one is to select the position of the character. As there are at most 4 characters on a single key, a user selects 1 through 4 to select the position of the character (for example, press 2 to select K since it is at the second position in the group). Therefore, a user should press 5 followed by 2 to enter the character K [18].

Both Multi-tap and Two-key methods require more than one keystroke per character (KSPC). This reduces the efficiency of data entry techniques. Besides for characters other than letters the number of key press will increase [4, 18].

Predictive text is the third text entry method. This method is used in both key-based and pen-based paradigms. It is dictionary based method (for example, T9 from Tegic Inc., iTap from Motorola, and eZiText from Zi Inc.) and it requires only one key press to enter each character. For example, to enter "THE", the user enters 8-4-3-0. The 0 key is for "space" that delimits words and terminates the predicting activity. In this method, the moment the key is pressed, the system compares the key sequence with the word possibilities in a linguistic database to guess the intended word. Sometimes, ambiguity arises when two or more words match the given key sequence. Under such condition, the word with the highest frequency of occurrence is chosen. What is more, a special "next" key is used to choose an alternative word if the intended word is different from the displayed one [4, 18, 52].

In recent years, dictionary-based disambiguation mechanisms have appeared in various forms. These include N-gram frequencies, syntactical information, or other statistical information of letter and word frequencies [35].

Based on previously written character(s) N-gram character prediction could be used to predict the next character (N^{th}) which is going to be written at the writing tablet. For example, due to the high frequency of words such as "the, then, them, then, these, etc." the character "e" has a

high probability to appear with the context “th” than any other character whereas character “z” has zero probability to appear with the context “th” [8].

Natural language text is highly redundant and this redundancy could be used to produce more efficient text entry by analyzing that text [36]. A large collection of documents – corpus – is analyzed to generate the statistical properties (frequency) of characters, digrams (pairs of characters), trigrams, words, or phrases in the language of interest in order to suggest or predict letters or words when part of the text is entered [18].

There are three different types of word prediction. *Word Completion* is the kind of word prediction system which is currently usable in the applications like Microsoft-Word and Microsoft-Excel. The second one is *Bigram/Trigram Prediction* which uses two/three word patterns and their corresponding frequency for the sake of prediction. And the last one is *Linguistic Word Prediction* in which the system knows the grammatical value of each word in the dictionaries (for example, Co-Writer). Unlike the previous ones, the system of this prediction technique learns new words frequently and assigns grammar to the new words automatically. With grammar-based intelligence the system can accurately predict words within the framework of valid sentence structures. In general the third type of word prediction is the most effective one [40].

Word prediction system is more effective than character prediction system in which word prediction system relies upon several words of past context whereas character prediction

relies only upon characters in the current word [37]. Having some previous context will enable the system to predict the next character based on the probability associated with it [8].

Currently there are a number of predictive systems developed by different companies for mobile or handheld devices. ***T9,*** developed by Tegic Inc. is one of these systems and it uses an internal database to determine the correct word by investigating the possible occurrences. This system also updates its dictionaries by capturing users' unique text messaging language such as slang abbreviations, symbols, mixed alphanumeric, URLs and e-mail addresses in order to increase its recognition accuracy. In addition to English, T9 supports a number of other languages such as Chinese, Japanese, Korean, Arabic, and the like. T9 is used on most of today's mobile phones [42].

eZiText, eZiTap, and Decuma, products of the Zi Corporation, are other predictive systems. ***eZiText*** is developed for one touch predictive text entry, ***eZiTap*** is developed for intelligent multi-tap entry and ***Decuma*** combines handwriting recognition software with word completion/text prediction and can be configured to predict full words in two languages at once. Like T9 system, it is also available for different languages including Arabic, Chinese, and Japanese. It also updates its prediction pool with user created shortcuts and "used words" [43, 44].

Eatoni develops three different predictive solutions for handheld devices. These include: ***LetterWise*** which is a kind of predictive system that works by pressing a key with the desired letter and keep pressing "Next" until the desired word comes, ***WordWise*** is a predictive

system which is suitable for power users who want to touch type, lastly the ***EQx*** series of efficient QWERTY style keyboards for mobile devices [45, 46].

All the above discussed predictive systems add values to the technology advancement. These systems increase the usability of handheld devices, such as mobile phones, PDAs, gaming devices and the applications in them including SMS, e-mail, Internet browsing.

Using an *electronic pen* (*stylus*) on pen-based computers introduced a new boundary for interactive systems. This input technique is suitable to perform different operations such as entering texts, selecting, and dragging [10, 19].

A *soft keyboard or virtual keyboard* is one of data entry techniques which are used to enter data into pen-based computers. In this method, a picture of a keyboard is displayed on the screen where users can touch keys with the stylus (pen) to enter the data [16]. For pen-based computers, using the soft keyboard is like using a standard keyboard with one-finger and one-hand (hunt-and-peck typing). The only difference is that in the case of soft keyboard, a user taps the surface of an LCD display/digitizer using pen (stylus). A *touch screen keyboard* is another kind of data entry technique which uses fingers to enter the data. This technique is more similar to the soft keyboard [19].

Writing systems can be broadly classified as ideographic and phonetic. In the case of phonetic systems, symbols are used to represent the alphabets or letters of the language. On the other hand, ideographic writing systems represent the entire word, syllable or idea using single symbol. Chinese characters or Egyptian hieroglyphics are the type of ideographic writing systems whereas Latin and Ethiopic characters are classified as phonetic writing systems [16].

Handwriting and speech are universal human communication methods and they are natural alternatives to typing. As compared to keyboard, these two interfaces are easier to learn by new users. Handwriting interface provides higher degree of privacy and confidentiality over speech. However, it requires users to be literate in order to use it [2]. Additionally, speech recognition systems suffer from noisy environments, require considerable training, and they are not efficient for text editing and manipulation [15].

PDA (personal digital assistant) is a small portable device that, at the very least, should be able to store phone numbers, tasks, and appointments [53]. PDAs use on-screen letter recognition system. Graffiti and Jot are some of these letter recognition systems [4].

Graffiti, the handwriting recognition system used by handhelds using the Palm operating system, requires the user to write using specified scripts designed for this system. The script is easy to learn. Thus, it is easy to use. ***Jot*** is also a kind of handwriting-recognition program from Computer Intelligence Corporation (CIC). Unlike Graffiti, Jot represents characters using several scripts. For instance, it provides different ways to write the character "e". Windows CE devices typically use Jot as their handwriting recognition software, although versions for the Palm OS (Palm Operating Systems) are available [53].

Handwritten text can be entered into the computer using different kinds of on-line and off-line input devices. Online data is temporal order of the points which is captured using stylus based devices such as tablet-displays and digitizing tablets. Devices like scanners of flat-bed and paper-fed are used to enter an off-line data which return an image as a bit-map of pixels [22].

Character recognition is a process done by a machine to produce a meaningful output from a text-based input pattern entered using either on-line or off-line devices. The system that handles this operation is called Character Recognition System (CRS). The possible output of the system includes: sequences of symbols (for example, 'Y E S'), the date on a Cheque (for example, 'Feb. 14, '94'), and the signature. Character recognitions are associated with images or other representations of handwritten characters, words, or sentences entered using data entry techniques [22].

Offline handwriting recognition approach does not require an immediate interaction with the user rather it works on the images; days, months, or years later. Off-line data input, for example, is used in applications for storing information that has already been acquired through forms [2, 11, 12, 23, 26, 32, 28].

As stated in [26], the main disadvantage of online handwriting recognition is that it requires special equipment to use the system. The advantages of OHWR over off-line handwriting recognition are:

- It is interactive. In an editing application, for example, writing of symbols can cause the display to change appropriately and recognition errors can also be corrected immediately.

- It has an advantage of adaptation. When the user sees that some of his/her characters are not being accurately recognized, he/she can alter the drawing to improve recognition.

- It has better recognition accuracy. The temporal or dynamic information that is captured by online devices consists of the number of strokes, the order of the strokes, the direction of each stroke, and the speed of the writing within each stroke. A stroke is the writing from pen-down to pen-up.

The main drawback of off-line handwriting recognition is that scanning or photographing causes "noise" (such as lines or patterns on the paper, extra marks from dust or scratches or a printing process), which distracts the main image [28].

Furthermore, handwriting recognition system may be classified as writer-independent and writer-dependent systems. Writer-independent means that the system can handle the variations in individual writing styles, whereas a writer-dependent system is trained and optimized to recognize an individual's writing [7].

A word is the most natural unit of handwriting. It is constructed by combining alphabets found in the symbol set of the language. Words can be written in isolated, cursive or mixed form. ***Isolated*** (***discrete***) form is a way of writing a word by separating characters using spaces whereas writing a word by connecting characters to one another is called ***cursive*** writing style. When both styles are used together, it is called ***mixed*** form. For example, words written in Latin alphabets can be written in both isolated and cursive form, words written in Arabic language are in cursive form and Amharic words are written in isolated form. This variation of writing styles makes handwritten word recognition process difficult [9, 30].

Word recognition is a process of getting a handwritten word image and/or a lexicon as an input and determines the most appropriate word that best matches the specified parameter.

The matching process can be improved by using the contextual knowledge of characters of the word in the lexicon. For example, it is well known that the identity of a character is constrained by the identity of its predecessor in the word [31, 32].

Handwritten word recognition methods could be lexicon-driven or lexicon-free method. In the case of lexicon-driven method, the recognition is based on the given handwritten word image with a list of possible target words – the lexicon. This method is an appropriate method for applications such as recognition of street, city and state words in postal processing, and for bank check processing. On the other hand, lexicon-free handwritten word recognition method works without any lexicon and it requires a very strong linguistic post-processing. It is useful for applications such as automatic data entry from scanned images [33].

During handwritten word recognition process, words are treated in one of the two approaches: Holistic or Analytical. Holistic approach treats words as a whole without any segmentation process whereas Analytical approach recognizes the word as a sequence of smaller size units which must be easily related to character [13, 30, 34].

Since the holistic approach is related to a fixed lexicon of word description, modifying the lexicon requires human training. On the other hand, the analytical approach is related with dynamically defined vocabulary and hence doesn't require human training when its vocabulary is required to be modified. Moreover, in the analytical approach the lexical knowledge, which can either be described by means of a lexicon of ASCII words or by statistical information on the letter co-occurrence (n-grams, traditional probabilities, etc.), is used for the purpose of recognition. The above described characteristic of the two approaches

states the drawback of the holistic approach and the advantage of the analytical approach. The advantage of holistic approach over analytical is that no word segmentation is required [34].

2 Related Works

1 Character Recognition

A lot of researches have been done on character recognition for other languages and few for the Amharic language. Character recognition as it has been discussed could be either on-line or off-line. In this section, though one work on offline character recognition for Amharic language and one work on online character recognition for other language are presented, mainly the works done on on-line character recognition for the Amharic language are presented.

Worku, [47], tried to use the offline printed technique to the Amharic scripts. According to him the Amharic language scripts are formed by taking 24 characters from Sabaean, 2 characters from Ge'ez and by adding 8 additional characters to get sounds which were not there in either of the Sabaean or Ge'ez.

