# AUTOMATIC STEMMING FOR AMHARIC TEXT: AN EXPERIMENT USING SUCCESSOR VARIETY APPROACH

**BY: Genet Mezemir Fikremariam**

**A thesis submitted to the school of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master's of Science in Information Science**

**January, 2009**

**ADDIS ABABA UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

**FACULTY OF INFORMATICS**

**DEPARTMENT OF INFORMATION SCIENCE**

# AUTOMATIC STEMMING FOR AMHARIC TEXT:

# AN EXPERIMENT USING SUCCESSOR VARIETY APPROACH

**BY: Genet Mezemir Fikremariam**

**Name and Signature of Members of the Examining Board:**

---------- , Chair person, Examining Board ----------

Ato Ermias Abebe, Advisor

---------- ---------- ----------

---------- ---------- ----------

Chair person, Faculty Signature Date

---------- ---------- ----------

Chair person, Graduate Council Signature Date

# ACKNOWLEDGEMENT

This thesis is the result of a one semester research work. During this thesis work, many people have involved directly or indirectly to the success of it. Though it is difficult or impossible to list them all explicitly, I would like to thank them in general and express my thanks to only those whose contributions are fairly significant.

First and foremost, my heartfelt gratitude goes to my advisor Ermias Abebe, without whose wholehearted, constructive comments and suggestions, this research would not have been a reality in such a short period of time. My gratitude also goes to Daniel Yacob for his committed responses to the questions I send him via email as well as for the material support he has been providing me. I would also thank my family and parents, for their critical material and moral support wherever needed.

I want also to thank the other staff members and my friends in the Department of Information Science, Faculty of Informatics, Addis Ababa University for whatever contributions they have put into the success of this research. Among the staff members, Ato Mesfin Getachew, who helped me to have a good start, deserves the highest of my thanks for his valuable comments. Last but not least, I would like to thank my colleague in ZTE Corporation, Ato Mezgebe Hailemariam for his great support during my hard times of code debugging.

My friends (Mesfin Worku, Sisay Adugna, and Tsegaw Kelela) thank you all for the unforgettable periods of two years we spent together sharing knowledge and experience.

Genet Mezemir Fikremariam

# TABLE OF CONTENT

**ACKNOWLEDGEMENT** ..... i

**TABLE OF CONTENT** ..... ii

**TABLE OF FIGURES** ..... iv

**TABLE OF TABLES** ..... iv

**LIST OF ABBREVIATIONS** ..... v

**CHAPTER ONE** ..... 1

**INTRODUCTION** ..... 1

1.1 BACKGROUND ..... 1

1.2 STATEMENT OF THE PROBLEM AND JUSTIFICATION ..... 3

1.3 OBJECTIVES OF THE STUDY ..... 4

1.3.1 General objectives ..... 4

1.3.2 Specific Objectives ..... 5

1.4 METHODOLOGY ..... 5

1.4.1 Literature Review ..... 5

1.4.2 Data Collection ..... 6

1.4.3 Experimentation Methods ..... 6

1.4.4 Tools ..... 8

1.4.5 Evaluation ..... 8

1.5 APPLICATION OF RESULTS AND BENEFICIARIES ..... 8

1.6 SCOPE AND LIMITATION OF THE STUDY ..... 9

1.7 ORGANIZATION OF THE THESIS ..... 10

**CHAPTER TWO** ..... 11

**BACKGROUND** ..... 11

2.1 INDEXING ..... 11

2.1.1 Types of indexing ..... 11

2.2 STEMMING ..... 12

2.2.1 Conflation Methods ..... 13

2.3 OVERVIEW OF THE AMHARIC MORPHOLOGY ..... 18

2.3.1 The Amharic Writing System ..... 18

2.3.1.1 Some thoughts on Ge'ez Language speaking and writing rules on similar sounding multiple alphabets ..... 19

2.3.1.2 Sound similarity within the seven letter groups of the 33 alphabet characters ..... 20

2.3.2 Problems with the Amharic writing system ..... 21

2.3.2.1 Alphabets with the same sound, but multiple "redundant" characters ..... 21

2.3.2.2 Gemination of consonants and context sensitivity of Amharic words ..... 23

2.3.2.3 Problems in formation of compound words....24
2.3.2.4 Localized spoken and written variation of identical words, regional dialects....24
2.3.3 The Alphabet, Punctuation and Numbers in Amharic Language....25
2.3.4 Amharic Word formations....28
2.3.5 The Morphology of Amharic Language....29

**CHAPTER THREE**....31

**REVIEW OF RELATED LITERATURES**....31
3.1 INTRODUCTION....31
3.2 REVIEW OF RELATED WORKS....33

**CHAPTER FOUR**....37

**DESIGN AND DEVELOPMENT OF THE SYSTEM**....37
4.1 INTRODUCTION....37
4.2 THE ARCHITECTURE OF THE PROTOTYPE....37
4.3 DESCRIPTION OF SOFTWARE SYSTEMS....38
4.3.1 Document Preprocessor Subsystem....38
4.3.1.1 Document Cleaning Subcomponent....39
4.3.1.2 Characters Normalization Subcomponent....40
4.3.1.3 Transliteration Subcomponent....42
4.3.1.4 Tokenization....42
4.3.2 Successor Variety Stemmer....44
4.3.2.1 Successor variety generation based on Peak and Plateau Algorithm....44
4.3.2.2 Successor variety generation based on Entropy Algorithm....47
4.3.3 Testing Method....53

**CHAPTER FIVE**....55

**EXPERIMENTATION AND ANALYSIS**....55
5.1 INTRODUCTION....55
5.2 THE CORPUS....55
5.3 TRAINING THE SYSTEM....56
5.4 TESTING THE SYSTEM....57
5.5 DISCUSSION OF THE RESULT....58

**CHAPTER SIX**....62

**CONCLUSION AND RECOMMENDATION**....62
6.1 CONCLUSION....62
6.2 RECOMMENDATION....64

**REFERENCES**....66

**APPENDIX**....69

# TABLE OF FIGURES

Figure 1: Taxonomy for Stemming Algorithms ..... 14
Figure 2: The process diagram of Successor Variety Algorithm ..... 17
Figure 3: The Architecture of the prototype ..... 38

# TABLE OF TABLES

Table 1: Phonemes differentiation or categorization in speaking and writing, the emphasis lies on the underlined characters ..... 20
Table 2: Voice similarity within the seven letter groups of the Amharic characters ..... 20
Table 3: Table of Original derivation of words having similar sounding alphabetic symbol, the emphasis is on the underlined alphabets ..... 22
Table 4: Some examples of words with the same writing but context dependant meaning ..... 23
Table 5: Total Number of Characters in Amharic Alphabet (Fidel) ..... 25
Table 6: Amharic numbers ..... 28
Table 7: Gender, number and case marker suffixes ..... 30
Table 8: Sample Successor Variety generated for the word “lemetenten” ..... 46
Table 9: Sample Successor Variety generated and the successor letters with their frequency of occurrence in the corpus for the word “Cewatacewn” ..... 50
Table 10: Sample the prefix of the word and its number of words in a text body beginning with the i length sequence of letters in the corpus ($D_{\alpha i}$) ..... 50
Table 11: Sample Successor Variety generated with the calculated Entropy value for the word “Cewatacewn” ..... 52

# LIST OF ABBREVIATIONS

ENA: Ethiopian News Agency

IR: Information Retrieval

SV: Successor Variety

E.C: Ethiopian Calendar

G.C: Gregorian Calendar

# ABSTRACT

The extensive use of the World Wide Web and the increasing digital availability of information and documents accelerated the demand for technologies and tools for an online data retrieval and extraction application. The natural language research, with the aim of quick and reliable online information searching and access, is one major component of the current advanced information technology development. In this research, an indexing system was developed and programmed by using the Successor Variety Stemming Algorithm to find stems for Amharic words. The research has set out to discover whether the Successor Variety Stemming Algorithm technique with the peak and plateau, entropy and complete word methods can be used for the Amharic language or what the limitation would be. In addition, the peak and plateau method compared with the entropy and the complete words method. Stemming is typically used in the hope of improving the accuracy of the search reducing the size of the index. A corpus of 6270 words was obtained form the Ethiopian News Agency (ENA) and Walta Information Center and used to train and test the methods.

The experiment result showed that, the peak and plateau method had a performance of 71.8% level of accuracy, but the performance of the entropy and complete word methods are 63.95% and 57.99% level of accuracy respectively. Based on the observation made from the experimentation result, the successor variety algorithm with the peak and plateau method had a better performance than successor variety algorithm with the entropy method.

# CHAPTER ONE

# INTRODUCTION

## 1.1 BACKGROUND

Electronic forms of documents and information are exercising a rapid growth with the advancement in information technology. Organizing, managing and getting relevant documents from such huge collection of database are becoming difficult and time consuming. If the aforesaid activities are done manually, the difficulties of getting relevant and appropriate information (document) will become very tough. To overcome the above problems (i.e., organizing, managing and retrieving relevant information) many information processing systems have been developed, including management information systems, database information systems, data retrieval systems and information retrieval systems, among others. With this respect, information retrieval systems are identified as powerful and recent approach to design a mechanism that facilitates the accessibility of stored information (Baeza-Yates and Ribeire-Neto, 1999 and Rijsberen, 1999).

Information retrieval emphasizes on the representation, storage, organization of, and access to information items in general. According to Baeza-Yates and Ribeire-Neto, (1999) the representation and organization of the information items should provide users with easy access to the document in which they are intended or interested, specifically. According to Salton and McGill (1983) a typical information retrieval system informs on the existence or the nonexistence and whereabouts of the documents by relating to the request. The inquired information item includes text, image and video documents that contain partially

matched files with the user's requests. However, information retrieval does not intend to inform or change the knowledge of the user on the subject of his/her inquiry. Rather, the searching engine emphasizes on information containing documents as opposed to the retrieval of data or fact (Rijsberen, 1999).

Information retrieval is aimed to extract all relevant documents for a user query by using index items of natural language text. The text could be unstructured and ambiguous. In order to satisfy users in their searching, it is required to translate user request in to inquiry that can be processed by the information retrieval system. Among others, word stemming is an important attribute supported by recent indexing that produces a set of key words relevant to the document. It enables to improve recall by automatic handling of word endings via reduction of terms to their stems or roots during indexing and searching. Hence, stemming reduces the size of indexing structure and minimizes variants of the same stem or root words in order to have effective searching result (Al-Shalabi *et al.* 2005).

Successor variety is one of the stemming techniques in information retrieval system. Frakes and Beaza-yates (1992) state, the successor variety of a string is the number of different characters that follow the string in words in a corpus (the body of text). The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. Successor variety stemmer does not require preparation of suffix lists and removal rules, and hence can be adapted to changing text collection.

## 1.2 STATEMENT OF THE PROBLEM AND JUSTIFICATION

Amharic is the official working language of the federal government of the Federal Democratic Republic of Ethiopia and is estimated to be spoken by well over 20 million people as a first or second language. Amharic is the second most spoken Semitic language in the world (next to Arabic).

Amharic is widely used in text processing activities in governmental, non-governmental and private institutions. Increased availability of large scale Amharic documents in electronic format and affordable computer resources are good opportunities for developing automatic Amharic information retrieval systems.

Research work in automatic indexing for Amharic still needs much effort. One research in the area of automatic indexing is the work by Alemayehu (2002), conducted to develop a stemmer for the Amharic language. Alemayehu (2002) reported that there are very large numbers of word variants in the Amharic language that result from the complex nature of the morphological structure of the language. It indicates that using automatic stemmer for developing indexes for search engines improves their efficiency and effectiveness of the information retrieval systems.

According to the information obtained from various sources during the assessment of the current situation of the technology in the field (Amharic information retrieval), the following problems are identified:

- There is no available standard affixes dictionary for Amharic text,

- There is no general stemmer to be used in indexing for Amharic text, and
- There is lack of stop-word list.

According to Al-Shalabi *et al. (*2005) Successor Variety Algorithm has the ability to find a stem without the need to use a dictionary. And also, it can be used in several domains.