In his work, Worku has computed the frequency of occurrences of Amharic language scripts by taking a sample of 5000 words. To generate the frequency, percentage, cumulative frequency, and cumulative percentage he has used SAS program and in addition to that manual operation has been done as there are duplicate characters which have the same ASCII code. According to his work the character '□' is the most frequently used character which

occurs 946 times out of 20,259 characters this is accounted for about 4.6695% of the total occurrence of characters.

Moreover, he has put statistical information that shows which order characters are most frequently used characters. The result shows the 6th order characters are the most frequently occurring characters, next to this the 1st order characters occurred more, the 4th order is greater than the 2nd and 3rd order, the 3rd order is greater than the 7th order and the 5th order is the least occurring character.

In addition to this, he has done the same statistical computation on the characters which have the same sound but different shapes. Table 2.1 below shows the frequency and percentage of these characters.

Table 2.1: Statistical Information of Homophone Characters

Character	Frequency	Percentage
□	461	2.28
□	94	0.46
□	52	0.26
□	914	4.51
□	113	0.56
□	947	4.67
□	146	0.72
□	44	0.22
□	31	0.15

Even if the above discussed work is on off-line handwriting recognition, the work presented how statistical information is obtained such as the frequency of occurrences of characters and orders of characters. Thus, we have learnt how to get such information from this work.

Abinet, [1], has proposed a strategy to alleviate the problem of on-line handwriting recognition for Ethiopic characters but only for basic characters. She mainly considered the structural relationships between strokes, formation of an efficient structure to store the available strokes, and their order to form characters in order to avoid the inefficiency of both memory and speed.

She, has designed an algorithm for various preprocessing activities such as extra pen-up noise elimination, size normalization, filtering, and resampling, for feature extraction and for classification which contains three different layers such as coarse classification, detailed classification, and superimposing classification. She also has implemented the above algorithms. In this research she has used Structural approach.

As it has been mentioned by the researcher, recognizing a character based on stroke number and stroke order dependent approach, if the system is writer-independent it would widen the

search space and eventually reduce the efficiency of the system, however, the system she has applied in this research was writer-dependent system therefore, the problem mentioned above was minimal. Using the techniques and methods discussed she achieved an accuracy of 99.4%.

Daniel is another researcher who has done on-line handwriting recognition for Amharic language characters. He, [3], has proposed a writer-independent on-line handwriting recognition system. To accomplish his work he has implemented two preprocessing tasks namely extra pen-up point elimination and size normalization. And he also applied a classification algorithm that handles stroke number variation and the difference in number of sampled points exhibited due to writer speed and shape variation.

The researcher argues that depending on the stroke number to recognize a character is inappropriate approach to recognize a character because if in case the number of strokes used during training is different from number of strokes during recognition the character won't be recognized correctly. Therefore, he proposed a DTW matching algorithm so that it is possible to include the non-basic characters, Ethiopic numerals, labialization characters, and additional characters from Tigrigna and other languages as DTW is not a character set dependent system. However, he has done a test only on the first order Ethiopic characters. Moreover, the researcher focused on the first shortest distance reported by the recognizer. He achieved 71.9% overall accuracy average for the shortest distance recognition.

Yonas, [48], has proposed a constrained and writer-independent recognition system with simplified character sets which reduce the complexity of the character set and introduced a standard way of writing which in turn increases the accuracy of recognition engine. As it has

been stated in the paper to determine the usability and acceptance of recognition systems he has used parameters such as recognition accuracy, ease of use, and understandability of the recognition system and writing and recognition speed. He also designed simplified scripts to represent Ethiopic character with specific number and order of strokes and specific direction for writing the strokes. A single symbol is assigned to monophonic characters and he designed a different text entry environment for numeric and alphabetic characters as well.

Preprocessing, feature extraction and classification modules which communicate to each other through flat files are used. The researcher also designed an algorithm that segments the input character pattern into its subsequent sub-stroke or directed segments and represents each sub-stroke with a single code and corresponding magnitude information.

On the first trial the researcher, achieved an average accuracy rate of 88.64% and on the second trial he achieved an average accuracy rate of 93.17%. In general, he achieved an overall average accuracy rate of 90.19%. And the character group evaluation result was about 97.01% for Ge'ez while other group characters recognized with unevenly distributed error rates from the first to the next trial.

All the above discussed works are done on online handwriting character recognition. Researchers used, in their work, preprocessing, feature extraction and classification processes as these processes are required for character recognition. Even if the above papers are on handwriting recognition, they only focused on characters. Moreover, except [48] the others are done particularly on basic characters and in [48] even if non-basic characters are included

some most frequently characters are ignored and the system is also more of constrained. Since characters are the basis of any language, the above works provide useful information that will help contributes to our work as recognizing characters will be the first step of word prediction.

The next reviewed work is done on online handwriting character recognition for Japanese characters. Japanese language has a large number of characters [9]. The researchers have tried to describe the Fujitsu's latest interactive Online Character Recognition (OLCR) technology for pen-based computers.

The number of characters to be recognized, the similarity among characters, and the various types of characters combined in a sentence and unused spaces as a separator between words are the problems that the researchers tried to alleviate by developing a high-performance hybrid handwriting character recognizer, which integrates online and offline recognition module. This system recognizes more than 4400 character of the writing order, stroke concatenation and deformation. In addition to the above system they have also developed a high-performance context processing module which can discriminate between similar characters using a very small dictionary.

Different experiments have been done by Kazushi *et. al.* in [9]. The recognition accuracy achieved by hybrid recognizer was 86.8% whereas the recognition accuracy achieved by online and offline modules were 84.3% and 72.4% respectively.

Moreover, the experiment made on the context processing module showed an improvement on the recognition accuracy from 82.7% to 90.5% for non-kanji characters and from 90.6% to

93.8% for Kanji characters. Finally the researchers developed an adaptive context processing (ACP) technology which improved the recognition accuracy for specific user by automatically learning a string that has been entered by the user. The experiment made on this aspect showed that the recognition accuracy improved from 86.1% to 95.4%.

2 Word Recognition

Since no work has been done on online handwriting word recognition for Amharic language, in this section, the work done on online handwriting word recognition for other languages are discussed.

In [6], the researchers proposed an online handwriting recognition system for Turkish language. HTK software has been used for training and recognition purpose. To perform an experiment, two different models such as letter and word model have been designed. In the former model, a word in the lexicon is labeled according to the constituent letter models and forward-backward algorithm has been used to train the letter models and whereas in the case of word model, the model was trained for each word in the lexicon using different word samples. The word model can also be created from consecutively lining up letter models.

Esra *et. al.* used two stages to develop and test the system. In the first stage, a database of 1000 words collected from 20 people evenly has been used. In the second stage, a database of 3000 words has been used. The words of this database were collected from 30 people who have written 100 words each. Initially words have been collected from e-mail messages. Using these two databases the researchers made different experiments and achieved 97% and 95% performance accuracy for letter and word model respectively. They have recommended that to achieve a better performance there should be enough training data and because of the

dots and cedillas found in Turkish alphabets the researchers concluded that stroke ordering is not suitable for online word recognition.

In [2] researchers have proposed an online handwriting recognition system for Arabic language. These researchers developed an algorithm for delayed-stroke projection which handles delayed strokes as Arabic language has dots and additional strokes with the alphabets. The delayed algorithm has involved two steps such as identifying of delayed-stroke and incorporating the stroke in the body of the word part. In addition to this, a discrete HMM (Hidden Markov Model) for the purpose of recognition task has been used.

During the experiment, the training data set was used by collecting words from four persons who have written 800 words each. Similarly, the test data set has been collected from ten persons who have written 280 words each. Using these words the experiments have been conducted on five different dictionary sizes such as 5000 (5K), 10000 (10K), 20000 (20K), 30000 (30K) and 40000 (40K) words. Table 2.2 and Table 2.3 show the recognition rates of words and parts of the words for both writer-dependent (WD) and writer-independent (WI) systems [2].

Table 2.2: The average recognition rates for words

	5K	10K	20K	30K	40K
WD	96.47	95.50	92.86	90.84	89.75
WI	96.28	95.21	92.55	89.68	88.01

Table 2.3: The average recognition rates for word parts

	5K	10K	20K	30K	40K
WD	98.44	97.94	96.86	95.90	95.44
WI	98.49	97.78	96.54	95.12	94.40

What we have learnt from the above discussed papers is that in the case of word recognition there should be a word database in order to recognize a word. The recognition accuracy increases when large number of words are used for training data set. Moreover, the increase in dictionary size will decrease the recognition accuracy this is because for large dictionary size the search space will increase as a result it takes much time.

3

Text Prediction

Predictive Text Entry Speed on Mobile Phones

In [49], the researchers tried to address the problem of text entry on mobile phones. To address the problem they have designed the method for predicting potential expert user text entry speed for input methods that utilize the 12-key keypad and the model provides individual predictions for one-handed thumb and two-handed index finger use. They have discussed the three most common text entry approaches through out the paper such as Multi-press, two-key press and T9 methods.

Silfverberg *et. al.* made a formal analysis by taking 9025 most common words in English produced from the British National corpus and they found that the user must press the NEXT key only after about 3% of the words.

The designed model has two components such as a movement model (Fitts' law) and a linguistic model (digraph probabilities). Fitts' law is a quantitative model for rapid, aimed movement and it is used to calculate the potential text entry speed of an expert user. This model has been applied with success to pointing devices and on-screen keyboards. In the case of the linguistic model (digraph probabilities) based on the analysis made on representative sample of common English probability P_{ij} is given for each letter pair.

The researchers have made experiments and two types of tasks were given to the subjects during the first experiment. In the first task participants pressed only a single key whereas on the second case participants pressed two keys alternately for ten seconds. The average

movement time (in ms) between successive key presses was then calculated using the formula shown in Eq. 1 :

Eq. 1

Where MT is the Movement Time and N is the number of keys pressed during a 10 second interval.

According to the experiment result, the average MTs are 273ms and 309ms for index finger and the thumb respectively and hence this shows index finger was faster than the thumb. Table 2.4 below shows the result of the experiment made on the three text entry approaches [49].

Table 2.4: Results of model predictions (WPM)

Method	Index Finger	Thumb
Multi-Press		
– Wait for timeout	22.5	20.8
– Timeout kill	27.2	24.5
Two-Key press	25.0	22.2
T9	45.7	40.6

In general the “Timeout Kill” strategy is faster than “Timeout” strategy by 21% when using the index finger and 18% when using the thumb. When Multi-press uses timeout strategy it is slower than two-key press method. However, when it uses the timeout kill strategy it is faster than the two-key press method.

Te dictionary or linguistic model embedded in the T9 method is used to determine how often the NEXT function is required. The analysis shows out of 9025 words in the sample, 8437 (95%) can be entered and uniquely disambiguated. The number of words required 1, 2, 3, 4, or 5 presses of the NEXT function are 476, 83, 23, 5, and 1 respectively. The initial word for any key sequence is the one with the highest probability in the linguistic model, while subsequent words are produced in decreasing order of their probability.

From the above discussed paper, we have observed predictive text entry method has better performance than other text entry methods and when there is a convenient text entry method the data entry performance will be better. Moreover, linguistic model which has a probability or frequency of occurrence of each word is the core element of prediction system.