However, to the researcher's best knowledge, a direct stemming of Amharic words using successor variety stemming technique has not been done before. Since, words in Amharic language have rich set of morphological variants, it seems logical to experiment the applicability of this algorithm to stem words in the language.

Therefore, this study is initiated to explore the potential application of Successor Variety Algorithm for stemming (conflating) words in Amharic language.

## 1.3 OBJECTIVES OF THE STUDY

### 1.3.1 General objectives

The general objective of this research is to develop a prototype Successor Variety Stemmer for words in the Amharic language.

### 1.3.2 Specific Objectives

To accomplish the above general objective, the study focused on the following specific objectives:

- to review the foundation of successor variety algorithm for stemming,
- to review the linguistic features of the Amharic language with respect to information retrieval system,
- to setup the training and the test set by selecting Amharic documents,
- to explore the potential of successor variety algorithm,
- to develop a prototype successor variety stemmer for Amharic text,
- to train and test the prototype by using the training and testing dataset,
- to analyze the result of the experimentation, and
- to draw conclusions based on results of the experimentation.

## 1.4 METHODOLOGY

### 1.4.1 Literature Review

Extensive literature review was done to get more insight into the concept of information retrieval in general and stemming with successor variety in particular. A review of literature was also conducted to get familiarity with the basic Amharic text features in relation to information retrieval. Also, existing works related to this research, were also reviewed and discussed.

### 1.4.2 Data Collection

For the purpose of training and testing the successor variety stemmer, Amharic local news articles were collected from the Ethiopian News Agency and Walta Information Center. The reason of selecting these sources is the articles are available in electronic form. The documents were selected by using judgmental sampling. Because, most of the time local news articles in some specific domain contain specific terms repetitively. It makes the corpus inappropriate for this research. Therefore, using judgmental sampling allowed the researcher to incorporate representative terms for the corpus.

### 1.4.3 Experimentation Methods

Hafer and Weiss (1974) reported that 2000 terms can be stable number for successor variety algorithm in English language. From the source of data 6270 words were found. Therefore, the researcher used the corpus size of 6270 words for this research work. The selected corpus was divided into training set and testing set as a ratio of eighty-twenty. Eighty percent of the corpus was used for training and the rest twenty percent for testing; as is widely used across the researchers in this subject area.

According to Al-Shalabi et al.(2005), the successor variety of a string is the number of different characters that follow it in words in some body of text. The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. Successor variety stemmers are based on work in structural linguistics, which attempted to determine word and morpheme boundaries based on the distribution of phonemes in a large body of

text. The stemming method based on this work uses letters in place of phonemically transcribed utterances. When this process is carried out using a large body of text, the successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. At this point, the successor variety will sharply increase. This information is used to find the stem.

When the successor varieties for a given word have been derived, the information must be used to segment the word. Hafer and Weiss (1974) discuss four ways of segmenting words:

- Cut off method: Boundary is determined by the cutoff value or threshold value of successor variety. It needs a subjective judgment to determine the cutoff or threshold value.
- Peak and plateau method: A segment break is made after a character whose successor variety exceeds that of the character immediately preceding it and the character immediately following it.
- Complete word method: Break is made if the segment is a complete word in the corpus. It needs to get a huge collection of words with its stem to check the word stem or root. It consumes a lot of computer resources like RAM and secondary storage devices, and needs a huge corpus.
- Entropy method: absolute successor variety value is not reliable because it counts the number of different characters following a given prefix but not how many times they occur. Entropy Method takes

advantage of the distribution of successor variety letters to calculate the cutoff or threshold value.

Out of these four major automatic words segmentation methods in successor variety stemming, peak and plateau, entropy, and complete word methods were used to generate stem. The reason is that these three methods (Peak and Plateau, Entropy and complete word methods) use objective judgments to set segmentation/breaking point.

### 1.4.4 Tools

To build the system, the researcher used JAVA and Python programming languages, because the researcher is familiar with these languages, and the object oriented approach, embedded in these languages, could be best exploited for this research.

### 1.4.5 Evaluation

Comparison was made between the result of the prototype and that of the language experts' judgment, before the submission of the research.

## 1.5 APPLICATION OF RESULTS AND BENEFICIARIES

The result of this research is applicable in Amharic documents text retrieval system development. It helps for document retrieval engine developers to compare and select appropriate stemmer to develop effective and efficient information retrieval systems. In addition, the research output could be used as

an input for other researchers to apply the methodology to other Ethiopian languages.

Practitioners (such as search engine developers, documentation centers, and news agencies, among others) will also be benefited from the outcome of the experiment as to how to apply the principles and methodology of developing a stemmer for document retrieval system.

## 1.6 SCOPE AND LIMITATION OF THE STUDY

### Scope

The scope of this thesis is limited to developing and testing a successor variety stemmer model for Amharic text.

### Limitation

During the course of this research, many internal and external shortcomings were faced. Some of them include: the very limited flat-rate budget in spite of the requirement of the thesis work, lack of computational resources and internet connection, limited cooperation and absence of timely response from linguistic experts.

## 1.7 ORGANIZATION OF THE THESIS

This thesis is organized in six chapters. In chapter one, the general introduction of the problem, the objective and the justification of the study as well as the methodology used for this research is made.

In chapter two, background of the Amharic language, and discussion of the methods used are discussed. Chapter three discusses related works on indexing, stemming in general and successor variety stemming in particular, the overview of the writing system and morphology of Amharic language.

The architecture of the system is described in chapter four. The description of the architecture of the system includes the software components used in preprocessing the input, stemming, and post processing the output.

Experimentation and testing as well as analysis of the result are reported in chapter five. Chapter four also describes the corpus used in the research.

Finally, in chapter six, conclusions and recommendations of further research that have been identified from observation of results of the experimentations are made.

# CHAPTER TWO

# BACKGROUND

## 2.1 INDEXING

### 2.1.1 Types of indexing

By definition, index is a list of important key words from the documents stored in some efficient structure to speed up the searching. Indexing is the process of preparing the index terms. In general, indexing can be classified as manual indexing and automatic indexing.

- **Manual indexing** is a process of preparing an index term manually. Human indexers try to summarize the contents of the whole document in a few keywords. That is, indexers analyze and represent the content of a document through keywords. It is based on intellectual judgment and semantic interpretation of indexers. It requires the indexers prior knowledge of the terms that will be used by the user, indexing vocabulary and collection characteristics to get good keywords or index terms. It is a labor intensive and time taking activity. Therefore, it is costly to apply it for huge collection of documents.

- **Automatic indexing** is the assignment of content identifiers, with the help of modern computing technology. The system extracts significant terms from the original texts of information items to build the index terms. The

human may involve to set the parameters or thresholds, or to choose components or algorithms.

The importance of automatic indexing increases due to enormous amount of information being generated from day to day activities; massive information available in electronic format and on internet, and its cost effectiveness as well as it needs short time to generate index terms when compared with human indexing. According to Salton and McGill (1983), indexing is could be classified as controlled and uncontrolled indexing.

- **Controlled indexing:** is a process of selecting an index term which represents the group of terms and eliminate synonyms. The selected term must be representative of most terms in that class. It also identifies semantically related terms. Controlled terms improve information retrieval performance to select appropriate documents for the given query.

- **Uncontrolled indexing:** is an indexing which considers all variety of vocabulary (in the natural language) within the document. It may cause more ambiguity and error.

It implies that to avoid ambiguity and error using controlled index term is better than that of uncontrolled index term.

## 2.2 STEMMING

Stemming is the process of reducing morphological variants of a word into a common form. According to Manning, Raghavan & Schutze (2008), for morphologically less complex languages like English or Swedish, stemming

usually involves removal of suffixes, but for languages like German and Arabic that have a much richer morphology, stemming process also involves dealing with prefixes, infixes and derivatives in addition to the suffixes. Stemming is widely used in information retrieval, with the assumption that morphological variants represent similar meaning. It is applied during indexing and is used to reduce the vocabulary size, and it is used during query processing in order to ensure similar representation as that of the document collection.

Benno and Martin (2007) stated that most of the words in a text document have various morphological variants. Since the variants have a similar semantics they can be considered as equivalent for the purpose of many retrieval tasks. Consider for example the words “connecting" and “connect" they are not recognized being equivalent without having them reduced to their stem. A stem is the portion of a word that is common to a set of inflected forms; it is not further analyzable into meaningful elements and carries the core meaning of words for which it stands. Stemming is the process of reducing a word to its stem, and a stemmer or a stemming algorithm is a computer program that automates the task of stemming.

### 2.2.1 Conflation Methods

Conflation algorithms are used in Information Retrieval (IR) systems for matching the morphological variants of terms for efficient indexing and faster retrieval operations. The conflation process can be done either manually or automatically. The automatic conflation operation is also called stemming. Frakes (1992) categorizes stemming methods into four groups discussed as follows:

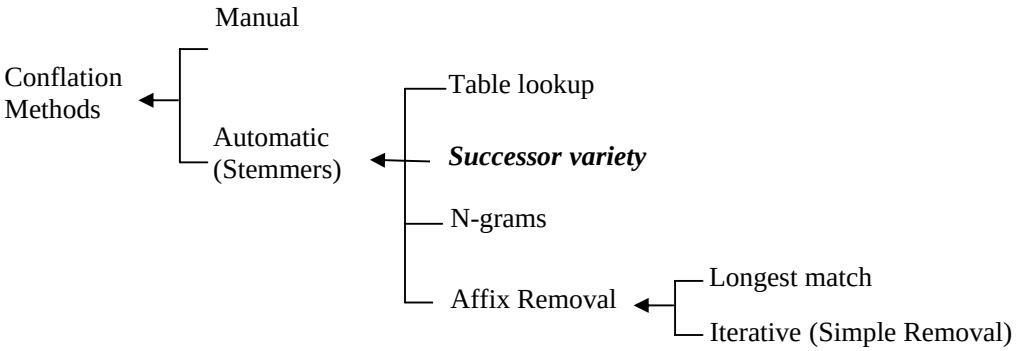


*Figure 1*: Taxonomy for Stemming Algorıtnms

1. **Affix Removal Stemming**: Affix removal algorithms remove suffixes and/or prefixes from terms leaving a stem. It is performed by striping off affixes from each word variant and checking whether any context-sensitive rules apply. One of the most popular examples for affix removal stemming is Porter stemmer. The Porter algorithm consists of a set of condition/action rules. The condition fall into three classes: Conditions on the stem (The measure, denoted m, of a stem is based on its alternate vowel-consonant sequences), Conditions on the suffix (list of suffix or prefix within that domain), and Conditions on rules (The rules are divided into steps. The rules in a step are examined in sequence, and only one rule from a step can apply).

   Affix removal stemmer is language dependent, as an input affix removal stemmer needs an affixes dictionary (suffix and prefix), which is, in most cases, not available. The stemmer requires the language domain knowledge and it needs for extensive manual involvement.

2. **Table Lookup approach to stemming:** Terms from queries and indexes could be stemmed by using table lookup. In general the stemming results are correct. But, using this approach requires creating and storing a

machine readable dictionary/table of all index terms and their corresponding stems. It is difficult to construct the dictionary which incorporates all possibilities of the specific language. It requires multiple dictionary entries for similar concept words which leads the size of the table/dictionary becomes large. So it leads storage and processing time overhead.

3. **N-gram Stemming:** is a string similarity approach to term conflation. It uses similarity measures based on the number of n-grams in common instead of terms, then applying clustering techniques. The main theme of this approach is that the character structure of a word can be used to find semantically similar words and word variants. In this approach, association measures are calculated **between pairs of terms** based on shared unique n-grams.

   - Procedures to compute similarity between two words
     - Create sets of strings of n characters from each word
     - Identify the set of unique n-grams for each word
     - Compare elements of the two sets to find similar n-grams
     - Computing similarity of two terms using Dice's or overlap coefficient
   - Terms that are strongly related by their number of shared n-grams are considered similar and hence conflated or clustered into groups of related words.

To perform the above procedures for generating a cluster or conflate it into the same group requires high computational resource (system and memory intensive).