Leveraging Word Prediction to Improve Character Prediction in a Scanning Configuration

Lesher *et. al.* in [37] mentioned that character prediction depends only on characters in the current word whereas word prediction can be designed to take advantage of several words of past context and also stated that word prediction method should generate character distribution by tabulating the probabilities of each character across each predicted word. For example, if after typing "th" the predicted words and associated probabilities were (the = 0.5, there = 0.2, this = 0.2, though = 0.1) the predicted characters could be (e = 0.7 = 0.5 + 0.2, i = 0.2, o = 0.1). This example shows that the character "e" has a higher probability than others.

Lesher *et. al* made an extensive research on advanced n-gram word prediction (the past n-1 characters are used to predict the current n^{th} character) and achieved prediction model which

provide keystroke savings of nearly 60% in a direct selection paradigm with a six word prediction list. Furthermore, since the character probabilities skewed by insufficiently large sample set when short prediction list is used and the accuracy of the character prediction is determined by the length of the word prediction list they have recommended using longer prediction list. Therefore, they have stated that 50 words list, would be sufficient to ensure the accuracy of word-based character prediction.

They have used scanning model with supplemental character list in order to scan a prediction list of five characters in a linear fashion before a static character matrix scanning takes place in a row-column fashion.

In this work, seven representative testing texts with lengths five and ten thousand words were reproduced by using an IMPACT evaluation platform. In addition, this platform keeps track of the required number of switch activation for each representative testing text. The result showed that a substantial performance improvement has been achieved from the word-based character prediction. They have achieved a consistent result through out the seven testing texts and the switch savings increase by about 8% points from a baseline of 22.6% to 30.4%. They have tried to identify that the increasing of the relative weight of the word-based character prediction engine will provide substantial gains. Hence when the weight of the word-based component reached approximately 50% the performance stays leveled and when the weight is above 90% it falls down.

Effects of N-gram order and Training Text size on Word Prediction

In [50], the researchers believed that having more accurate predictions will provide a number of advantages like improving the quality as well as the quantity of message production for young people, persons with language impairments, and for those who have learning disabilities and disambiguate sequences from ambiguous keypads and correct spelling errors.

Lesher *et. al.* in addition to the establishment of a set of performance measures for word prediction using n-grams of higher orders (bigrams and trigrams) have also established 21 experimental conditions by combining three different n-gram orders (unigram, bigram, and trigram) with seven representative testing texts of at least 2500 words for each. The content of the testing texts was independent from that of the training texts. To accomplish the above stated tasks they have used a corpus which is evenly collected from Brown Corpus, the LOB (Lancaster-Oslo/Bergen**)** corpus and the collection of Time Megazine articles which has a size ranging from 100 thousand to 3 million words.

The researchers stated that irrespective of the n-gram order the performance of the system increases with an increase of training text size. However, the increase is much more on sounds for trigram (7.5 percentage units) than for unigrams (4.5 percentage units). Moreover, with higher n-gram orders the keystrokes savings for the given training texts increases constantly. A large jump in keystroke savings was realized when moving from unigram to bigram word prediction (6.4 percentage points at 3 million words) reflecting the transformation from context-insensitivity to context-sensitivity. The performance gain in moving from bigram to trigram prediction is considerably less dramatic (0.8 percentage points), although the difference grows for larger training texts.

Spot tests with higher order n-grams revealed an even smaller performance difference between trigram and quadgram (n = 4) word prediction, although with larger training texts this difference may also increase.

This paper indicated that a corpus is the necessary element to work on a prediction system. Having larger corpus presents information about how frequently words are used in a document

Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment

In [5], researchers proposed keyboard mapping scheme called ETWirelessKeyBoard to perform keyboard mapping and also proposed new algorithm called EasyET for predictive text inputting in order to achieve practical and efficient composing of message texts with Ethiopic scripts on a wireless device in particular on a wireless phone. They have also tried to eliminate four sets of characters in order to reduce the number of characters in the font set. These reduced sets are 'Superfluous' characters, replaceable characters, Ethiopic numerals and labialized characters. By reducing these sets they have reduced the number of characters used to write the messages by 40%. They have studied two entry methods such as Multi-press and Three-key input methods as well. In order to determine the frequency of words in the dictionary they have used three rating attributes. Moreover, they have developed a wireless application for Ethiopic text messaging.

Character Prediction for Online Handwriting Recognition

Character prediction is used to reduce the number of errors in applications such as handwriting recognition. In [8], researchers have studied the character prediction and its potentials for increasing recognition accuracy and also provided a character predictor based on a character-level n-gram with an optimal length of context for application to handwriting recognition.

The next character with some probability associated with it could be predicted by having the previous context and hence character n-grams could be used in this aspect. For example, by having the context “th” it is very probable that an “e” will follow, due to the high frequency of words such as “the, then, them, these, etc”. However, the probability of “z” following “th” is zero, in the English language.

Researchers generated a database that contains the list of words and their respective frequency in the language. They have used a large corpus which is about 320,000,000 words in the form of online office correspondence to determine how long should be the list of words and to identify the size of N. Out of the above corpus about 273,000 distinct words were extracted and then 40,000 most frequently used words were extracted. Using the 273,000 and 40,000 words researchers tried to decide the size of N.

Beigi *et. al.* found N = 4 is an optimal choice after checking the average fan-out perplexity for large and small vocabulary for different Ns and the cost of looking up an n-gram in the bigger database. The generated 4-grams and their corresponding probabilities were stored in a TRIE-like database. This database contains the probabilities associated with each letter in the

tetra-gram based on its previous letters which means that there are four probabilities stored for each tetra-gram. After an experiment has done an accuracy of 92% was achieved.

In this paper we have learnt that in order to start prediction, the n-gram should be decided and we have also learnt the way to decide the n-gram. The necessity of large corpus is also presented for prediction process.

3 Summary

In this chapter, different works which are done on Character Recognition, Word Recognition, Character and Word Predictions for different languages have been discussed.

The works done on Amharic languages were reviewed to see the difference and similarity between character recognition of Amharic language and other languages, to understand the characteristics of Amharic language characters or alphabets, and to check whether there are papers related directly to our work so that it is possible to apply or integrate the necessary components of the work.

According to the findings, only one paper focused on predictive text for Amharic language. However, this paper was not done on online handwriting word prediction. Therefore, it is necessary to do a research on online handwriting word prediction to fill the gap seen.

All the above reviewed papers provided information that is used during development of word prediction model, designing algorithms which later on be implemented.

CHAPTER THREE

Word Prediction Model for Amharic Online Handwriting System

In chapter 2, we have presented the relevant researches conducted on character recognition, word recognition, and predictive text for different languages. Though few researches have been done on online handwriting character recognition for Ethiopic characters, in almost all of the researches, the nature of Ethiopic characters has been clearly discussed. Thus, this has given us a clear idea on how to model the Amharic word prediction system and design appropriate algorithms for Amharic word prediction system.

In this chapter, the developed word prediction model, the architecture of the word prediction engine and the developed algorithms will be presented.

1. The Word Prediction Model

To accomplish a task of word prediction, one has to have statistical information such as the frequency of occurrence of words. This can be achieved by using a corpus. Since the Amharic word corpuses are not available easily, we prepared an Amharic word corpus in order to design the word prediction model. Since having a good collection of words in the corpus helps to design a better model, we prepare the corpus from various sources that include private and government newspapers, books which are written by different authors on different

issues such as politics, religion, history, and love. The list of the sources used to prepare the corpus is shown in Appendix A.

The corpus is used to generate statistical information such as the frequency of occurrence of each word, the average word-length of Amharic language as there is no clearly specified average word-length like English language. Moreover, the corpus is used to decide the N in the N-gram model, where N is the number of characters after which the system starts to predict the intended word. Both the statistical information and the value of N are used to design the word prediction system and the algorithm that will be implemented.

Worku, in [47], has computed using SAS program the frequency of occurrence of characters by taking 5000 sample Amharic words. He has also shown which character among the homophone characters is the most frequently used. We have used WordSmith to extract the statistical information mentioned earlier. About 1,614 different files, collected from various sources mentioned above, are provided to the tool and a total of 641,333 words are generated out of which 131, 399 of them are distinct words. All the 1,614 files are converted to UTF16 format in order to be used by the WordSmith tool. The word corpus which we refer it as: ***"The Word Prediction Corpus of Nesredien"*** that is used in this work is, thus, composed of 131, 399 distinct words.

Since newspaper is one of the resources used to extract the corpus, there might be typographical errors like ignoring space between words, using character other than the intended one and forgetting to use the proper fonts and so on. However, all these errors do not

have much impact on the prediction process as their proportion to the total correctly written words is minimal or negligible.

Table 3.1 lists thirty most frequently used distinct words, their frequencies, the percentage of occurrence of that word compared to the total size of the corpus, the number of files in which the word is found, and the percentage of number of files compared to the total number of files.

Furthermore, the tool, WordSmith, is used to find the average word length and get information about which word length is the most frequently used one. Table 3.2 shows the sample data of ten randomly selected files out of 1,614 files used for generating the corpus and Table 3.3 shows the sample data taken from five different Amharic dictionaries written by different authors, one list of words which contains personal and place names, and one dictionary which is made from two different dictionaries. According to the statistics found, even though there are words which have more than 8-letters length, they are not included in both of these tables, Table 3.2 and Table 3.3, as their number is much insignificant. In addition to this, some of the words which have more than eight characters get that length because of typographical errors.

Table 3.1: List of Top 30 Distinct Words Extracted using the WordSmith Tool

No	Words	Freq. of occurrence of words	% of Words	Files out of 1614 files	% of files
1	ነው	8,523	1.09	1,174	72.74
2	ላይ	6,134	0.79	1,289	79.86
3	ቀን	3,291	0.42	1,598	99.01
4	ወደ	3,037	0.39	762	47.21
5	ጋር	2,875	0.37	874	54.15
6	አቶ	2,747	0.35	646	40.02
7	ቤት	2,700	0.35	744	46.10
8	ውስጥ	2,616	0.33	853	52.85
9	ነበር	2,559	0.33	340	21.07
10	ጊዜ	2,549	0.33	574	35.56
11	ነገር	2,282	0.29	363	22.49
12	ግን	2,260	0.29	556	34.45
13	እና	2,252	0.29	539	33.40
14	ሰው	2,081	0.27	220	13.63
15	አንድ	1,982	0.25	497	30.79
16	እንደ	1,817	0.23	262	16.23
17	ሁሉ	1,783	0.23	207	12.83
18	ሲሆን	1,626	0.21	836	51.80
19	በኋላ	1,613	0.21	702	43.49
20	ብቻ	1,598	0.20	442	27.39
21	ብለዋል	1,561	0.20	710	43.99
22	መሆኑን	1,533	0.20	737	45.66
23	የኢትዮጵያ	1,521	0.19	657	40.71
24	ምን	1,388	0.18	217	13.44
25	ሰዎች	1,314	0.17	378	23.42
26	አለ	1,308	0.17	142	8.80
27	ይህ	1,265	0.16	409	25.34
28	መንግሥት	1,216	0.16	457	28.31
29	ወይም	1,192	0.15	235	14.56
30	ቤቱ	1,176	0.15	322	19.95

Table 3.2: Sample Data on 10 Different Files to Show Word Lengths

Data Item	File1	File2	File3	File4	File5	File6	File7	File8	File9	File10
Words in text	242	184	295	348	419	137	2018	271	159	486
Distinct Words	188	131	192	228	267	112	1131	213	117	363
Mean Word Length	4	4	4	4	4	5	4	4	4	5
1-letter	2	6	0	6	4	4	20	6	5	1
2-letter	19	20	25	30	78	10	282	25	22	34
3-letter	42	56	106	87	99	16	558	59	33	86
4-letter	70	47	58	112	80	22	459	69	52	136
5-letter	48	31	44	44	55	30	303	52	16	98
6-letter	42	10	25	37	43	26	242	34	18	68
7-letter	10	4	16	12	25	17	107	15	4	37
8-letter	5	4	12	5	10	6	23	4	0	20

The dictionaries listed in Table 3.3 are described as follows [54].

1. አማርኛ - እንግሊዝኛ መዝገበ ቃላት (Amharic-English Dictionary) - by Dr Amsalu Aklilu, Kuraz Publishers, Addis Ababa, 1987. It contains about 16,231 words in three columns. In this dictionary, he applies the classic spellings of words.

2. **የኢትዮጵያ ደራሲያን ስም ዝርዝር** (List of Ethiopian Authors) - by Solomon Gebre Christos, Addis Ababa, Haile Selassie I University Library, 1971. It contains about 906 Personal and place names and their transliterations.

3. **አዲስ የአማርኛ መዝገበ ቃላት** (New Amharic Dictionary) - by Desta Teklewold Artistic Printers, Addis Ababa, 1970. It contains 51, 519 words.

4. **የዐማርኛ፡መዝገበ፡ቃላት** (Amharic Dictionary) - by Kasatie Birhan Tessema, Artistic Printers, Addis Ababa, 1959. It contains 44,194 words in two columns.

5. The Transliteration of some Amharic Names of Person, Places and Things, Part One – by Berhan Ayalew, Institute of Ethiopian Studies, Addis Ababa University, Addis Ababa, January 1980. It contains about 6,648 collections of personal and place names and their transliterations.

6. Amharic Dictionary – by Fulass Armbruster. It contains about 13,146 words.

Table 3.3: Description of Number of Letters per Word from Different Dictionaries

	Amharic-English Dictionary	List of Ethiopian Authors	New Amharic Dictionary	Amharic Dictionary	Transliteration of Some Amharic Names	Amharic Dictionary	Combined Dictionary
Author	Amsalu Aklilu	Solomon G/Christos	Desta T/Wolde	Kisate Birhan Tessema	Birhan Ayalew	Fulass Armbruster	
Distinct words	16, 231	906	51,519	44,194	6,648	13,146	17, 137
Mean Word length	4	4	4	4	4	4	4
1-letter	62	7	811	4,404	0	99	6
2-letter	2,790	87	16,183	8,851	766	773	1,296
3-letter	6,313	357	27,544	23,597	2,675	3,672	4,913
4-letter	5,543	283	24,457	20,362	2,047	5,088	4,758
5-letter	902	107	12,788	9,201	777	2,678	2,060
6-letter	915	42	5,820	4,225	298	831	814
7-letter	55	30	471	1,010	72	93	72
8-letter	15	4	26	149	10	0	11

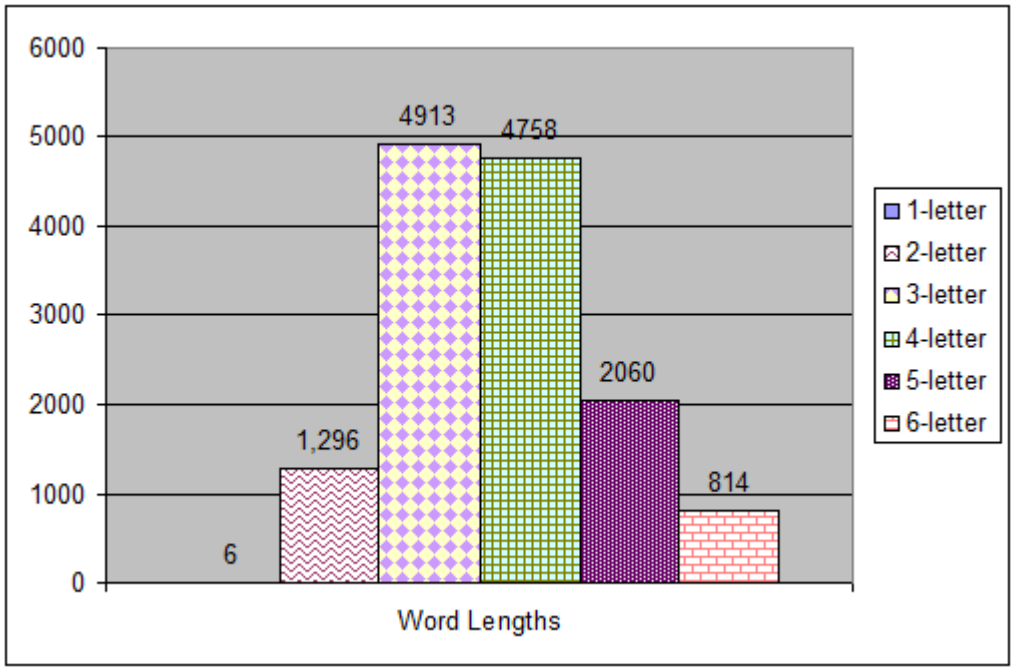

Figure 3.1: Word-Length Distribution in the Reference Dictionary

As it has been discussed so far, the objective of this research is to predict words what users intend to write based on the first few characters. Figure 3.1 shows the word length distribution of the reference dictionary. The above two tables, Table 3.2 and Table 3.3 and the chart shown in Figure 3.1 have given information about which word length is the most frequently used one and what is the average word-length of Amharic language. This information is used to determine the value of N for N-gram model, where N is the number of characters after which the system will start prediction, for Amharic word prediction system. What we have observed from the above two tables and the chart is the 3-letters word is the most commonly used word-length in Amharic language and the average word-length of Amharic language is four letters long.

In order to know how wide the search space will be during searching words in the dictionary (Lexicon) using the first two characters, an algorithm, shown in Figure 3.2, is designed that

can be used to extract and count the number of words which have two similar first characters from the corpus and later on select five top most used number of words that begins with pair of characters in each character family of Amharic language. Appendix B shows the five top most number of words begin with two characters that are used to analyze the search space. From this data, what we have observed is that even though there are large numbers of words obtained using some pair of characters, in most cases the numbers are not much big to search. Moreover, when it comes to the dictionary, which consists of words about 17, 137, those large numbers, shown in Appendix B, will not be large enough as the corpus. Of course, the number of words will decrease if more than two similar first characters are used.

According to the analysis done on the pair of characters listed in the Appendix B, it would be better if more than two characters are used to start prediction, however, the result of the tables, Table 3.2, Table 3.3 and the chart shown in Figure 3.1, indicates the most frequently used word-length is 3-letters. This shows starting prediction using more than two characters will let the word prediction system to perform normal operation (no prediction).

The analyses made to identify the most frequently used Amharic word-length, the average Amharic word-length, and the width of the search space provide a clear idea on after how many characters the system must start prediction. Thus, based on these arguments, it is decided that the value of N for N-gram model to be two (N=2), where N is the number of characters after which the prediction starts. Thus, we have 2-gram model for our Amharic word prediction system.

```
Input: Corpus Table
Output: The list of top 5 most number of words begins with pair of characters of each
character family.
Count <– 0      Count1 <– 0    Str <– null       wordCount <– 1
Letter [ ] //An array of all Amharic basic and Non-basic characters
Repeat
        Repeat
                Str <– Letter [Count] + Letter [Count1]
                Do
                        Search a record from Corpus table that begins with Str
                        If (found)
                                wordCount <– wordCount + 1
                                write the word in the file
                While (EndofFile)
                Write str and wordCount to the file wordCountFile
                Count1 <– Count1 + 1
                wordCount <– 1
        Until (Count1 >= length of Letter array)
        Count <– Count + 1
Until (Count >= length of Letter array)
letterCount <– 0
Read wordCountFile
Do
        Repeat
                Compare the wordCount of each word of each character family
                Store the top five word counts
                letterCount <– letterCount + 1
        Until (letterCount >= 34)
```

Figure 3.2: Algorithm to extract and count words from the lexicon and select the five top most number of words

Shiferaw et. al. in [5] discussed two factors that have impacts on the prediction of Amharic words. The first one is the structure of Amharic language and the statistical distribution of the order of characters, and the second one is the habits of users to write words, phrases and sentences. And also in his work, it was mentioned that to predict a word or phrase 'apriori' is difficult unless a dictionary is used in order to build a library of words, phrases and sentences commonly composed by the user.

Considering the above mentioned factors and the difficulties of acquiring words commonly composed by a user, an appropriate dictionary (lexicon) as it has been discussed in the previous section is selected.

Since one of the inputs required for word prediction engine is a lexicon, we have chosen a commonly used Amharic dictionary. Even though there are a number of Amharic dictionaries written by different scholars, we have used a combination of the ***"Amharic-English Dictionary (አማርኛ - እንግሊዝኛ መዝገበ ቃላት)"*** of Amsalu Aklilu and the ***"List of Ethiopian Authors (የኢትዮጵያ ደራሲያን ሰም ዝርዝር)"*** by Solomon Gebre Christos which contains personal and place names for this work.

The two dictionaries are combined to form one dictionary as there are missing personal and place names in the Amsalu Aklilu's dictionary which could be found in the Solomon Gebre Christos's list of authors. Similarly, there are number of words which are found in the Aklilu Amsalu's dictionary that are not found in Solomon Gebre Chiristos' list of authors. The combined dictionary consists of about 17,137 words. Hence, the prediction algorithm is designed based on this dictionary.

As it has been discussed earlier in this chapter, the prepared corpus contains list of words and their corresponding frequencies of occurrences in the source documents. This corpus is useful in order to map the frequency value of each word to the corresponding word in the dictionary as the frequency of these words is required in order to set the ranking policy of the prediction engine. We have designed the Algorithm, shown in Figure 3.3, to map the frequency values of words in the corpus to the corresponding words in the dictionary.

In addition to the dictionary, the word prediction system takes recognized handwritten characters from an online Amharic handwriting character recognition system designed first by Abinet in [1] and later on implemented by Abera in [14]. Abinet in [1] designed her algorithm without considering the constraints of the handheld devices such as PDAs (Personal Digital Assistant). Abera took this into consideration and developed his system for PDAs. Therefore, in order to give the recognized character to the word prediction system as an input, we have to adopt the algorithms for Amharic handwriting recognition developed by Abinet and Abera. However, the works of both Abinet and Abera are just to recognize and display a single character not to write Amharic text. To work on word prediction for online handwriting recognition, an online handwriting recognition text editor is required. The proposed word prediction model requires, as it has been discussed in section 3.1, writing the first two characters of a word before prediction.

```
Input – The corpus and Lexicon tables
Output – The updated Lexicon table
//corpusWord and corpusFreq defines the words and frequencies found in the corpus
//Count – counts the number of records in the corpus
Count  <– 1
Read corpus table
Read Lexicon table
Repeat
        Read corpusWord[count]
        Repeat
                Search for matching words in the Lexicon table
                If (found)
                        Frequency of word in the Lexicon <– corpusFreq[Count]
        Until (end of records in the Lexicon table)
        Count <– Count + 1
Until (end of records in the corpus table)
```

Figure 3.3: Algorithm to Map Frequencies of Words from Corpus to Lexicon

To let the character recognition system work automatically, and recognize more than one character written consecutively, we need to set a wait time limit between two characters of a word. To determine this time, two options were considered. The first is to take the time it takes to write a character with more than one stroke, and use this time as the waiting time before starting to recognize the character. However, this option will let characters with single stroke to wait much time in order to be recognized which will reduce the efficiency of the system. The second option is to get the time difference between the Pen-Down and Pen-Up times between strokes while writing each character which has two or more strokes and calculate the average of all the time differences.