4. **Successor Variety Stemming:**

According to Hafer and Weiss (1974), the successor variety of a string is the number of different characters that follow it in words in some body of text. The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. Successor variety stemmers are based on work in structural linguistics, which attempted to determine word and morpheme boundaries based on the distribution of phonemes in a large body of text. The stemming method based on this work uses letters in place of phonemically transcribed utterances. When this process is carried out using a large body of text the successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. At this point, the successor variety will sharply increase. This information is used to find the stem.

When the successor varieties for a given word have been derived, the information must be used to segment the word. Hafer and Weiss (1974) discuss four ways of segmenting words:

- *Cut off method*: Boundary is determined by the cutoff value or threshold value of successor variety. It needs a subjective judgment to determine the cutoff or threshold value.

- *Peak and plateau method*: A segment break is made after a character whose successor variety exceeds that of the character immediately preceding it and the character immediately following it.
- *Complete word method*: Break is made if the segment is a complete word in the corpus. It needs to get a huge collection of words with its stem to check the word stem or root. It consumes a lot of computer resources like RAM and secondary storage devices, and needs a huge corpus.
- *Entropy method*: absolute successor variety values are not reliable because it counts the number different characters following a given prefix but not how many times they occur. Entropy Method takes advantage of the distribution of successor variety letters to calculate the cutoff or threshold value.

The successor variety algorithm has the advantage of avoiding the need of affix removal rules that are based on the morphological structure of a language. Therefore, the aim of this research is applying this algorithm to test and analyze the efficiency and effectiveness of this method for Amharic language. The following figure shows the general process model of successor variety stemmer.

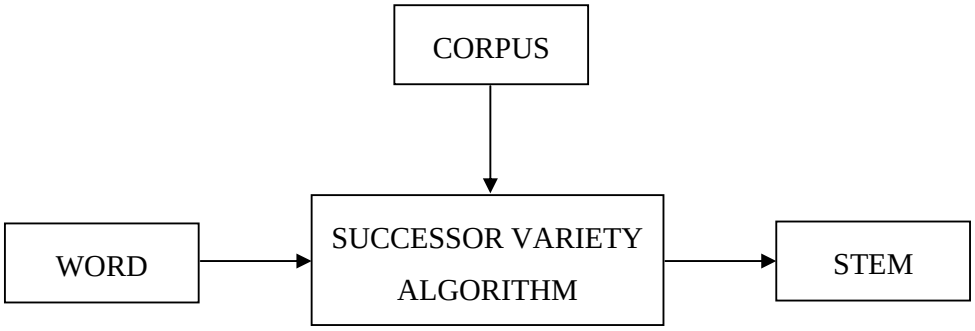


*Figure 2*: The process diagram of Successor Variety Algorithm

## 2.3 OVERVIEW OF THE AMHARIC MORPHOLOGY

### 2.3.1 The Amharic Writing System

Ayele (1994) defined the writing systems as a component of knowledge systems that have philosophical behavior, which the writing systems assist in synthesizing ideas, thoughts, and deeds through the use of signs, symbols or other pictorial renderings. Particularly, writing is a way people record, objectify, and organize their activities and thoughts through images and graphs and a means to inscribe meanings that are expressed through sounds. This means that writing facilitates the proper recording and transmissions of events and deeds from one generation to another.

Lo (1996) defined a writing system as “a set of visible or tactile signs used to represent units of language in a systematic way”. Writing systems are categorized broadly into many types depending on the way of representing their underlying languages. Some of the types are logographic, logophonetic, syllabic, consonantial alphabetic, syllabic alphabetic, and consonant and vowel alphabet. But every script doesn’t fit into single type of writing system rather it shares characteristics that belongs into different classes. Such as the Ethiopic script used by Amharic language is one of such scripts using different types of writing system. It is mostly considered as a syllabic system rather than alphabetic. In the Amharic syllabic writing system, each character stands for a syllable rather than a single sound.

According to National African Language Resource Center and other literatures, the development of the Ethiopic writing system is dated back far beyond the birth of Christ, but no definite time has been mentioned. Some literatures indicate the probable time and others simply stated that it developed in the same time with that of other Semitic writing systems.

Bender et al. (1976) discussed that the present Amharic writing system was adopted form Ge'ez writing system, which belongs to the class of Semitic languages. The earliest known inscriptions in the Ge'ez script date to the 5th century BC. Amharic script which is a successor of Ge'ez and, according Microsoft Encarta, dates back to 300 AD is used for writing in Ethiopia and Eritrea, for languages like Amharic and Tigrigna.

According to Bender et al. (1976), Amharic adopts all Ge'ez alphabet symbols (in Amharic called fidäl (ፊደል) and added some new symbols of its own. Ge'ez is still used especially as a language of liturgy in the Ethiopian Orthodox Church and Church literature.

#### 2.3.1.1 Some thoughts on Ge'ez Language speaking and writing rules on similar sounding multiple alphabets

Ge'ez language was the most widely used written language in the historical Ethiopia and the Ethiopian Orthodox Tewahido Church. We inherited art works, governmental documents and religious scripts, which are widely available in the church and governmental possessions. Finding the primary reasons for the differently written but similarly sounding letters would raise the question how they were used in this ancient Ge'ez language, which is still the liturgy language of the above church. According to my discussion with peoples of specialized knowledge, the most likely main driving factor for the creation of those letters in the Ge'ez language were primarily the nasal, flap/trill, dental, and velar phonemes. In the table below some common words with their attribution to specific alphabets are listed.

*Table 1: Phonemes differentiation or categorization in speaking and writing, the emphasis lies on the underlined characters*

| Word | Phonetical expression |
|---|---|
| ኃ<u>ይ</u>ል | Mid central with higher stress, long |
| ሙ<u>ሃ</u>ኑ <u>ህ</u>ብት | Mid central with moderate stress, short |
| ፀ<u>ሐይ</u> | Mid central with less stress |
| <u>ሠ</u>ራዊት | Dental, palato-alveoral palatal, with more stress |
| <u>እ</u>ሳት | Dental, palato-alveoral palatal, with less stress |

#### 2.3.1.2 Sound similarity within the seven letter groups of the 33 alphabet characters

There are Amharic alphabets sound similarities within the same 33 character groups. They seem to be replaceable to each other. The following table shows these groups of characters.

Table 2: Voice similarity within the seven letter groups of the Amharic characters

| *Number* | *First column (ግዕዝ)* | *Second column (ካብዕ)* | *Third column (ሳድስ)* | *Fourth column (ራብዕ)* | *Fifth column (ሃምስ)* | *Sixth column (ሳድስ)* |
|---|---|---|---|---|---|---|
| 1 | ሀ፣ሐ፣ኀ፣ኸ | | | ሃ፣ሓ፣ኃ፣ኻ | | |
| 2 | አ፣ዐ | | | ኣ፣ዓ | | |
| 3 | | ዉ | | | | ው |
| 4 | | | ፌ | | | ፄ |
| 5 | ፈ | | | ፍ | | |
| 6 | የ | | | | ዬ | |
| 7 | | | ዪ | | | ይ |
| 8 | | | ጂ | | | ጅ |
| 9 | | | ገ | | | ጎ |

The primary applications of the group of letters shown in the above table (in Ge'ez writing) were mainly in the gender (masculine, feminine) conjugation and in singular vs. plural constructs. This seems to be very special to the Ge'ez, and it may readily be interchangeably used in Amharic, since the above discussed grammatical application did not seem to appear in Amharic.

### 2.3.2 Problems with the Amharic writing system

In spite of the easiness of reading Amharic documents, there are some problems of Amharic writing system that have to be resolved. Four of them are discussed in detail as follows.

#### 2.3.2.1 Alphabets with the same sound, but multiple "redundant" characters

According to Getachew (1967) and Bender et al. (1976), the first problem is the presence of different alphabets that share the same sound (redundant characters) and the same sound for the first and fourth order alphabets in the language writing system. One of the interesting special characteristics of the Amharic language is the availability of multiple "redundant" characters for the "same" spoken sound. From the strict Amharic writing and punctuation rule point of view, especially with sound reading and writing background of Ge'ez language, these ambiguities would not surface significantly. As a guideline and common practice there are lead words on which alphabetic symbol to use for the one or the other Amharic word as shown in the table below.

Table 3: Table of Original derivation of words having similar sounding alphabetic symbol, the emphasis is on the underlined alphabets

| Phonetic sound | Letter | Lead word | Example words |
|---|---|---|---|
| hä | ሀ<br>ሐ<br>ኘ | ሃ<u>ሌ</u>ታው ሀ<br><u>ሐ</u>መሩ ሐ<br>ብዙ<u>ኃ</u>ን ኘ | <u>ሃ</u>ይማኖት፦ን፣ <u>ሃ</u>ገር፣ <u>ሃ</u>ብት፣ው<u>ሃ</u>፣ <u>ሃ</u>ቅ<br>በ<u>ሐ</u>ይ፣ <u>ሐ</u>ግ፣ <u>ሐ</u>ጻን፣ ማ<u>ሐ</u>በር፣ <u>ሐ</u>ብረት<br><u>ኃ</u>ብስት፡ አም<u>ኃ</u>፣ <u>ኃ</u>ይል፣ ሆ<u>ኃ</u>ት |
| *s'*ä | በ<br>ጸ | <u>በ</u>ሐዩ በ<br><u>ጸ</u>ሎተ፡ ጸ | <u>በ</u>ላት፣ <u>በ</u>ዋ፣ <u>በ</u>ጥታ፣ <u>በ</u>ሐፊ<br><u>ጽ</u>ጌ፣ <u>ጸ</u>ያፍ፣ <u>ጸ</u>በል፣ <u>ጻ</u>ድቅ፣ <u>ጸ</u>ሎት |
| ʾä | ዐ<br>አ | <u>ዐ</u>ይኑ ዐ<br><u>አ</u>ምዱ አ | <u>ዐ</u>ለም፣ ሠ<u>ዓ</u>ት፣ <u>ዐ</u>ቢይ፣ <u>ዓ</u>ባይ<br><u>እ</u>ሳት፣ <u>አ</u>ዳም፣ <u>አእ</u>ምሮ፣ |
| *s*ä | ሠ<br>ሰ | <u>ሠ</u>ራዊቱ ሠ<br>እ<u>ሳ</u>ቱ ሰ | ንጉ<u>ሥ</u>፣ መ<u>ሠ</u>ረት፣ <u>ሠ</u>ረገላ፣ <u>ሠ</u>ርግ፣<br><u>ሠ</u>ናይ<br><u>ሰ</u>ላም፣ <u>ስ</u>ሜት፣ <u>ስ</u>ም፣ እን<u>ስሳ</u> |

The Amharic language inherited some alphabetic symbols for the same spoken sound from the writing norms and rules of the Ge'ez language. Here, the selection of specific alphabetic symbol is strictly adhered to the word groups and origins and generally should not be used interchangeably.

But in the practical world, especially on the internet, the above mentioned rule seems not to be always respected. Even the lead words are being interchangeably with the one or the other alphabetic symbols. As an example, the different characters such as ኘ, ሀ, ሐ, ኸ; አ, ዐ; ሰ, ሠ and ጸ, በ can be used interchangeably in one word. In addition, ዘ, ዛ; ሔ, ሓ; ኀ, ኃ; ጰ, ጳ; and ዐ, ዓ are also used interchangeably in the same word and for the same meaning. For example, to write the name of the former Ethiopian Emperer Hileselasie one can write his name by using either ዘጀስላሴ, ዛጀስላሴ, ሔጀስላሴ, ሓጀስላሴ, ኀጀስላሴ or ኃጀስላሴ. This example shows only varying the first letter, but there are other varieties by

changing the fourth and/or the six letters as well. This may create confusion for the reader and, if a search algorithm is implemented based on one of the representation; the others may be missed even if they represent the same thing. It is to be seen over time if this linguistic approach of using different letter symbols for the same sounding words is going to fad away and use single letter instead, as is common in other languages. For this research work, it will be of practical necessity to consider both writing scenarios and work on that.

#### 2.3.2.2 Gemination of consonants and context sensitivity of Amharic words

Having gemination of consonants is the second problem of Amharic writing system, which leads a reader to understand the word differently than it is intended; for example the word "ይመታል" which can be read yəmätall 'he hits', or yəmmättall 'he is hit' gives different meaning to the reader . Amharic words are extensively context dependent to reach the level of intended communication and understanding.