We choose the second option as it gives better efficiency than the first one. An algorithm, shown in Figure 3.4, is designed to accomplish the task of the second option. This algorithm collects the pen-down and pen-up times between strokes of each character except the pen-down time of the first stroke and the pen-up time of the last stroke. It, then, calculates the inter-stroke time difference by subtracting the pen-up time from the pen-down time. Finally, it calculates the average of all the time differences of all Amharic characters.

Table 3.4 shows the Amharic letters which are written using two or more strokes, the number of strokes used, the inter-stroke time gap of five trials, the average time of the five trials of each character, and finally the average time difference of all characters. The data is collected from a single subject as the system is writer-dependent system. Mouse has been used as an input device to collect these data.

As it is shown in this table, the average time difference of all characters is 1932.85 milliseconds. In order to use this time difference as a waiting time to the system, we think it is

better if we round it to 2000 milliseconds. This time will be used by the system to start recognizing as well as predicting if a user doesn't start writing within this period of time. If, during collecting data, pen (stylus) has been used to write characters, the average time difference would have been much less than this.

```
//Calculating the Average of all the Time Differences between penUpTime and
//penDownTime
// Count and Count1 – count the number of inter-stroke gaps
//totalDifference – defines the sum of all the differences
//averageTimeDifference – define the average of all the diffeneces

//Input – Pen-Up time of the current Stroke and Pen-Down time of the next stroke
//Output – The average time of all time differences

penUpTime <– 0,       penDownTime <– 0,   Count <– 0
Count1 <– 0,
//To get the difference of each stroke
Repeat
        penUpTime <– The pen-up time of the current stroke
        penDownTime <– the pen-down time of the next stroke
        timeDifference [count] <– penDownTime – penUpTime
        Count <– Count + 1
Until (all basic Amharic characters are written)
//To calculate the total and average difference time
totalDifference <– 0
Repeat
        totalDifference <– totalDifference + timeDifference[Count1]
        Count <– Count + 1
Until (all the time differences are added)
averageTimeDifference <– totalDifference / count
```

Figure 3.4: Algorithm to find the Average Time Difference of all the Time Differences between Strokes for Basic Amharic Characters having Two or More Strokes

Table 3. 4: The List of Time-Differences between Strokes in Milliseconds

Letter With Two or More Strokes	No. of Strokes	Inter-Stroke Gap (1st Trial)	Inter-Stroke Gap (2nd Trial)	Inter-Stroke Gap (3rd Trial)	Inter-Stroke Gap (4th Trial)	Inter-Stroke Gap (5th Trial)	Average Time
ለ	2	2210	2330	2220	3120	2600	2496
ሐ	3	2270	1400	1230	2570	3260	2146
		3540	2040	1030	4770	4740	3224
ሠ	2	1530	1700	1210	3290	4190	2384
ሰ	2	1140	1860	2570	3160	2900	2326
ሸ	3	1790	2230	1950	5020	3230	2844
		1300	1480	880	4790	3320	2354
ቀ	2	1640	1330	1420	3300	2650	2068
ተ	2	1360	1490	1550	2490	2360	1850
ቸ	3	1280	2120	1250	1670	2440	1752
		940	2330	1030	1550	2450	1660
ኀ	2	3310	1740	1110	1500	4460	2424
ኘ	2	1230	780	1480	980	1790	1252
አ	2	3050	1630	1510	3040	2900	2426
ኸ	3	1220	2750	1410	2650	2790	2164
		910	1300	1040	1510	2400	1432
ዘ	3	1170	1630	2170	1150	2600	1744
		1990	1890	780	2040	1810	1702
ዠ	3	2650	2090	2560	1760	2670	2346
		1950	1000	1560	1120	2510	1628
ደ	2	1270	810	750	2100	2840	1554
ጀ	4	2070	1560	1420	1310	2400	1752
		1780	1590	1300	2390	2480	1908
		1090	1310	1310	1880	2190	1556
ጠ	2	1670	1110	1410	2760	1220	1634
ጨ	2	1920	2390	1530	2810	1630	2056
ጰ	3	920	990	1880	970	770	1106
		3390	1460	1310	2510	1370	2008
ጸ	2	2180	1000	1440	1700	1170	1498
ፀ	2	1140	1450	1060	1710	1220	1316
ፈ	2	1750	3480	980	1980	1580	1954
ፐ	2	1180	960	1030	830	900	980
ቨ	2	6320	1880	1020	970	1010	2240
						Average	**1932.85**

2. Architecture of the System

As shown in Figure 3.5, the architecture of the system has two parts. The first part, Character Recognition Engine, has been discussed in detail by Abinet and Abera in [1] and [14] respectively. The second part of the architecture, Word Prediction Engine, which is circled using dotted line, will be discussed in detail as it is the main concern of this work.

The recognized characters from the character recognition engine and the list of all words in the Amharic lexicon will be provided as an input to the proposed Word Prediction Engine. As it has been mentioned in this chapter earlier, every word in the lexicon has a frequency value which is assigned from the corpus. To start predicting the intended word the prediction engine needs two or more characters. Hence, the character recognition process will be done repeatedly at least twice and at most n times, where n is the length of the word, as there will be a probability that a user writes a word which does not match any word in the lexicon (dictionary).

The Word Prediction Engine has three components which are participating in the prediction process. These are: Start Engine, Word Selector and Word Ranker. The ***Start Engine*** component, for the first time, gets two recognized characters one by one from the character recognition engine. After these two characters are received, this component waits for the time frame set up, which is 2000 milliseconds, to initiate the Word Selector component. On the next process, in case the intended word is not predicted, the Start Engine component initiates Word Selector component after getting every recognized character.

Once the ***Word Selector*** is initiated, it will start searching for words in the dictionary (lexicon) of which their first two or more characters match with the recognized characters. The list of words found and their corresponding frequency will be delivered to the next component, Word Ranker, of the word prediction engine**.**

Word Ranker, by considering the frequency of each word, provides a rank to each word in the list of found words. Words with highest frequency will get highest rank and those with least frequency will get the least rank. In the case of two or more words having the same frequency; the Word Ranker will decide the rank of the words by considering their alphabetical order. These ranking policies are used to determine the word(s) to be predicted.

As mobile devices have screen size constraints, it is not possible to display large number of predicted words so that a user can select his/her desired word from the list. Thus, some trials are done to decide the convenient number of words to be displayed. As a result of these trials, it is found that displaying five words at a time will be much convenient. Hence, the Word Prediction Engine will display the top five ranked words as an output of the system.

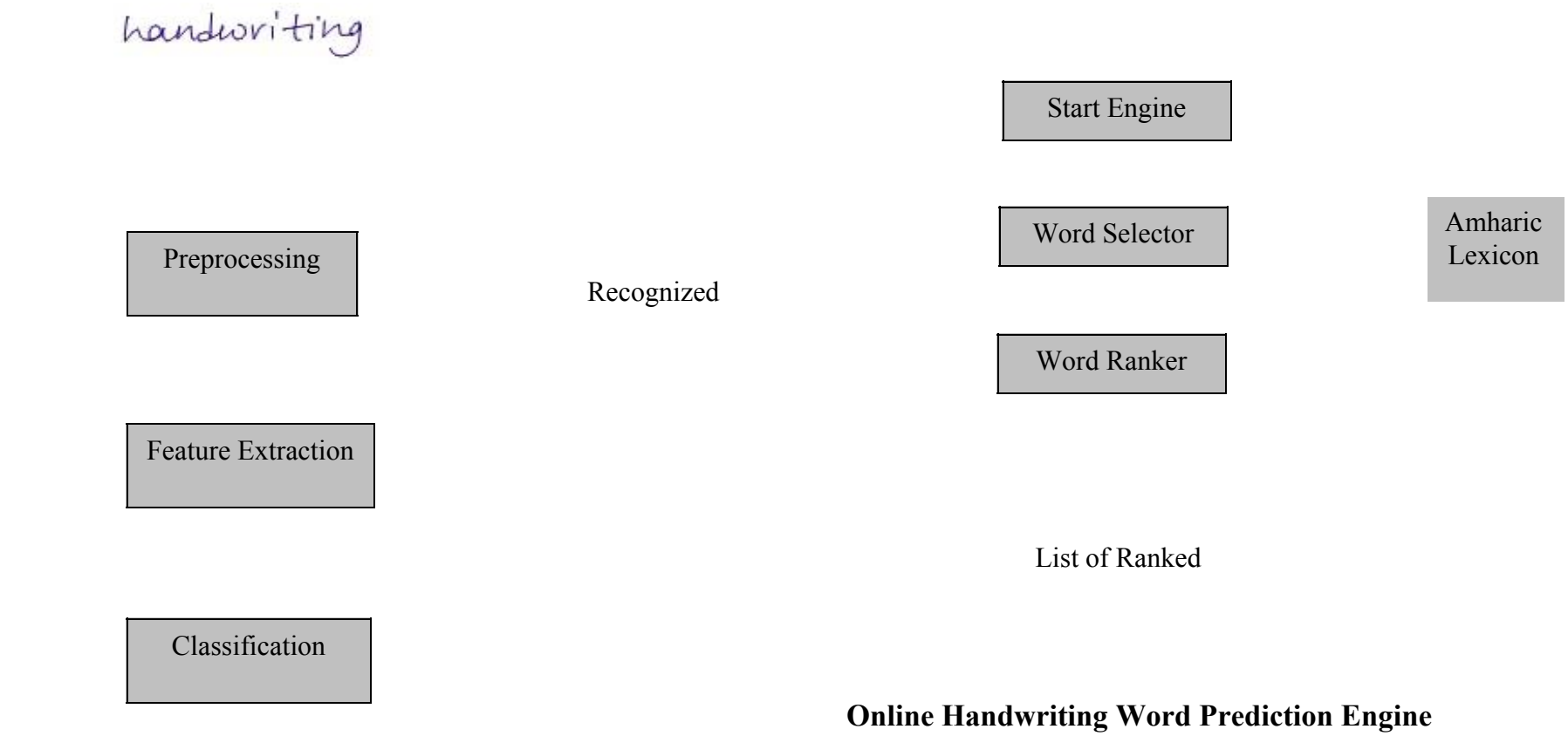

Figure 3.5: Architecture of Online Handwriting Amharic Word Prediction Engine

3. Prediction Algorithm

This section deals with the developed prediction algorithm, shown in Figure 3.6. This algorithm starts by reading the dictionary file. To start the prediction process for the first time, a user writes two Amharic characters using pen (stylus), or mouse and these characters will be stored every time they are recognized.