*Table 4:* Some examples of words with the same writing but context dependant meaning

| Example | First meaning, with less vocal stress | Second meaning, with strong vocal stress |
|---|---|---|
| ይበላል | (he) will eat | (it) can be eaten |
| ይመታል | (he, it) can hit | (he, it) can be beaten |
| ይሸኛል | (he) can accompany, would need me | (he) will be seen-off |
| ትበላ | (she, it) would eat | (it) can be eaten |
| ብር | Fallow of barley and wheat | Ethiopian currency system, silver |
| አለ | (he) said | There is |
| ይቀራል | (he) may not show up | (meeting, etc.) could be disregarded |

#### 2.3.2.3 Problems in formation of compound words

The third problem is the formation of compound words. This problem seems to be common in several languages. In English language the evolution of compound words begins frequently with a hyphen between the evolving words like “follow-up” or “geo-science”. Such words frequently appear in professional terminologies than in non technical communication in the English language. This practice is even more pronounced in German language where one can use one or more words to get another new word like in “Geshwindigkeitsmessung” for “velocity measurement” According to Bender (1976), compound words in Amharic language are sometimes written as separate words or as a single word. For example, the word ‘school’ can be written as “ትምህርትቤት” or “ትምህርት ቤት”, similarly, ‘blanket’ can be written as “ብርድልብስ” or “ብርድ ልብስ”; ‘Church’ can be written as “ቤተክርስቲያን” or “ቤተ ክርስቲያን” and so on. Any stemming effort has to take this special situation into account.

#### 2.3.2.4 Localized spoken and written variation of identical words, regional dialects

The forth problem of Amharic writing system is different spelling used for the same word such as “ኢትዮጵያ” and “ኢትዮጲያ” to write ‘Ethiopia’, “ኤሌክትሪክ” and “ኤሌትሪክ” which means ‘Electric’. There are variations in writing and speaking of different cities and regions. There are also variations in writing like in አቾነ ፉ አቿነ ፉ ባቄላ በኤላ ጉግና ጅግና ጀግና ሃገ ር ኤገ ር ጥቂት ጥኤት, and so on are very common in Amharic. There is also the sort of single sounding Amharic

letters of the first and the fourth in the Amharic called fidäl (ፊደል). One would write ውሀ ውሃ ዐይን ዓይን for “water” and “eye”.

### 2.3.3 The Alphabet, Punctuation and Numbers in Amharic Language

The Amharic orthography, as it is represented in the Amharic character set, called *fidäl* (ፊደል), consists of 276 distinct symbols. These symbols are classified into four groups. In the first category, according to Bender et al. (1976), there are 33 core characters, each of which occurs in seven orders (one basic and six non basic forms called order), which represent syllable combinations consisting of a consonant and a following vowel. The second category consists of four labialization symbols, which have five orders. The third categories are numbers which are represented by 20 different symbols and finally there are 6 punctuation marks. The following table shows the number of characters in each group.

Table 5: Total Number of Characters in Amharic Alphabet (Fidel)

| No | Type of Amharic Character | Number of Characters |
|---|---|---|
| 1 | Core Characters | 231 |
| 2 | Labialized characters | 20 |
| 3 | Numbers | 20 |
| 4 | Punctuation Marks | 6 |
| | **Total** | **277** |

**The Alphabet**

As mentioned above, the Amharic language consists of a core (basic) of thirty three primary characters or alphabets and six “reproduced” per primary character. It could be roughly assumed that, each fidäl (ፊደል) represents a

collection of consonant and vowel alphabets together. The vowels are fused to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change the order. For example, the first order 'sä' ሠ is transformed into the second order symbol 'su' ሡ by attaching 'ু' to it, also in the same way the fifth order symbol 'si' ሢ is produced by attaching 'ি' to first order symbol etc. This indicates that Amharic doesn't use independent symbols for vowels in representing a syllable, in which its characterization is called syllabic (Bender et al., 1976). However, according to Baye (1997) and Hudson (2001) reported that currently there is a debate about Ethiopic actually syllabic or alphabetic. Hudson (2001) argued that when the system is Alphabetic writing systems the consonants and vowels could present separately like English and Greek; but in Ethiopic the consonant and vowel are differentiated by their attachment symbols. In contrast, Baye (1986) argued that Ethiopic is alphabetic each of the symbols can be broken down into consonant and vowel phonemes which independently represented by a separate symbols. From the above, it can be understood that Ethiopic is a syllabic-alphabetic script for our further investigation purpose.

According to Baye (1997), a set of 38 phones, seven vowels and thirty-one consonants, makes up the complete inventory of sounds for the Amharic language. Amharic consonants are generally classified as stops, fricatives, nasals, liquids, and semi-vowels Leslau (2000). The Amharic vowels are አ, ኡ, ኢ, ኣ, ኤ, እ, ኦ and ዐ, ዑ, ዒ, ዓ, ዔ, ዕ, ዖ. Amharic characters ሀ, ሐ, ኀ, ኸ all representing the /h/ sound, like it is pronounced in "house"; አ, ዐ represent the /a/ sound pronounced in "ant"; ሰ, ሠ represent the /s/ sound pronounced in "sun"; ጸ, ፀ represent the /sss'/ sound which is difficult to pronounce it in English. These

characters are considered as redundant by some scholars who proposed to remove them from Amharic alphabet. To the contrary, there are people who disagree the removal of this characters and they argue that words like ንጉሥ should not be written as ንጉስ or እሳት can not be written as እሣት.

**Amharic Punctuation**

There are a number of symbols for punctuation in Amharic text. Usually word boundaries use either a space like in Roman writing system or they are indicated by two vertically-placed dots like a colon ":" A sentence boundary is indicated by four dots "::" the symbol ' is used as a comma in Roman script, the colon, semicolon and preface colon are represented as ፡ , ፣ and :- respectively. The question mark symbol, three vertically-placed dots ፧, is no more in use and replaced by the Roman Script question mark "?". The modern Amharic writing system adds many foreign systems to its punctuations marks. Some of them are: Quotes are usually in the French style <<...>> and parentheses and exclamation marks are as in the Roman system: (...), !.

**Amharic Number**

Bender et al. (1976) stated that Amharic numbers are derived form Greek letters. And some of them were modified to look like Amharic character. They are represented by single letter and each of them has a horizontal stroke above and below as shown in the following table.

*Table 6:* Amharic numbers

| ፩ | ፪ | ፫ | ፬ | ፭ | ፮ | ፯ | ፰ | ፱ | ፲ | ፳ | ፴ | ፵ | ፶ | ፷ | ፸ | ፹ | ፺ | ፻ | ፼ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 1000 |

The fact that there is no symbol for zero in Amharic script makes it difficult using Amharic numbers for arithmetical computations. Mostly, Ethiopic numbers are used in writing dates and page numbers in text. As a result, most people use the Hindu-Arabic numbers.

### 2.3.4 Amharic Word formations

Written Amharic words are easy to understand for the people who speak the language. There are some words that can be interpreted wrongly just by looking at the words without the context. The meaning of these morphologically similar words depends on the context. For example the word "አለ" *can be read as "alä"* 'he said' or "*allä"* 'there is'; the word "**ይመታ·ል**" *which can be read yəmätall* 'he hits, or will hit some thing, some one', *yəmmättall* 'he is hit, some thing or some body will hit him'. Hadis Alemayehu, the known Ethiopian novelist, tries to resolve this problem in his novel "Fikir Iske Meqabir" **ፍቅር እስከ መቃብር** by placing a dot above the characters, whose consonants were geminated.

According to Baye (2000E.C), in Amharic there is a clear distinction of person, number and gender that plays a role within the grammar of the language. Considering personal pronouns I, she, he, and they in English can be represented as **እኔ** "*ənе"*; **እሷ** "*əsswa"*, **እሱ** "*ə*su" and **እነርሱ**"*ə*nesu".

Amharic distinguishes eight combinations of person, number, and gender. For first person, there is a two-way distinction between singular ('I') and plural ('we'), whereas for second and third persons, there is a distinction between singular and plural and within the singular a further distinction between masculine and feminine ('you masculine singular', 'you feminine singular', 'you plural', 'he', 'she', 'they').

Baye (2000E.C) stated that all Amharic verbs agree with their subjects; that is, the person, number, and ($2^{nd}$ and $3^{rd}$ person singular) gender of the subject of the verb are marked by suffixes or prefixes on the verb.

According to Baye (2000E.C), Amharic has a fairly rich morphology marking. The subject is marked on the verb using subject suffix pronouns, as in ሰበርኩ 'I broke'; the direct object is optionally marked on the verb, as in ሰበረኝ 'he broke me'; some prepositional phrase complements are optionally marked on the verb as in እስኪሰብረኝ 'until he broke me'; functional elements like negation marks, conjunctions and some auxiliary verbs are also bound morphemes and are attached to the verb. For example, in አልሰበርኩም 'I will not be broken', the negation is marked by አል. Amharic verbs often have additional morphology that indicates the person, number, and ($2^{nd}$ and $3^{rd}$ person singular) gender of the object of the verb.

### 2.3.5 The Morphology of Amharic Language

Amharic language creates inflectional and derivational word forms by using both prefixing and suffixing.

Baye (2000E.C) explained that inflection morphemes of Amharic words derived from nouns by using gender, number and case marker suffixes. The following table clarifies this idea by using example.

*Table 7:* Gender, number and case marker suffixes

| Word | Gender Marker | | Number | | Case | |
|---|---|---|---|---|---|---|
| | **Masculine** | **Feminine** | **Singular** | **Plural** | **Nominative** | **Accusative** |
| ድመት 'dmet' (cat) | ድመት | ድመት-ኢት | ድመት | ድመት-ኦች | ድመት | ድመት-ን |
| ልጅ 'lj' (child) | ልጅ | ልጅ-ኢት | ልጅ | ልጅ-ኦች | ልጅ | ልጅ-ን |
| ጥጃ 'Tja' (calf) | ጥጃ | ጥጃ-ኢት | ጥጃ | ጥጃ-ኦች | ጥጃ | ጥጃ-ን |

(Source: የአማርኛ ሰዋሰው - የተሻሻለ ሁለተኛ እትም Baye (2000E.C) )

Derivational nouns are created by adding prefixes, infixes or suffixes to basic nouns, adjectives, verbs, stems and roots.

Moreover, Amharic conjunction words are written in different forms sometimes it attaches with the other word or it stands alone. For example '**አበበና ከበደ**' or '**አበበ እና ከበደ**', the meanings of both phrases are the same "Abebe and Kebede". It indicates that "and" is written as '**ና**' attached with other words or '**እና**' as a stand alone word. In written and spoken applications, the "omission" of እ from such conjunction is common and would be used interchangeably.

# CHAPTER THREE

# REVIEW OF RELATED LITERATURES

## 3.1 INTRODUCTION

Since 1940s the problem of information storage and retrieval attracted attention at alarming rate. Getting appropriate and relevant documents quickly from vast amount of unstructured stored information is tedious and time consuming. Historically, before the dawn of the computer era, most of the documents were gathered and managed in hardcopy format, preventing any means of access except the physical one. Those days retrieving relevant documents electronically were given little attention. Rijsbergen (1999) stated that the consequence were duplication of effort that led to wastage of time and capital. However, the introduction of price wise affordable Personal Computers and the subsequent wide implementation of Networking and World Wide Web Technologies have revolutionized every aspect of information retrieval, and handling as it were merely indexing. But also, especially the World Wide Web served as intelligent retrieval system that becoming *"a universal repository of human knowledge and culture which allowed unprecedented sharing of idea and information in a scale never seen before"* (Beaza-Yates and Ribeiro-Neto: 1999: 2) (Beaza-Yates and Ribeiro-Neto, 1999; and Rijsbergen, 1999).

Besides, manually done information storage and retrieval mechanism such as cataloging and general administration transformed to computer based work. But,

the problem of having effective searching remains unsolved. Such problems triggered and necessitate the extraction of relevant document from the whole documents. In addition, the natural language interface enables the user to pass the need of detailed communication protocols, machine location and operation system. As a result, users started demanding an effective and relevant documents with respect to their request or query with little effort and at almost no cost (Baeza-Yates and Ribeiro-Neto, 1999 and Rijsbergen, 1999).