After waiting for 2000 milliseconds, the two recognized characters will be passed to the Searching method. This method will then search for words in the dictionary that have the same first two characters as the recognized characters. As a result, the list of words is passed to another method called Ranking so as to perform ranking based on the ranking policy of the system.

Once the ranking process has done, the word prediction engine will display the top five ranked words, if any. If the intended word is not displayed, a user will start writing the next character of the word he/she wants to write. Based on this additional character, the word prediction engine starts to perform the prediction process again and display the new result. This process will keep on until either the intended word is predicted or the user writes the last character of the word.

```
Input: Words from the dictionary and the recognized characters
Output: The predicted word and list of alternative words
Read the Lexicon file
Str ← null, Ch ← null, count ← 0
Sorted [ ], List [5] // Arrays used to store the sorted words and the top five ranked words
                     //respectively
ListOfWords [ ] //An array used to store words that matches the string.
Repeat
        User writes a character
        Ch ← Call character Recognition Method
        Str ← Str + Ch
Until (string length ≥ 2)
Wait (2000)                 //Waits for 2000 milliseconds to initiates the prediction process
Call Searching (Str)        //This method searches words based on the Str and pass
                            // the list of words to Sorting method with their frequency
Sorted [ ] ← Call Ranking (ListOfWords [ ]) //Sort the list of words by their frequencies and
                            //returns the list
Repeat
        List [count] ← Sorted [count]
        count ← count + 1
Until count >= 5
Display List [ ]
```

Continued to the next page

```
If (More character is written)
        count ← 0
        Ch ← Call character Recognition Method
        Str ← Str + Ch
        Call Searching (Str)   //This method searches words based on the Str and pass
                                // the list of words to Sorting method with their frequency
        Sorted [ ] ← Call Sorting (ListOfWords [ ]) //Sort the list of words by their frequencies
                                //and returns the list
        Repeat
                List [count] ← Sorted[count]
                count ← count + 1
        Until count > = 5
Display List[ ]
```

Figure 3.6: Algorithm for Word Prediction

4. Summary

In this chapter, the necessity of the corpus and how the corpus is prepared has been discussed. This corpus contains the list of words and their frequency of occurrences in the source document. This corpus has been used to know what the average Amharic word-length is, which word-length is the most frequently used one, and how wide the search space will be. Therefore, based on the information obtained from the corpus, the N for N-gram, where N is the number of characters to start prediction, is decided.

The inputs, the components of the word prediction engine, the function of the components and the relationship between components have been looked into in this chapter. Besides, the number of words that will be listed at a time as an output of the prediction engine is decided. We have also designed different algorithms that help the word prediction engine to accomplish its task.

CHAPTER FOUR

Implementation and Experiment

In this chapter, the tools and environments that are used to implement the designed algorithm and the experiment that is conducted to demonstrate the word prediction accuracy will be presented.

1. Implementation

Developing a prototype to demonstrate the validity and usability of the proposed word prediction system is one of the objectives of this work. In order to implement the algorithms and make the necessary experiment on the system we have used different tools and development environments. This section will talk about the tools and development

environments used to implement the algorithms and the interfaces used for training and prediction purposes.

1. Used Tools and Development Environment

SuperWaba is one of the software development platforms which are designed for handheld devices. It is an open-source, Java-compatible platform [56, 61].

The SuperWaba SDK (SuperWaba Software Development Kit), which runs under Windows operating system, contains the SuperWaba Virtual Machine (VM) and class libraries. The VM is modified to particular type of handheld devices, for example, devices running Palm and Windows CE Operating Systems. SuperWaba SDK has two versions: Community and Professional. The professional version has a package, PDBDriver or Litebase, which can facilitate the searching operation [56, 58, 61]. However, because of its inaccessibility the Community version (free version) has been used in this work.

There are two files, SuperWaba.pdb and SuperWaba.prc, in ***SuperWaba SDK*** for devices that use PalmOS (Palm Operating System). The first file is the Palm database file which contains the SuperWaba class libraries and the second one is the Palm executable file which contains the actual SuperWaba VM applications [56, 57]. PalmOS is an operating system developed by Palm Source Inc.; is a 32-bit operating system running inside PDA and other embedded systems [59, 60]. SuperWaba has two instructions, ***Warp*** and ***Exegen,*** which are necessary to create .pdb and .prc files for the user application respectively [57].

Tauschke MobileCreator is an IDE that is used to build mobile applications. It is designed to be used with SuperWaba 5 [63]. In this work, the MobileCreator Personal 1.6 version is used for writing codes.

As the standard PalmOS font does not support full Unicode ranges [62], it is not possible to display Ethiopic characters on the interfaces prepared. Thus, ***ufolib*** tool that generates font file with full Unicode ranges is used. Thus, it helps us to use Unicode character encoding standard in this work.

The ***PalmOS Garnet Simulator*** software is used to simulate the device running PalmOS 5 to test the system under Windows operating system. In addition to the above discussed tools, Java 2 Standard Edition has been used to implement the designed algorithms and MySQL has been used to manage the lexicon and corpus databases.

2. User Interfaces

This section presents the developed interfaces which are used for the training of the Amharic characters in the online handwriting recognition system and the proposed word prediction system. To test the prediction system, the system should first be trained with the user's handwriting since the writer-dependent system of Abinet [1] and Abera [14] is adopted. Thus, to train the system the interface, shown in Figure 4.1, has been prepared. This interface

consists of some user interface components which are used to accomplish the training process. These components are:

- *A Text Box* – to display the next character to be trained

- *An Input Area* – to write the desired character online using an input method. In this case a mouse.

- *A Button* – to store the sequence codes of the written character and to display the next character to be trained.

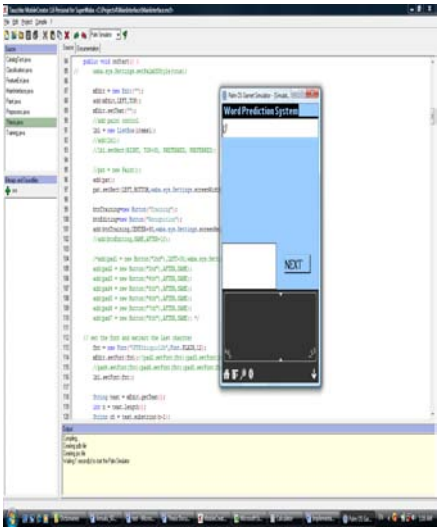

Figure 4.1: Screen Shot of the Training Interface

Once the training process has finished, the Amharic word prediction system for online handwriting recognition can be ready to be used. However, as it has been discussed, a writer

must write two characters consecutively so as to initiate the prediction system. The interface, shown in Figure 4.2, is prepared to perform this operation. The interface has some user interface components which will be used in this operation. These components are:

- *Buttons* – to initiate the recognition process and to display the non-basic characters of the recognized character

- *An Input Area* – to write the desired character

- *A Text Box* – to display the recognized characters afterwards to display the top most ranked word

- *A List Box* – to display four top ranked words as an alternative to the word displayed on the text box.

One has to click one of the buttons labeled 2nd through 8th (which stands for the 2nd through the 8th order of the basic Amharic letters) whenever he/she wants to write the non-basic character accordingly. The moment two characters are written, the system will automatically start the prediction process. The word shown in the text box on the figure 4.3 is the top ranked predicted word and the words listed in the list box are the alternative words on which the user will select if the word in the text box is not the intended word. These words are the predicted words from Rank 2 to 5.

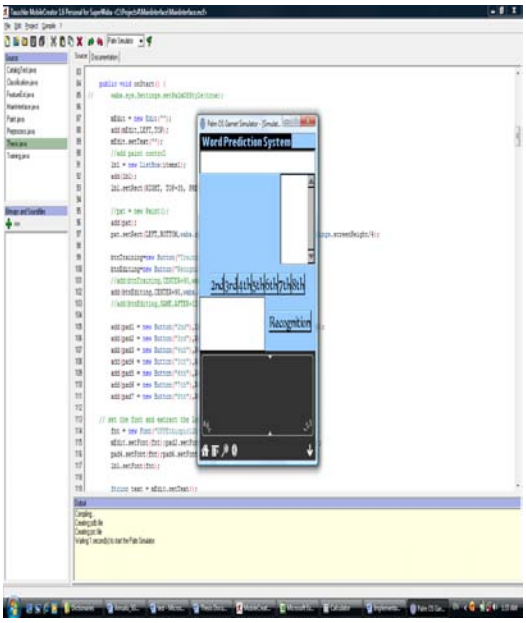

Figure 4.2: Screen Shot of the Word Prediction Interface

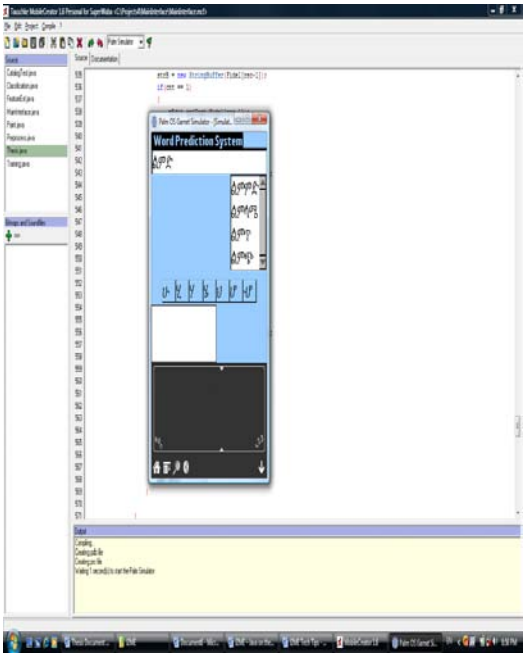

Figure 4.3: Screen Shot of the Predicted Words

2. Experiment

The experiment conducted in this work, will determine the accuracy of the online handwriting word prediction system for Amharic language. This section presents how the test data has been collected and the word prediction accuracy of the system.

1. Data Collection

The test data of this work is collected from two private newspapers, Addis Admas and Addis Neger which are published on Saturday May 24, 2008 by Admas Advertizing PLC and Mesfin Publishing respectively and twenty volunteer persons who are randomly selected. These volunteers are composed of 13 males and 7 females which have different Educational backgrounds and different professions (See Table 4.1).

Table 4.1: Educational Background and Gender of the twenty persons

Educational Background	Male Subjects	Female Subjects
12 Completed	3	1
TVET Students (IT and Secretarial Science)	4	2
Diploma Holders		3
Bachelor Holders	4	
Post Graduate Students	2	1

In addition to educational background, the individuals have different professions. Out of the 20 individuals three of them (one Bachelor holder and two 12 Completed) were business men,

four were college instructors (one post graduate student and three bachelor holders), two of them were working at private college as receptionist (TVET student) and cashier (diploma holder), two diploma holders were working as secretary in government and private offices, and the rest were students and unemployed.

Each individual is requested to write 20 words of his/her preferences. By the time of data collection, none of the individuals were informed the intention why this data is collected and none of them were informed what others have written. The reason behind this is to get more variety of words and to know whether words are frequently written or not. In addition to this, 50 words are collected from each newspaper randomly. This makes the test data total of 500 words. Thus, the system is tested using these 500 words.