Most modern information retrieval (IR) systems implement some sort of what is commonly known as “stemming”. A system using stemming conflates derived-word forms to a common stem. The benefits of such a procedure is two-fold: by conflating several forms to the same stem, the number of entries and the size of the search index are reduced. More importantly, stemming is potentially helpful for free text retrieval, where search terms can occur in various different forms in the document collection. Stemming makes retrieval of such documents independent from the specific word form used in the query. In this experiment, concentration is given for the impact of stemming on retrieval effectiveness.

According to Martin (2003), the main reason for the use of stemming is the hope that through the increased number of matches between search terms and documents, the quality of search results is improved. In terms of the most popular measures for determining retrieval effectiveness, precision (amount of relevant documents retrieved compared to all documents retrieved) and recall (amount of relevant documents retrieved compared to total number of relevant documents in the collection), stemmed terms retrieve additional relevant documents that would have otherwise gone undetected, i.e. they improve recall. There is also potential

for improved precision, since additional term matches can contribute to a better weighting for a query/document pair.

Manning, Raghavan & Schutze (2008) stated that stemming has a wide range of applications in different fields, such as natural language processing, automated text processing (indexing), speech synthesis and recognition, machine translation, handwriting recognition, grammar checking, and sentence generation. However, the principal use of stemmers is for information retrieval purposes. One of the main problems involved in information retrieval is variations in word forms. The most common types of variations are spelling errors, alternative spellings, multi-word constructions, transliteration, affixes, and abbreviations. One way to avoid such problems is to use stemming. Information retrieval systems use stemming to improve the matching algorithms. Since this research emphasis is automatic stemming by using successor variety stemmer, indexing will further be discussed in the next sub-chapter.

## 3.2 REVIEW OF RELATED WORKS

Many researches have been conducted on stemming different languages using different approaches. Among them are: Porter (1980) for English using affix removal approach; Alemayehu and Willett (2002) for Amharic using affix removal approach; Lemma (2003) for Wolayta using affix removal iterative approach; Wakshum (2000) for Afaan Oromoo using affix removal approach; Betelihem (2002) for Amharic using n-gram method; Al-shalabi et al (2005) for Arabic using successor variety approach; and recently Atelach and Lars (2007) for Amharic

using table lookup approach. This section covers a review of some automatic indexing researches conducted so far using different methods.

Alemayehu and Willett (Alemayehu and Willett, 2002) developed the stemming of Amharic words for information retrieval and presented an implementation of an iterative stemmer for removal of both prefixes and suffixes. To implement the algorithm it needs stop word and affix lists. They used a two step process for the preparation of stop-word list. First automatically generate high-frequency words as potential stop-word lists. Then they selected and prepared stop word lists manually from the potential stop word lists. Similarly the affix lists were prepared by extensive manual involvement.

In their research, the researchers used 1221 different sample words to test the accuracy of the stemming process. They reported that 95.9% of words were stemmed appropriately/correctly when they assessed manually. Out of 1221 different words given to the stemmer it produced 607 different stem or root words. The result shows that the stemmer compresses the given sample by 50.3 percent. This research work was done by applying the first conflation (affix removal) method mentioned above.

Another prototype automatic indexing for Amharic text was developed by Bethlehem Mengistu (Bethlehem 2002) where she used N-gram based approach to calculate similarity and cluster similar words into groups and represent the group by one stem or root term. She developed the system by assigning n-values by bi-grams and tri-grams. The researcher also used word based indexing to compare the precision and recall with the result of the n-gram approaches. 100 documents and 24 queries were used for testing the system. The comparison of

the result was made and reported that the word based indexing had more effective than n-gram based retrieval. However, the researcher concluded that still the use of bi-grams and tri-grams have comparable performance with word based indexing. The researcher reported that the average precision observed was 3.4%, 5.3% and 14.2% by using bi-grams, tri-grams and word-based approach respectively. When the researcher used the recall to measure the performance of the techniques she got 94.2%, 88.6% and 44.6% for bi-grams, tri-grams and word-based approach respectively.

Recently, Atelach and Lars (2007) did a work that an Amharic stemmer which reduces words to their citation forms by implementing a dictionary lookup. The researchers applied a rule based approach supplemented by occurrence statistics of words in a machine readable dictionary. Their corpus size was 3.1 million words; it was composed from news articles and classic fiction text. They reported that the stemmer accuracy for the old fashioned fiction text was 60% and for the news articles 75%.

Al-Shalabi *et al.* (2005) tested the application of the Successor Variety Stemming Algorithm technique with the Cutoff Method and with Entropy Method for Arabic language. The researchers applied the algorithm to 2000 Arabic words. Their report explained that the successor variety stemming algorithm technique by using the cutoff method achieved 80% level of correctness. In contrast, 75% level of accuracy was reported when they used successor variety stemming algorithm technique with Entropy method. They compared the result of the Successor Variety Stemming Algorithm technique with the Cutoff Method and the Successor Variety Stemming Algorithm technique with Entropy Method. The researchers

concluded that result of their research shows that the successor variety stemming algorithm technique with the cutoff method showed better performance.

Despite the fact that they have their own weaknesses and limitations, there have been a lot of research attempts and encouraging achievements that were gained in relation stemming. Some of these works demand affix lists and affix removal rules like affix removal stemmer, some of them require machine readable full fledged dictionary, like table look and others require availability of high capacity resources like n-gram stemming.

In order to benefit the best out of these research works, exploring a stemming algorithm that can best fit to the Amharic corpus is crucial. There is still too much work left on this part. The main focus of this research is on exploring the potential of successor variety algorithm to conflate Amharic terms in to their root (stem).

Successor variety algorithm reduces the need of exhaustive listing of affixes and affix removal rules. Up to the knowledge of the researcher, there is no previously conducted research in Amharic language by applying successor variety method. Therefore, the researcher has an interest to apply the successor variety methods for Amharic news and test the performance of its result.

# CHAPTER FOUR

# DESIGN AND DEVELOPMENT OF THE SYSTEM

## 4.1 INTRODUCTION

In this chapter, the design and development of the prototype conflation system by using successor variety algorithm is discussed. The architecture of the system and the description of the developed as well as the adopted subcomponents of the systems are also discussed in this chapter.

## 4.2 THE ARCHITECTURE OF THE PROTOTYPE

The following figure shows the architecture of the Amharic Successor Variety Stemmer. In the diagram showing the architecture arrows indicate the flow of data, the boxes show the program and the multi-document symbols show the files containing the inputs and the outputs of the software components of the system. Section 4.3 describes the subcomponents of the system in detail.

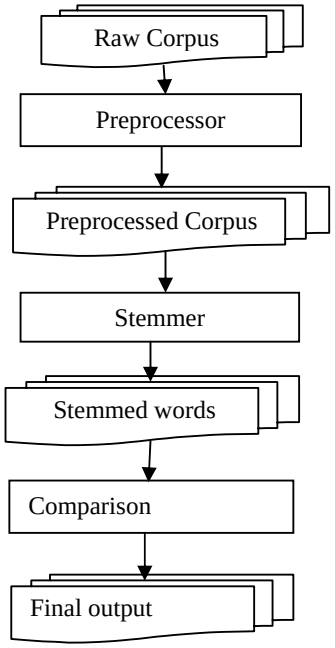


Figure 3: The Architecture of the prototype

## 4.3 DESCRIPTION OF SOFTWARE SYSTEMS

The developed prototype successor variety stemmer system has three subsystems, each of which has one or more software component to accomplish the intended task. These subsystems are: document preprocessor, successor variety stemmer, and post processor subsystem.

### 4.3.1 Document Preprocessor Subsystem

The document acquired had substantial level of “impurity” and preprocessing prior to its further use to the actual stemmer. Document preprocessing is the preparation of the text for further processes by treating digits, hyphens, punctuation marks, and the case of letters in the given document. For example, digits without any combination of words may not use for indexing; therefore it must be removed from the document. In addition, it removes extraneous

characters from the text that do not contribute to the content description and that may facilitate processing of the text. Examples of extraneous characters are punctuation marks, control characters (line feed, carriage return, tab).

In most of Amharic fonts, including the visual Ge'ez font, both upper case and lower case letters of the same alphabet are used to represent two different symbols (order of Amharic fidäl like 'h' represents the first order "ሀ" and 'ሐ' represents the sixth order of the same fidel "ሀ"). As a result case conversion was not needed and has not been done as part of text preprocessing.

In Amharic fonts, except Unicode supported ones, characters which have diacritic markings need more than one byte in their internal representation so as to use one byte for the basic character and additional one byte for the diacritic marking. For example, "ሉ" needs two bytes to represent internally, one byte for "ሀ" and another byte for "ሉ". This creates difficulty to use Amharic alphabet/ fidäl that has diacritic markings by considering it as a single unit. For this reason, the document was first changed in to Unicode representation and then transliterated it into American Standard Code for Information Interchange (ASCII) letters.

#### 4.3.1.1 Document Cleaning Subcomponent

The acquired document contains numbers, punctuation marks and control characters in the text of each document. These characters do not contribute to content description of the document. To remove these characters from the document, a program using python programming language, was developed. This program accepts the file containing the raw document, cleans it, and writes the cleaned text into a file as needed.

**Removal of Irrelevant Characters**

Numbers, punctuation marks and control characters in the text of each document were removed under the assumption that they do not contribute to content description of the document. Among the different characters, slash '/' was handled in two different ways. The first one is if it appears at the beginning and at the end of the word it might be used as a bracket. Therefore, it was removed form the content of the file. For example, "ከንቲባ አርከበ "ችግር የሚፈቱ ቁልፍ የልማት ተግባራትን ለማከናወን ከወዲሁ ትኩረት ሠጥቶ የገቢ አቅምን ለማዳበር በእቅድና በጥናት ላይ ተመስርቶ ሊሠራ ይገባዋል" ሲሉ ገልጸዋል /ተናግረዋል/::" In this example the slash '/' is used as a bracket, it is eliminated at the preprocessing time. On the other hand, if it appears between letters within a single word it may have more than one meaning. Such as the abbreviation 'አ/አ' may be converted in to 'አዲስ አበባ', 'አገር አቀፍ' or 'አለም አቀፍ'; and 'ኃ/ማርያም' may be converted into 'ኃይለ ማርያም', or 'ኃብተ ማርያም'. It creates a difficulty to convert the abbreviations into its expanded form. Therefore, the abbreviation was used as it is. Other punctuation marks like hyphens, commas were handled here by applying the developed preprocessor program.

#### 4.3.1.2 Characters Normalization Subcomponent

After cleaning the document, there is yet another problem that needs to be fixed. That is, in Amharic writing system, there are different symbols having the same sound. For example, ኃላፊ, ሀላፊ, and ሃላፊ have the same meaning, as discussed in 2.4.2.1. The change of the alphabet into one common representation does not cause a meaning difference, but it increases getting the same term with different characters. Even though this would violate the linguistic rules and the norms of

the Amharic language in general, it will increase the success rate of getting similar or related words in an online indexing and searching process. Unlike spell checkers and dictionaries, the amalgamation and concatenation of similar sound Amharic alphabets in corpus construction will not adversely impact the wider applicability of the end result. It is not the objective of the stemming routine if the word shall be written as **ጸሃይ, ፀሃይ, ጸሐይ, ፀሐይ, ፀሓይ, ፀኀይ** or **ጸኀይ.** But by looking all these potential combination of Amharic letters to build the word "sun" the potential success rate in indexing and searching speed will increase substantially.

Therefore, for this research the different symbols with the same sound are considered as equivalent and changed in to one common representation. For example 'ሀ', 'ሃ', 'ኅ', 'ኃ', 'ሐ', 'ሓ' and 'ኻ' are converted into 'ሀ'; 'ሰ' and 'ሠ' are converted into 'ሰ'. In addition all non basic orders of the alphabet and other similar sound alphabets are converted in to their respective common representatives. The detailed conversion equivalences of the alphabets are shown at Appendix 1.

To normalize the Amharic characters having the same sound into one common representation, a program was written using the Python programming language. This program accepts the cleaned file, maps the characters into one common character, and writes the normalized text into a file.

#### 4.3.1.3 Transliteration Subcomponent

To use the successor variety stemmer, transliteration of the normalized text is required. The cleaned and normalized text is used as an input for the transliteration system. To transliterate the Amharic document so that the Amharic alphabet can be represented by ASCII letters, a python program from Ermias Abebe was adopted. This program accepts the cleaned and normalize file, maps the Amharic characters into ASCII letters using the mapping table, and writes the transliterated text into a file.

In order to simplify the analysis, after removing the irrelevant characters and normalizing the characters, all the Amharic texts were transliterated into the Ethiopic Unicode definition in SERA (Firdyiwek and Yacob, 1997) by using the transliteration program discussed above.

#### 4.3.1.4 Tokenization

According to Grefenstette and Tapanainen (1994), *Lexical analysis or tokenization* is the process of converting an input stream of characters into a stream of words or tokens. Tokenization is the first stage of automatic indexing, and of query processing. A token or word is defined as a string of characters separated by white space and/or punctuation marks. *Tokens* are groups of characters with collective significance. Tokenization involves the chopping of the document into words or tokens by considering different criteria. Numbers, punctuation marks and other impurities that are not removed by document cleaning subcomponent are considered in this step.

By applying the following algorithm the input text file was changed into tokens, that is an input for the experimentation of the research work. This algorithm was implemented by python.

Algorithm to identify a token

1. *Read an input file*
2. *Read a character from the input text up to white space character and put it in to a variable called 'test'*
3. *Check the test content:*
   *if the length of word in the test variable is less than 3,*
   *goto step 2*
   *else if test contains '/'*
   *goto step 2*
   *else If it is already in the token goto step 2*
   *Else append the word in to the token list*
4. *If end of text input, write the list into a file and Exit.*
5. *Else goto step 2.*

The tokenizer accepts the transliterated file, tokenizes it, and writes the tokens into a file.

Before tokenizing the transliterated file, abbreviations that were not handled at the cleaning stage and other short words (terms having less than three characters) were eliminated by the tokenization program. The reason for eliminating them is the assumption, that abbreviations as well as some very short

words would not be considered as index terms. After eliminating these “non-useful” words from the corpus, the tokenizer breaks the corpus into list of words that are ready to be incorporated into the stemmer.

### 4.3.2 Successor Variety Stemmer

The algorithm that was implemented to generate the successor variety and the stem for the given word are described as follows.

#### 4.3.2.1 Successor variety generation based on Peak and Plateau Algorithm

The following algorithm is adopted from Hafer and Weiss (1974) and modified based on Successor Variety Peak and Plateau Method

Step 1: To determine the successor variety of the word, read the word from the corpus and call it ‘PeakWord’

*Starting from i=1to the length of PeakWord:*
For the rightmost i letters in PeakWord:
i. Set a user defined variable list called PeakSuccList which has two parts (String and integer)
ii. Count the number of letters in the corpus that follow the first ith right most letters of PeakWord.
iii. Append the successor substring from the above step into PeakSuccList.word and different number of characters into PeakSuccList.frequency.

Step 2: To Segment the word by using the peak and plateau method:
i. Set the variable named PeakSegment list, with type String

```
ii. For each substring at position j in the PeakSuccList
    If (PeakSuccList[j-1].frequency <
        PeakSuccList[j].frequency &&
        PeakSuccList[j].frequency >
        PeakSuccList[j+1].frequency && PeakCutValue <
        peakSuccList[j].frequency)
            PeakCutValue=peakSuccList[j].frequency
            temp= peakSuccList[j].word
        j=j+1
iii.  If (PeakCutValue!=0)
        PeakSegment.addElement(temp)
        PeakSegment.addElement(PeakWord(temp.length()))
    else
      PeakSegment.addElement(PeakWord)
iv.   if the content of the corpus ends
      Goto step 3
       else
      Goto step 1
```

Step 3: To Select the segment as a stem by using the peak and plateau method:

```
i. Set a variable named PeakStem
ii.   For each segment in PeakSegment List
    for i starting from 1 up to the length of
    PeakSegment list
    count=0
    temp=PeakSegment[i]
     for j starting from 1 to length of PeakSegment list
          if(temp==peakSegment[i])
               count=count+1
               if(count<=7)
          PeakStem.addElement(Temp)
iii. If the selection of the stem ends write the
    content of the PeakStem content into
    PeakPlateauStem file
Exit the process.
```

The first step is giving the word to the stemmer. For example the word "lemetenten" (**ለመተንተን**) is given to the stemmer as an input to be stemmed. After reading the input word, the program opens the corpus file. Then, it generates the sub list that starts with the same letter with the test word from the corpus. After that the prefix and its successor letter counts are generated. The following table shows this step.

*Table 8: Sample Successor Variety generated for the word "lemetenten"*

| Prefix | Successor Varity (Count of Different successor letters) |
|---|---|
| l | 14 |
| le | 25 |
| lem | 7 |
| leme | 18 |
| lemet | 2 |
| lemete | 2 |
| lemeten | 2 |
| lemetent | 1 |
| lemetente | 1 |
| lemetenten | 1 |

After the generation of the prefix and different successor character counts, the next step is deciding the peak and plateau cutoff point. If the count of different successor character value of the given prefix is greater than the previous and the next prefix count of different successor character, that count value is considered as a candidate peak and plateau cutoff value. From the above table, 25 (14 < 25 and 25 > 7) and 18 (7 < 18 and 18 > 2) satisfy the condition. Therefore, both values are candidates for cutoff value. For this system, if the result of the

comparison is only one value, that value is taken as cutoff value. If there is more than one value available, the biggest value is taken as cutoff value.

In the next step, the given word is segmented by using the result peak and plateau cutoff value. If the cutoff value is zero (i.e., there is no value that satisfies the condition) the word itself is taken as a segment. For the above example the peak and plateau cutoff value is 25. Therefore, the word “lemetenten” (**ለመተንተን**) is segmented into “le” (**ለ**) and “metenten” (**መተንተን**) and stored into the peak and plateau segment file.

Finally, depending on the frequency of occurrence of the segment in the peak and plateau segment file, one of the segments is taken as a stem. When the frequency of the segment is less than the fixed value (i.e., 8 for this particular research), it is taken as the stem of the word. For the above example, the occurrence of “metenten” (**መተንተን**) is less than eight and it is taken as a stem for “lemetenten” (**ለመተንተን**).

#### 4.3.2.2 Successor variety generation based on Entropy Algorithm

The algorithm of Successor Variety Entropy method is shown as follows

```
Step one: Determine the successor varieties for a word. Take
          a word from the corpus and name it EntWord:
     Starting from i=1to the length of EntWord:
      i.  set a user defined variable list called
        EntropyList which has two parts (string , float)
      ii. To calculate the Entropy cutoff value
           For the rightmost i letters in EntWord:
```

$\alpha$ = i length of substring form EntWord

c = character at i+1

$D_{\alpha i}$ = list of words in the corpus which starts with $\alpha$

$|D_{\alpha i}|$ = number of words in the corpus which starts with $\alpha$

$|D_{\alpha i}|$= number of words in $D_{\alpha i}$ starts with $\alpha$ & char at i+1 is c

iii. To calculate the Entropy value for *successor j by using*

$$H = \sum_{j=1}^{j='\max'} -\frac{|D\alpha ij|}{|D\alpha i|} \times \log_2^{|D\alpha ij|/|D\alpha i|}$$

*Store the result in to EntropyList.Word*=$\alpha$ and EntropyList.frequency=H*.*

Step 2: To Segment the word by using the entropy method:

i. Set the variable named EntropySegment list, with type String

*ii.* For j starting from 1 the length of *EntropyList*

*If (EntropyList[j-1].frequency < EntropyList[j].frequency && EntropyCutValue < EntropyList[j].frequency)*

*EntropyCutValue=EntropyList[j].frequency*

*temp= EntropyList[j].word*

*j=j+1*

*iii. If (EntropyCutValue!=0)*

*EntropySegment.addElement(temp)*

*EntropySegment.addElement(EntWord(temp.length()))*

*else*

*EntropySegment.addElement(EntWord)*

*iv. if the content of the corpus ends*

*Goto step 3*

*else*

```
          Goto step 1
Step 3: To Select the segment as a stem by using the Entropy
        method:
     i. Set a variable named EntropyStem
     ii.   For each segment in EntropySegment List
         for i starting from 1 up to the length of
         EntropySegment list
         count=0
         temp=EntropySegment[i]
                 for j starting from 1 to length of
          EntropySegment list
               if(temp==EntropySegment[i])
                    count=count+1
                    if(count<=7)
               EnrtopyStem.addElement(Temp)
     iii. If the selection of the stem ends write the
        content of the EntropyStem content into EntropyStem
        file
    Exit the process
```

The first step is giving the word to the stemmer. For example the word "Cewatacewn" (ፈዉ፟ታ፟ቸ፟ው፟ን) is given for the stemmer as an input to be stemmed. After reading the input word, the program opens the corpus file. Then, it generates the sub list that starts with the same letter with the test word from the corpus. After that the i length prefix of the word, different successor characters that follows the given prefix at i+1 position and the frequency of occurrence of each successor character at the i+1 position of the given word with in the subset corpus is stored into a single file. In addition, the prefix and the number of words that begins with the given prefix are stored into another file. The following table

depicts sample of the successor variety generation for the test word 'Cewatacewn' from the input to stem list file.

*Table 9:* Sample Successor Variety generated and the successor letters with their frequency of occurrence in the corpus for the word "Cewatacewn"

| **Prefix** | **Successor variety letters with their frequency of occurrence in the corpus ($D_{\alpha ij}$)** |
|---|---|
| C | $e^8$, $k^2$, $l^2$, $n^2$, $q^2$, $u^3$ |
| Ce | $l^2$, $m^3$, $q^2$, $w^5$ |
| Cew | $a^5$ |
| Cewa | $t^5$ |
| Cewat | $a^5$ |
| Cewata | $\$^2$, $c^2$, $w^3$ |
| Cewatac | $e^2$ |
| Cewatace | $w^2$ |
| Cewatacew | $n^2$ |
| Cewatacewn | $\$^2$ |

*Table 10:* Sample the prefix of the word and its number of words in a text body beginning with the i length sequence of letters in the corpus ($D_{\alpha i}$)

| Prefix | Number of words beginning with the given prefix ($D_{\alpha i}$) |
|---|---|
| C | 19 |
| Ce | 12 |
| Cew | 5 |
| Cewa | 5 |
| Cewat | 5 |
| Cewata | 7 |
| Cewatac | 2 |
| Cewatace | 2 |
| Cewatacew | 2 |
| Cewatacewn | 2 |

When these two tables are available the calculation of the entropy value is done by applying the Entropy formula and generates the successor variety table which contains the prefix of the word along with the entropy value of each prefix.

$$H\alpha i = \sum_{j=1}^{j='\max'} -\frac{|\,D\alpha ij\,|}{|\,D\alpha i\,|} \times \log_2^{|D\alpha ij|/|D\alpha i|}$$

The sample calculation and the result table are shown as follows.

Number of $D_{\alpha i}$ values are available at table 3.3, frequency of words that contain prefix $\alpha$ with successor character j ($|D_{\alpha ij}|$) is available in table 3.2

For i=1, $\alpha$= 'C',

Entropy value of 'C' = (- 8\19 * $\log_2$(8\19)) + 3(-2/19 * $\log_2$(2/19)) +(- 3\19 * $\log_2$(3\19))

= 2.31346

For i=2, $\alpha$= 'Ce'

Entropy value of 'Ce' = 2(-2\12 * $\log_2$(2\12)) + (-3/12 * $\log_2$(3/12)) +(-5\12 * $\log_2$(5\12))

= 1,88792

For i=3, $\alpha$= 'Cew'; i=4, $\alpha$= 'Cewa'; i=5, $\alpha$= 'Cewat'; i=7, $\alpha$= 'Cewatac'; i=8, $\alpha$= 'Cewatace'; i=9, $\alpha$= 'Cewatacew'; and i=10, $\alpha$= 'Cewatacewn' have a single successor letter j, therefore the results are similar.