The experiment is conducted on Palm OS Garnet Simulator software. This simulator needs the SuperWaba SDK library files, the font file designed by the tool discussed in section 4.1.1 and the .PDB and .PRC files. Mouse is used to write characters to be recognized and Windows Vista Operating System is used.

2. Result of the Experiment

This section presents the result that is found after the experiment has been conducted. The result, shown in Table 4.2, depicts the number of predicted and fully written words for the corresponding word-lengths. The column named as Top Ranked Predicted Word represents the number of highest ranked predicted words after two characters were written. These words are automatically displayed in the text box. The column, Predicted Words from Rank 2 to 5, shows the number of predicted words after two characters were written, but these words are

found in the list of predicted words ranked 2 through 5. These words need to be selected from the list. The last five columns indicate that the number of words either predicted after 3 or more characters or fully written words depending on the word-length. For example, the column, ***After 4,*** shows the number of predicted words in the case of 5-, 6-, or 7-letter words and the number of fully written words in the case of 4-letter word. The shaded cell of each row in this table, Table 4.2, represents the number of fully written words, by the writer, for the corresponding word-lengths.

In addition to this, Table 4.3 shows the percentage of predicted or fully written words of the respective word-lengths. In this table, the shaded cell of each row represents the percentage of fully written words of the corresponding word-lengths.

Table 4.2: The Number of Predicted and Written Words

Word Length	Collected Words	Top Ranked Predicted Words	Predicted Words from Rank 2 to 5	After 3	After 4	After 5	After 6	After 7
3-Letter	229	88	97	44				
4-Letter	208	73	87	15	33			
5-Letter	43	18	11	7	0	7		
6-Letter	12	3	2	0	2	3	2	
7-Letter	8	0	0	2	3	1	0	2
Total	**500**	**182**	**197**	**68**	**37**	**11**	**2**	**2**

Table 4.3: The Percentage of Predicted and Written Words for each Word Length

Word Length	Predicted (%)	Listed (%)	After 3 (%)	After 4 (%)	After 5 (%)	After 6 (%)	After 7 (%)
3-Letter	38.43%	42.36%	19.21%				
4-Letter	35.10%	41.83%	7.21%	15.87%			
5-Letter	41.86%	25.58%	16.28%	0.00%	16.28%		

6-Letter	25.00%	16.67%	0.00%	16.67%	25.00%	16.67%	
7-Letter	0.00%	0.00%	25.00%	37.50%	12.50%	0.00%	25.00%

It is obvious that the efficiency of the word prediction system will decrease while a user selects a word from the listed alternative words. However, since the main target of this work is to determine the word prediction accuracy, the percentages of all columns of each word length except the last one will be added to get the word prediction accuracy of the particular word length. The cumulative word prediction accuracy of each word-length and the average word prediction accuracy of the system are shown in Table 4.4. As it is shown in this table, the result of the experiment indicated that the average word prediction accuracy of the system is 81.39%.

Table 4.4: Prediction Accuracy of each Word Length

Word Length	Words Predicted	Prediction Accuracy
3-Letter	185	80.79%
4-Letter	175	84.13%
5-Letter	36	83.72%
6-Letter	10	83.33%
7-Letter	6	75.00%
Average Prediction Accuracy		81.39%

3. Summary

Different tools and environments such as Tauchke Mobile Creator, SuperWabaSDK, etc were used to develop the prototype and interfaces for Training and prediction purpose. An experiment has been conducted using 500 words collected from 20 volunteers and two private newspapers in order to validate the word prediction system proposed and the advantage that

can be gained by the use of Amharic word prediction system. The experiment has shown that a prediction accuracy of 81.39% is achieved.

The word prediction accuracy of the system can be increased if standard and properly digitized Amharic dictionary (lexicon).

CHAPTER FIVE

Conclusion and Future Works

Word prediction is a process of predicting a word that a user intends to write after the first few characters and based on a lexicon and statistical information obtained from the corpus.

Text writing using an online handwriting recognition system follows a complex process that also requires training. When word prediction is used for handwriting recognition, the possible errors of character recognition can be minimized.

To accomplish the task of word prediction process for Amharic language, a good collection of words which will be used as a corpus is a must. However, since there is no such corpus for Amharic language, collecting of words from different sources and preparing the corpus became the first task of this research.

Analyses have been done on the corpus prepared. These analyses used to get information like the average word-length of Amharic language, the most frequently used Amharic word-length and the like. These information have been used to decide the core element of word prediction engine which is N for N-gram model, where N is the number of characters after which the prediction process starts. Based on the analyses done, the value of N has been decided to be two (N=2).

Moreover, in this work, an online handwriting word prediction engine for Amharic language has been developed. The algorithms required for assigning word frequencies from the corpus to the dictionary (lexicon) and for predicting words have been designed and implemented.

Based on the experiment conducted our word prediction system has shown a prediction accuracy of 81.39%. In this work, the searching efficiency of the system has not been taken into consideration.

As this is the first ever work in designing on Amharic word prediction system, it is recommended to suggest the possible future works in this domain. Thus, we identify the following works in the future:

- As it has been mentioned so many times in this work, having a complete corpus is the base to do word prediction. Thus, preparing error adequate and better size corpus must be one of the tasks that need to be done in the future. In addition to this, having a standard dictionary with maximum possible word size is very important and will increase the accuracy of the word prediction system.

- This work can be done for other local languages like Tigrigna etc. which use Ethiopic character sets.

- Since the searching efficiency of the system has not been taken into consideration in this work, one can extend this work by adding this feature. We suggest the

professional version of SuperWaba SDK and some data structures in order to attain this goal.

- Integrate this word prediction system with Amharic text editing systems.

- Extend this work into context-level prediction system

- Extend this work to phrase and sentence level prediction system

References

1. Abinet Shimeles “Online Handwriting Recognition for Ethiopic Characters” (Masters Thesis), Department of Computer Science, Addis Ababa University, June 2005.

2. Fadi Biadsy, Jihad El-Sana, and Nizar Habash. “Online Arabic handwriting recognition using Hidden Markov Models”, In Proceedings of the 10th International Workshop on Frontiers of Handwriting and Recognition, 2006.

3. Daniel Negussie “Writer Independent Online Handwriting Recognition for Ethiopic Characters” (Masters Thesis), Department of Computer Science, Addis Ababa University, June 2006.

4. Saied B. Nesbat , “A System for Fast, Full-Text Entry for Small Electronic Devices”, Proceedings of the Fifth International Conference on Multimodal Interfaces, ICMI 2003 (ACM-sponsored), Vancouver, November 5-7, 2003.

5. Shiferaw Abebe, Tewodros Seyoum, Solomon Atnafu, Samuel Kinde Kassegne, “Ethiopic Keyboard Mapping and Predictive Text Inputting Algorithm in a Wireless Environment”, ITEs-2004, Addis Ababa, Ethiopia, 2004

6. Esra Vural, Hakan Erdogan, Kemal Oflazer, Berrin Yanikoglu, "An Online Handwriting recognition system for Turkish", in Proceedings of SPIE Electronic Imaging Symposium, 2005.

7. What is Handwriting Recognition, WiseGEEk, http://www.wisegeek.com, April 6, 2007.
8. Homayoon S. M. Beigi, "Character Prediction for On-Line Handwriting Recognition", Proc. of Canadian Conference on Electrical and Computer Engineering, Toronto, Canada, Sep. 13-16, Vol. II, , 1992
9. Kazushi Ishigaki, Hiroshi Tanaka, Naomi Iwayama, "Interactive Character Recognition Technology for Pen Based Computers", Fujitsu Sci. Tech. J., 35, 2, p.191-201, December 1999
10. Niels, R.M.J., & Vuurpijl L.G, "Dynamic Time Warping Applied to Tamil Character Recognition", In Proceedings of the Eight international conference on document analysis and recognition (ICDAR'05), pp. 730-734, 2005.
11. Plamondon R. Srihari S. N. "On-Line and Offline Handwriting Recognition: A Comprehensive Survey" IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol.22, No.1. January 2000.
12. Sornlertlamvanich V. et al,. "The State of the Art in Thai Language Processing" Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000), Hong Kong, pp 597-598, October 2000.
13. V. Govindaraju, A. Shekhawat, and S. N. Srihari "Interpretation of Handwritten Addresses in U.S. Mail Stream", In ICDAR AIPR-IEEE. Proceedings of the Second International Conference on Document Analysis and Recognition, Tsukuba Science City, Japan, October 1993. IAPR.
14. Abera Abebaw, "Implementation of Online Handwriting Recognition System for Ethiopic Character Set", Masters Project, Department of Computer Science, Addis Ababa University, June 2007
15. Barry McCaul and Alistair Sutherland, "Predictive Text Entry in Immersive Environments", Proceedings of the IEEE Virtual Reality 2004 (VR'04), P: 241, 2004
16. "19 Text-Input", Icie.cs.byu.edu/UIBook/19-TextInput.pdf, Last Referenced Date: September 4, 2007

17. Kumiko Tanaka-ishii, “Word-Based Predictive Text Entry Using Adaptive Language Models”, Natural Language Engineering 13 (1): 51–74. 2006 Cambridge University Press, 15 February 2006

18. MacKenzie, I. S., & Soukoreff, R. W, “Text Entry for Mobile Computing Models and Methods, Theory And Practice”, Human-Computer Interaction, 17, 147-198, 2002

19. MacKenzie, I. S., Nonnecke, B., Riddersma, S., McQueen, C., & Meltz, M., “Alphanumeric Entry on Pen-Based Computers”, International Journal of Human-Computer Studies, 41, 775-792, 1994

20. Isabelle Guyon & Colin Warwick, “Handwriting as Computer Interface”, http://cslu.cse.ogi.edu/HLTsurvey/ch2node7.html, August 08, 2007

21. V. Jagadeesh Babu, L. Prasanth, R. Raghunath Sharma, G.V. Prabhakara Rao, “HMM-Based Online Handwriting Recognition system for Telugu Symbols”, Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 1, pp 63-67, 2007

22. Nawwaf N. Kharma and Rabab K. Ward, “Character Recognition System for None Expert”, IEEE Canadian Review – Autumn/Automne (University of British Columbia), 1999

23. Peter Burrow, “Arabic Handwriting Recognition”, Master of Science Thesis School of Informatics University of Edinburgh, 2004

24. Claus Bahlmann and Hans Burkhardt, “The Writer Independent Online Handwriting Recognition System Frog on Hand and Cluster Generative Statistical Dynamic Time Warping”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 3, pp 299-310, March 2004

25. Hiroshi Tanaka, Naomi Iwayama, Katsuhiko Akiyama, “Online Handwriting Recognition Technology and Its Applications”, FUJITSU Sci. Tech. J., Vol 40-1, June 2004

26. Charles C. Tappert, Ching Y. Suen, and Toru Wakahara, “The State of the Art in Online Handwriting Recognition”, IEEE Transactions on Pattern Analysis and Artificial Intelligence, Vol:12(8), Pp: 787 - 808, 1994