Entropy value of 'Cew' = (-5\5 * $\log_2$(5\5))

= 0.0

For i=6, $\alpha$= 'Cewata'

Entropy value of 'Ce' = 2(-2\7 * $\log_2$(2\7)) + (-3/7 * $\log_2$(3/7))

= 1.55666

*Table 11:* Sample Successor Variety generated with the calculated Entropy value for the word “Cewatacewn”

| Prefix | The Entropy value of the prefix |
|---|---|
| C | 2.31346 |
| Ce | 1.88792 |
| Cew | 0.0 |
| Cewa | 0.0 |
| Cewat | 0.0 |
| Cewata | 1.55666 |
| Cewatac | 0.0 |
| Cewatace | 0.0 |
| Cewatacew | 0.0 |
| Cewatacewn | 0.0 |

To determine the cutoff value of the successor variety entropy method, the system starts to compare the entropy value of each prefix with its previous prefix entropy value, if the result is greater than the previous prefix entropy value it assigns that value as an entropy cutoff value. For the above example the entropy value of the prefix ‘Cewata’ (1.55666) is greater than ‘Cewat’ entropy value (0.0). Therefore, the entropy cutoff value is 1.55666.

The next step is segmenting the word. If the entropy cutoff value is zero then the word is not segmented. Otherwise the word is segmented at the point the entropy cutoff value is equal to its entropy value of the given prefix. For the above example, the entropy cutoff value 1.55666 is equal to that of entropy value of the prefix ‘Cewata’, therefore, the word ‘Cewatacewn’ (ፈጨዋታችውን) is segmented into “Cewata” (ፈጨዋታ) and “cewn” (ችውን).

Finally, the stem of the word is selected based on the frequency of occurrence of the segments in the entropy segment list. For this example the segment “Cewata”

(ፊጨዋታ) appears in the entropy segment list less than eight times, therefore it is taken as a stem for 'Cewatacewn' (ፊጨዋታችውን) word.

### 4.3.3 Testing Method

The performance of the system output was measured using a program developed with the Python language for this thesis. Algorithm of the testing program is shown as follows:

```
1. Read ManualStem file
   Set the variable named ManStemList as type string
   for i =0 to ManualStem.length
     temp=ManualStem[i]
          for j=0 to ManStemList.length
       if(temp==ManStemList[j])
          break
     else
          ManStemList.append(temp)
2. Read PeakPlateauStem file
   Set the variable named PeakStemList as type string
   for i =0 to PeakPlateauStem.length
     temp=PeakPlateauStem[i]
          for j=0 to PeakStemList.length
       if(temp== PeakStemList [j])
          break
     else
          PeakStemList.append(temp)
3. Read EntropyStem file and assign it for EntropyStemList
   Set the variable named EntropyStemList as type string
   for i =0 to EntropyStem.length
     temp=EntropyStem[i]
          for j=0 to EntropyStemList.length
       if(temp== EntropyStemList [j])
```

```
          break
    else
          EntropyStemList.append(temp)
```

4. To culculate the performance of successor variety peak and plateau method

```
count=0

for i =0 to PeakStemList.length
temp=PeakStemList[i]
  for j =0 to ManStemList.length
      if(temp==ManStemList[j])
       count=count+1
     else: break
PeakPerformance= (count/ManStemList.Length)*100%
```

5. To culculate the performance of successor variety Entropy method

```
count=0
for i =0 to EntropyStemList.length
temp=EntropyStemList[i]
  for j =0 to ManStemList.length
      if(temp==ManStemList[j])
       count=count+1
     else: break
EntropyPerformance= (count/ManStemList.Length)*100%
```

# CHAPTER FIVE

# EXPERIMENTATION AND ANALYSIS

## 5.1 INTRODUCTION

In this chapter, the experiment conducted by using the prototype and the findings from the experiment were reported. In addition, the corpus, training and testing of the system were discussed.

## 5.2 THE CORPUS

For the purpose of this research, corpus was obtained from Walta Information Center in electronic form. The size of the total corpus is 6270 words. Since the corpus is news items, the domain of the corpus is so diversified that it includes topics like political, economic, sport and religion. A sample of the text documents along with its transliterated equivalents are attached in appendix 2.

### The Training Set

The total corpus was divided in to two parts as a twenty-eighty proportion randomly. Eighty percent of the corpuses, i.e., 5016 words were used to train the stemmer.

### The Test Set

After taking the training data from the corpus, the remaining twenty percent of the total corpus, i.e., 1254 words were used as a test data to measure the performance of the prototype system.

## 5.3 TRAINING THE SYSTEM

The initially developed system was trained by using the training set and adjusted continuously to improve and get a better performance. Some of the modifications which resulted in significant improvement in the performance of the system are discussed as follows:

- In many words, the number of occurrence of the first character with different successor characters are high, it leads the system to get a wrong segment point for both Peak and plateau, and Entropy cutoff points. The researcher solved this problem by ignoring the first character frequency. Because, the researcher believes that there is no (may rarely occur) single-character stem as discussed in the preceding chapter on tokenization.
- In successor variety Peak and Plateau method multiple pick points were found within a single word. For example, the word “bemegenaNa” (**በመገናኛ**) has four pick points that satisfy the condition of peak and plateau algorithm cutoff point: after “be”, “beme”, “bemegen” and “bemegenaN” as shown bellow.

| Prefix | Number of different characters after the given prefix |
|---|---|
| b | 21 |
| *be* | *32* |
| bem | 15 |
| *beme* | *20* |
| bemeg | 6 |
| bemege | 2 |
| *bemegen* | *3* |
| bemegena | 1 |
| *bemegenaN* | *2* |
| bemegenaNa | 1 |

But except after “be” (which is the correct cutoff point), the other pick points lead the system to wrong cutoff point. Therefore, the system was adjusted to assign the biggest pick point among the candidates for the peak and plateau cutoff value. This adjustment gives a good performance improvement for the system.

- The input corpus for this research work incorporates a diversity of topics. It makes some difficulty to apply the rule of selecting the stem when the segment occurs less than or equal to twelve times as suggested by Hafer and Weiss (1974). For this research (because of the limited corpus size and domain diversity), the occurrence of the segment was adjusted to five times to get the right stem, by observing progressively.

## 5.4 TESTING THE SYSTEM

After training and making some adjustments to the developed system by using the training set, the final program was tested for its performance by using the test set. The analysis of the output of the program on the test set is discussed as follows.

The system is run on the testing set after being trained on the training set. The system, after reading words from a file containing the test data, produces the stem that was judged to be correct using peak and plateau, entropy and complete word methods. After generating the stem of the test words and getting the stems by using peak and plateau, entropy and complete word methods, the output of the system was compared with manually stemmed results automatically.

The stemmer was tested on 1254 words (20%) selected as a test set from the total corpus for each of the three methods (peak and plateau, entropy and complete word methods) and the following results were found:

- Peak and plateau method properly stemmed 71.85% (901 words) of the total 1254 words. Out of the 28.45% wrongly stemmed words, 20.49% (257 words) were under-stemmed and the remaining 7.66% (96 words) were over- stemmed.
- Entropy method properly stemmed 63.95% (802 words) of the total 1254 words. Out of the 36.05% wrongly stemmed words, 18.66% (234 words) were under-stemmed and the remaining 17.39% (218 words) were over-stemmed.
- Complete word method properly stemmed 57.99% (727 words) of the total 1254 words. Out of the remainder 42.01% words, 13.64% (171 words) were over-stemmed, but for the remainder 28.37% (356 words) there were no stem generated. (The total stem generated by using complete word method was only 679)

Hence, peak and plateau method out performed entropy and complete word methods on this given test set.

## 5.5 DISCUSSION OF THE RESULT

The total corpus of test set was 1254 words. The expert judgment stem of the given test set produced 876 stems. This process reduced the size by 31.63%.

In the first testing result, Peak and plateau method properly stemmed 50.7% (636 words) of the total 1254 words. Out of the 49.3% wrongly stemmed words, 35.9%

(450 words) were under-stemmed and the remaining 13.4% (168 words) were over- stemmed. On the other hand, Entropy method properly stemmed 44.2% (554) of the total 1254 words. Out of the 55.8% wrongly stemmed words, 28.8% (361 words) were under-stemmed and the remaining 27% (339 words) were over-stemmed.

As indicated in the result the performance of both Peak and plateau and Entropy methods were low. The reason is that, the result of the successor variety stemmer was compared with the expert judgment that was performed by stemming words into their root form through removing infixes, prefixes and suffixes. However, the system developed in this research only conflates words into their common term. For example, words like “Cemrew” ጨምረው and “Cemro” ጨምሮ were changed in to “Cmr” ጭምር when they are stemmed by the expert. But the system stemmed both terms in to “Cemr” as a common stem. Besides, “hone” ሆነ, “honom” ሆኖም, “honuna” ሆኑና and “bihon” ቢሆን were stemmed at “hone” ሆነ by the Expert whereas the system stemmed them as “hon” ሆን. This difference contributed to reduced performance of the system.

The other reason that leads test result of the system to low performance is that, the domain diversity of the corpus affected the number of occurrence of the segment leading to taking improper segment. In addition, the complex nature of the Amharic language morphology, especially infix contributed more to the difficulty to get the right stem.

To improve the performance of the system, the system output was compared with the manually conflated words instead of the root of the word, since the developed system produced the stem by conflating words.

The system showed an improvement in performance when the output comparison was changed from using the root word by the stem of the word. As a result, the successor variety stemmer peak and plateau method performance was improved form 50.7% in to 71.8%, and the successor variety stemmer Entropy method performance was improved form 44.2% in to 63.95%.

In addition to the above methods of successor variety stemming, the complete word method was also tested to conflate terms. The result is discussed as follows.

To check the performance of the complete word method, the researcher tested the system by changing the number of the total corpus. When the number of the total corpus was 5,464, the complete word method generated 539 words as the stem. Out of 539 words, 338 words were correct stems, and 141 words were under stemmed (when compared with manually conflated words). This made the performance of the system to be 45.43%.

The total corpus was changed into 11,262 words. In this case, the complete word method generated 591 stems; out of these, 440 words were correct stems. The change of the corpus resulted in an increase in the performance of the system to 50.23%.

Again, the total corpus size was grown in to 24,555 words. When presented with this sample, the complete word method generated 646 stems. Out of 646 stems, 485 stems were correct. The performance of the system increased to 55.36%. Finally, the researcher tested the complete word method by using 43,088 corpus size. As a result, the complete word method generated 679 words as a stem. Out of that, 508 were correct stems. Its performance increased to 57.99%.

When the corpus size nearly doubled (from 24,555 to 43,088 words) the performance improvement was only 2.63%. That is, it increased by a decreasing rate. In general, it is observed that, the performance of the complete word method increased when the total corpus size increases.

# CHAPTER SIX

# CONCLUSION AND RECOMMENDATION

## 6.1 CONCLUSION

There have been a lot of related researches in the area of stemming for Amharic text. Alemayehu & Willett (2002) used the affix removal method and reported an accuracy of 95.9%; Betelihem (2002) used the n-gram method and reported the highest accuracy of recall 94.2% (obtained using bi-gram method); Atelach & Lars (2007) used the dictionary lookup method and reported the highest accuracy of 75% on news articles. None of the researches conducted so fare implemented and tested successor variety stemming method for Amharic texts. On the other hand, successor variety stemming method was implemented for stemming words in other languages. For instance, Al-Shalabi *et al.* (2005) used successor variety approach using cutoff method and Entropy method to stem words in Arabic text (reported 80% and 75% level of accuracy for cutoff and Entropy method respectively).

As indicated in chapter one the main objective of this research is to model and implement a successor variety stemming method for Amharic text.

To build the successor variety model, the researcher had performed data collection, preprocessing, experimentation, and testing (evaluation). The methodology adopted to this research was peak and plateau and Entropy successor variety stemming methods. In the course of conducting the experimentation, the size of the corpus had been changed to see the effect on the performance of the algorithm.

Based on the output of the experimentation, the researcher had made two remarkable observations and accordingly made decisions to improve the performance of the system (model) developed in this research.