27. Sutat Sae-Tang and Ithipan Methaste, “Thai Online Handwritten Character Recognition Using Windowing Back-Propagation Neural Network”, Proceeding Modelling, Identification, and Control, 2002
28. “Online Handwriting Recognition”, http://www.visionobjects.com/handwriting-recognition/ , April 06, 2007
29. Scott D. Connell, Anil K. Jain, “Writer Adaptation for Online Handwriting Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 3, March 2002
30. Alessandro L. Koerich, “Large Vocabulary Off-line Handwritten Word Recognition”, PhD dissertation, # EEcole de Technologie Superieure de l'Universite du Quebec, Montreal, Canada, 2002
31. Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari, “Word Recognition with Multi-level Contextual knowledge”, Proc. of the 1st Int'l Conference on Document Analysis and Recognition, 1991
32. J. Park and V. Govindaraju, “Using Lexical Similarity in Handwritten Word Recognition”, In IEEE Conference on Computer Vision and Pattern Recognition, Hilton Island, South Carolina, 2000
33. Xia Liu, Zhixin Shi, “A Format-Driven Handwritten Word Recognition System”, Proceedings of the Seventh International Conference on Document Analysis and Recognition, Vol2 p 1118, 2003
34. Claudie Faure & Eric Lecolinet, “OCR: Handwriting”, http://cslu.cse.ogi.edu/HLTsurvey/ch2node6.html, Last Referenced Date: August 08, 2007
35. Hedy Kober, Eugene Skepner1, Terry Jones1, Howard Gutowitz1, and Scott MacKenzie, “Linguistically Optimized Text Entry on a Mobile Phone”, http://www.eatoni.com/research/chi.pdf, Last Referenced Date: April 30, 2007
36. Stuart M. Shieber and Rani Nelken , “Abbreviated Text input using Language Modeling”, Cambridge University Press, Vol 13: pp 165-183, Jan 2007
37. Gregory W. Lesher, Gerard J. Rinkus, “Leveraging Word Prediction to Improve Character Prediction in A Scanning configuration”, Proceedings of the RESNA 2002 Annual Conference, 2002

38. G. W. Lesher, B.J. Moulton, D.J. Higginbotham, and B. Alsofrom, "Limits of Human Word Prediction Performance", CSUN 2002, California State University, Northridge, 2002
39. Hisham Al-Mubaid, "Context-Based Word Prediction and Classification", The 18th International Conference on Computers and Their Applications CATA-2003, 2003
40. "Word Prediction–What's Good Enough?", http://www.donjohnston.com/products/cowriter/word_prediction.html, Last Referenced Date: April 06, 2007
41. Christina L. James, Kelly M. Reischel, "Text Input for Mobile Devices Comparing Model Prediction to actual performance", Proceedings of the SIGCHI conference on Human factors in computing systems, Seattle, Washington, United States, pp 365 - 371, 2001
42. "Tegic Communications Delivers Version 7.2 of Industry Leading T9® Text Input Software", http://www.timewarner.com/corp/newsroom/pr/0,20812,795551,00.html, Last Referenced Date: Sep 4, 2007
43. "Zi Corporation Extends the Global Reach of its Product Portfolio at 3GSM",http://www.canadait.com/cfm/index.cfm?It=106&Id=21697&Se=2&Sv=&Lo=442, Last Referenced Date: Sep 4, 2007
44. "Linux phone stack gains predictive text input software", http://www.linuxdevices.com/news/NS5212042442.html, Last Referenced Date: Sep 4, 2007
45. "Predictive Text for the Rest of Us", http://www.eatoni.com/wiki/index.php/Main_Page, Last Referenced Date: Sep. 4, 2007
46. "LetterWise – Eatoni", http://www.eatoni.com/wiki/index.php/LetterWise, Last Referenced Date: Sep 4, 2007
47. Worku Alemu, "The Application of OCR Technique to the Amharic Script", Masters Thesis, Addis Ababa University , 1997
48. Yonas Hailu, "Simplified Ethiopic Script for Online Handwriting Recognition" (Masters Thesis), Department of Computer Science, Addis Ababa University, June 2007

49. Miika Silfverberg, I. Scott MacKenzie, Panu Korhonen, "Predictive Text Entry Speed on Mobile Phones", Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI 2000, pp. 9-16. New York: ACM. , 2000

50. Gregory W. Lesher, Bryan J. Moulton, D. Jeffery Higginbotham, "Effects of N-gram Order and Training Text Size on Word Prediction", http://www.dynavoxtech.com/files/papers/LeMoHi99.pdf, Last Referenced Date: May 24, 2007

51. "Predictive Text", Wikipedia, the Free Encyclopedia, http://www.wikepedia.com, April 13, 2007.

52. Jun Gong, Peter Tarasewich, "Testing Predictive Text Entry Methods with Constrained Keypad Designs", Proceedings 2005 HCI International Conference, 2005

53. "Handheld Terms to Know", http://www.smartcomputing.com/, Featured Articles Vol.1 Issue 12, June 2006 •

54. "List of Lexicons", http://nlp.amharic.org/resources/lexical/word-lists/toponymic, Date Visited: September 2007

55. "WordSmith Lexical Analysis Software for PC", http://www.lexically.net/wordsmith/index.html, Date Visited: September 2007

56. "Building handheld applications with Waba", ibm.com/developerWorks, Date Visited: February 2007

57. "SuperWaba_Companion_GPL", http://www.superwaba.com.br/en, Date Visited: May 19, 2008.

58. "Palm Programming with Waba", http://www.onjava.com/pub/a/onjava/2001/04/19/java_palm.html, Date Visited: March 19, 2008.

59. "PalmOs programming in Java", http://membres.lycos.fr/pbriol/palmos/java.html, May 19, 2008.

60. "Handheld Computing and Palm OS for Mobile Handheld Devices", http://www.micsymposium.org/mics_2006/papers/HuKaoYangYeh.pdf, Date Visited: March 19, 2008

61. “Develop Portable Mobile Application”, http://www.superwaba.com.br/etc/SuperWabaFolderEnV3.pdf, Date Visited: March 19, 2008.
62. “UFOLIB”, http://jdict.sourceforge.net/ufolib/, Date Visited: March 25, 2008.
63. “Tauschke MobileCreator”, http://www.tauschke.com/products/tauschkemobilecreator/index.html, March 25, 2008

Appendices

Appendix A: List and Description of documents used to prepare the corpus

Sr. No	Title	Type	Author	Date
1	Abaye	News Paper		1997/98
2	Addis Admas	,,		1992/93/94/96/97/98/99
3.	Addis Zena	,,		1998
4.	Askwal	,,		1998
5	Ethiop	,,		1998
6	Hadar	,,		1998
7	Hzbawi	,,		1998
8	Lsanehzb	,,		1998
8	Minilik	,,		1998
9	Nation	,,		1998
10	Netsanet	,,		1998
11	Reporter	,,		1998/99
12	Satenaw	,,		1998
13	Tobia	,,		1998
14	Tomar	,,		1998
15	Zemonitor	,,		1998
16	Addis Raey	,,		1993
17	Addis Zemen	,,		1990/93/95/96
18	Agerie	,,		1990
20	Amero	,,		1990
22	Andinet	,,		1990
23	Askwal	,,		1992
27	Hayat	,,		1991
28	Hikma	,,		1991
19	Aleweledim	Book	Abe Gobegna	NI
21	Amharic Teret	,,	NI	NI
24	Awdenegast	,,	NI	NI
25	Etege Taitu	,,	Tadesse Zewolde	NI
26	Fiker Eiskemekaber	,,	Haddis Alemayehu	NI
29	Kidus Yohannes	,,	Tarekegn Birhanu	NI
30	Medhane Alem	,,	NI	NI
31	Melkee Giyorgis	,,	Tesfa G/Silasie	NI
32	Menged Semay	,,	Tesfa G/Silasie	NI
33	Minelik	,,	Gizaw H/Mariam	NI
34	Oromay	,,	Bealu Girma	NI
35	Tentawe Mesale	,,	Moges Equibegiorgis	NI
36	Wideder	,,	NI	NI
37	Witietama Yetadagi Wetatoch	,,	Nardos Mamo	NI

** NI – Not*

Appendix B: List of the Top Most 5 Number of Words Obtained by pair of Letters

Letter	Words Start with	Number of Words
ሀ	ሀብ	45
	ሃይ	42
	ሀገ	36
	ሁኔ	27
	ሀይ	26
ለ	ለመ	715
	ለ ማ	537
	ሊያ	343
	ለአ	273
	ለተ	260
ሐ	ሕዝ	29
	ሕገ	27
	ሕይ	26
	ሐረ	17
	ሕን	15
መ	ማስ	273
	መን	262
	መስ	241
	መሰ	156
	መል	155
ሠ	ሠራ	49
	ሥራ	43
	ሠር	33
	ሥጋ	30
	ሥር	25
ረ	ሪፖ	25
	ረድ	20
	ራስ	20
	ረስ	17
	ረብ	17
ሰ	ስለ	958
	ሲያ	496
	ሳይ	491
	ስላ	250
	ስት	219
ሸ	ሸማ	22
	ሸን	19

	ሹመ	17
	ሸን	17
	ሸማ	14
ቀ	ቀር	53
	ቀን	49
	ቅን	46
	ቀጥ	45
	ቅር	42
በ	በመ	1180
	በተ	676
	በአ	578
	ባለ	356
	በእ	243
ተ	ተመ	243
	ተቀ	221
	ተሰ	214
	ተከ	210
	ተጠ	206
ቸ	ችግ	48
	ችሎ	27
	ቸር	16
	ቻር	12
	ቻይ	9
ጎ	ኃይ	49
	ኃላ	26
	ኃጢ	23
	ጎብ	23
	ኃሳ	10
ነ	ነገ	115
	ነበ	58
	ንግ	45
	ነግ	44
	ነፍ	44
አ	እን	5880
	እየ	1077
	አይ	1054
	አስ	975
	አል	928
ከ	ከመ	697
	ከተ	637
	ከአ	340
	ከሚ	330
	ከማ	274
ዐ	ዓይ	60
	ዕድ	50
	ዓመ	30

	ዕው	25
	ዓለ	24
ወ	ወደ	502
	ወን	175
	ወዳ	101
	ወይ	91
	ወር	68
ዘ	ዘመ	78
	ዘር	38
	ዝግ	33
	ዘን	28
	ዘገ	27
ደ	ድር	89
	ደረ	87
	ደን	82
	ደር	82
	ድን	80
ጀ	ጀም	30
	ጀማ	30
	ጆሮ	28
	ጀመ	16
	ጃን	14
የ	የሚ	3328
	የተ	2096
	የማ	1049
	የም	1044
	የመ	837
ገ	ገብ	79
	ግን	67
	ግብ	60
	ገል	60
	ጉዳ	51
ፕ	ፕሮ	67
	ፓር	50
	ፕሬ	28
	ፖሊ	25
	ፖስ	12
ጸ	ጸሎ	15
	ጸሐ	13
	ጸን	12
	ጿጋ	11
	ጸላ	10
ፀ	ፀሐ	9
	ፀጥ	8
	ፀጋ	8
	ፀሀ	6

	ፀጉ	6
ፈ	ፈር	43
	ፍላ	41
	ፈረ	40
	ፍር	35
	ፍቅ	34
ጠ	ጥር	50
	ጥያ	44
	ጥቅ	38
	ጠበ	38
	ጠይ	37
ጨ	ጭን	29
	ጨር	23
	ጨም	22
	ጨረ	20
	ጨዋ	20