The first observation made from the output of the experimentation was in relation to the decision for the cutoff value. The peak and plateau method suggests that a segment be made at the character whose successor variety is greater than both its preceding and following character. However, there are cases where we can get many peak points for a successor variety of a single term. In this case, the Amharic terms broken at these points produced segments that are not acceptable by the Language. Hence, the researcher found that breaking the terms at the highest peak give more acceptable cuts for segmenting Amharic texts.

Secondly, the selection rules (to determine the segment is a stem or an affix) suggested by Hafer and Weiss (1974) did not apply for the Amharic text. That is, it is observed that the frequency of the segments obtained as the result of the experimentation commonly fall with in the range of seven to twelve. This led to the decision of incorrectly rejecting some segments into affix lists while they could be actually correct stems. Hence, the researcher decided that the number of occurrence of the segment must be lowered to seven in order to get correct stems from the segments.

When presented with the test sample, the test result of the system (model) developed in this research showed that peak and plateau method had a performance of 71.8% accuracy while entropy and complete word methods showed the performance of 63.95% and 57.99% level of accuracy respectively.

From the result of the experimentation made, the researcher concluded that decrease in performance of the system was mainly due to the nature of the input corpus (input article type, and word selection procedures). In addition, the problem arises due to the nature of Amharic words as they have multiple affixes from the beginning, at the end and sometimes in the middle.

Finally, based on the observation made from the experimentation result, the peak and plateau method had better performance than the entropy method. Hence, the researcher concluded that peak and plateau method is the preferred successor variety stemming approach for Amharic text.

## 6.2 RECOMMENDATION

The following further research areas are recommended by the researcher.

- Experimenting by using a large amount of corpus may give good stemming result. However, limited corpus was used in this research. Future researchers may need to experiment the successor variety stemming method by inputting large amount of corpus.

- It would also be better if infix morphology analysis and removal is done to enhance the performance of the system.

- In this research, training and adjustment of the system was done in an extensive manual involvement. If this task is fully automated, the researcher believes that it would shorten the development time thereby increasing performance. Hence, efforts should be made by future researchers to automate this process.

- In this research the peak and plateau, entropy and complete word methods of segmentation are used and compared. It would be good if the other segmentation method was tested by using a huge collection of corpus to segment, and the results of the four segmentation methods were compared so as to decide which method is the most appropriate method for Amharic.

# REFERENCES

Al-Shalabi, R., Kannan, G., Hilat, I., Ababneh, A. and AL-Zubi, A. (2005). Experiments with the Successor Variety Algorithm Using the Cutoff and Entropy Methods, Information Technology Journal 4 (1), P. 56-63.

Atelach Alemu and Lars Asker (2007). An Amharic Stemmer: Reducing Words to their Citation Forms, Stockholm University/KTH, Sweden.

Ayele Bekerie. (1994) African Writing System. Cornell University. At URL http://www.library.cornall.edu/africana/writing_system/Amharic.html

Baeza-Yates, Ricardo and Ribeiro-Neto, Berthier. (1999). Modern Information Retrieval, The ACM Press.

Baye Yimam.(1997). "የአማርኛ ሰዋስው".Addis Ababa. ት.መ.ማ.ማ.ድ.

Baye Yimam.(2000E.C.). "የአማርኛ ሰዋስው". የተሻሻለ ሁለተኛ እትም. Addis Ababa. እሌኒ ማ.ኃ.የተ.የግ.ማኅበር

Bender, M. L., Sydney W. Head, and Roger Cowley. (1976). The Ethiopian Writing system. In Bender et al. (Eds.) Language in Ethiopia. London: Oxford University Press.

Bethlehem Mengistu. (2002). N-Gram-Based Automatic Indexing for Amharic Text. Addis Ababa University.

Daniel Yaqob. (1997). Transliteration on the Internet: The Case of Ethiopic. At URL http://yacob.org/papers/etinet.pdf

ECo SA newsletter. (2000). Vol. 1,1.

Getachew Haile. (1967). The Problems of the Amharic Writing System. A paper presented in advance for the interdisciplinary seminar of the Faculty of Arts and Education. HSIU

Gregory Grefenstette, Pasi Tapanainen. (19994). What is a word, Wha is a sentence? Problems of Tokenization. Rank Xerox Research Center. Grenoble Laboratory. France.

Hafer, M.A. and Weiss, S.F.(1974). Word segmentation by letter successor Varieties. Information Storage and Retrieval,

Lawerence Lo. (1996-2005). The Blackwell Encyclopedia of Writing System. At URL http://ancientScripts.com/ws.html

Lemma Lessa Ferede. (2003). Development of Stemming Algorithm for Wolaytta Text. (Masters Thesis) Addis Ababa University Faculty of Informatics Department of Information Science. (Unpublished)

Leslau,W. (2000). Introductory Grammar of Amharic. Wiesbanden: Harrassowitz.

Martin Braschler.(2003) How Effective is Stemming and Decompounding for German Text Retrieval?.

Kanaan, G., Al-Shalabi, R. and Sawalha M. (2005). Information Technology: Improving Arabic Information Retrieval Systems Using Part of Speech Tagging. Information Technology Journal 4 (1), 2005: 32-37.

Alemayehu, Nega and Willett, Peter. (2002). Stemming of Amharic Words for Information Retrieval. University of Sheffield. Sheffield. UK.

Hudson, Richard. (2003). An Encyclopedia of English Grammar and Word Grammar. At URL http://www.phon.ucl.ac.uk/home/dick/enc/intro.htm

Rijsbergen, C.J., (1999). Information Retrieval (sec. edt.). University of Glasgow. Scotland.

Salton, G. and McGill, M. J. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

Solomon Teferra Abate, Wolfgang Menzel. (2005). Automatic Speech Recognition for an Under-Resourced Language-Amharic. Department of Informatics, Natural Language Systems Group, University of Hamburg. Germany. At http://nlp.amharic.org/members/solomon/papers/interspeah2005.pdf

William B. Frakes and Ricardo Baeza-Yates (Eds). Information Retrieval: Data Structures & Algorithms. Englewood Cliffs, NJ: Printice-Hall, 1992

Wakshum Mekonnen. (2000). Development of a Stemming Algorithm for Affan Oromoo Text. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa. (Unpublished).

Zelalem Sintayehu. (2001). Automatic Classification of Amharic News Items: The Case of Ethiopian News Agency. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa. (Unpublished).

# APPENDIX

**Appendix 1:** The conversion of different alphabet orders with the same sound into a common alphabet order

| Alphabet | Convert into |
|---|---|
| ሀ, ሃ, ኀ, ኃ, ሐ, ሓ, ኻ | ሀ |
| ሁ, ሑ, ኁ, ኹ | ሁ |
| ሂ, ሒ, ኂ, ኺ | ሂ |
| ሄ, ሔ, ኄ, ኼ | ሄ |
| ህ, ሕ, ኅ | ህ |
| ሆ, ሖ, ኆ, ኾ | ሆ |
| ሰ, ሠ | ሰ |
| ሱ, ሡ | ሱ |
| ሲ, ሢ | ሲ |
| ሳ, ሣ | ሳ |
| ሴ, ሤ | ሴ |
| ስ, ሥ | ስ |
| ሶ, ሦ | ሶ |
| አ, ኣ, ዐ, ዓ | አ |
| ኡ, ዑ | ኡ |
| ኢ, ዒ | ኢ |
| ኤ, ዔ | ኤ |
| እ, ዕ | እ |
| አ, ዖ | አ |
| ጸ, ፀ | ፀ |
| ጹ, ፁ | ፁ |
| ጺ, ፂ | ፂ |
| ጻ, ፃ | ፃ |
| ጼ, ፄ | ፄ |
| ጽ, ፅ | ፅ |
| ጾ, ፆ | ፆ |
| ቆ, ቈ,ቌ | ቆ |
| ቂ, ቊ, ቍ | ቂ |
| ኪ, ኩ | ኮ |
| ኋ, ኍ, ኌ | ኋ |
| ጎ, ጐ | ጎ |

## **Appendix 2:** The sample corpus of the research with its transliterated version

ዶ/ር አሸብር የኢትዮጵያ እግር ኳስ ፌዴሬሽን ፕሬዚዳንት ሆነው ተመረጡ
የኢትዮጵያ እግር ኳስ ፌዴሬሽን ለመጪዎቹ አራት ዓመታት በበላይነት የሚያስተዳድሩ አባላት ስም ባለፈው ቅዳሜ ሐምሌ 9/1997 በጊዮን ሆቴል በተደረገው የምርጫ ሥነ -ሥርዓት ይፋ ሆኗል። በጠቅላላ ጉባኤው ላይ ከትግራይ በስተቀር የአዲስ አበባ መስተዳድርና የድሬዳዋ ካውንስልን ጨምሮ የ10 ክልል ፌዴሬሽኖች ተወካዮች የፕሪምየር ሊግና የብሔራዊ ሊግ ክለቦች ተወካዮች የእግር ኳስ ማኅበራት ተወካዮች በተገኙበት ስብሰባ የኖርማላይዜሽን ኮሚቴውን በፕሬዚዳንትነት ሲመሩ የቆዩትን ዶ/ር አሸብር ወ/ጊዮርጊስን በከፍተኛ ድምፅ መርጠው የፕሬዚዳንቱን ሥልጣን መልሰው ተረክበዋል። ዶ/ር አሸብር አሸናፊ የሆኑት 63 ድምፅ በማግኘት ሲሆን አብረዋቸው በእጩነት የተወዳደሩት አምባሳደር ኡፋዮ አለሁ 3 ድምጽ ማግኘታቸው ታውቋል። ለሥራ አስፈፃሚነት በተደረገው የምርጫ ሥነ ሥርዓት ደግሞ 10 አባላት የተመረጡ ሲሆን እነርሱም ኢንስትራክተር ካሣሁን ተካ፣ አቶ አርአያ ተስፋዬ፣ አቶ ወርዶፋ በቀለ፣ አቶ ልዑልሰገድ በጋሻው፣ አቶ ጌታቸው ገ/ማርያም፣ ዶክተር ቶውፊክ አብዱላሂ፣ አቶ አሸናፊ እሸቱ፣ ዶክተር አደፍርስ በላቸው፣ ረዳት ፕሮፌሠር ሲሣይ ዘለቀ እና አቶ አቡ ያደታ ናቸው። ከላይ የተጠቀሱት አስሩ የሥራ አስኪያጅ ኮሚቴዎች ከ28-48 ድምጽ በማግኘት ነው አሸናፊ የሆኑት።

do/r exebr yeityoPya Igr kWas fEdErExn prEzidant honew temereTu
yeityoPya Igr kWas fEdErExn lemeCiwocu erat `ametat bebelaynet yemiyastedadru ebalat sm balefew qdamE HemlE 9/1997 begiyon hotEl betederegew yemrCa `sne -`sr`at yfa honWal. beTeqlala gubaEw lay ketgray besteqer yeedis ebeba mestedadrna yedrEdawa kawnsln Cemro ye10 kll fEdErExnoc tewekayoc yeprimyer ligna yebHErawi lig kleboc tewekayoc yeIgr kWas ma`hberat tewekayoc betegeNubet sbseba yenormalayzExn komitEwn beprEzidantnet simeru yeqoyutn do/r exebr we/giyorgisn bekefteNa dm`S merTew yeprEzidantun `slTan melsew terekbewal. do/r exebr exenafi yehonut 63 dm`S bemagNet sihon ebrewacew beICunet yetewedaderut embasader ufato elehu 3 dmS magNetacew tawqWal le`sra esfe`Saminet betederegew yemrCa `sne `sr`at degmo 10 ebalat yetemereTu sihon Inersum instrakter ka`sahun teka, eto ereya tesfayE, eto werdofa beqele, eto l`ulseged begaxaw, eto gEtacew ge/maryam, dokter towfik ebdulahi, eto exenafi Ixetu, dokter edefrs belacew, redat profe`ser si`say zeleqe Ina eto ebu yadeta nacew. kelay yeteTeqesut esru ye`sra eskiyaj komitEwoc ke28-48 dmS bemagNet new exenafi yehonut.

# DECLARATION

This thesis is my original work, has not been presented for a degree in any other university and that all sources of material used for the thesis have been duly acknowledged.

_____

**Genet Mezemir Fikremariam**

**THE THESIS HAS BEEN SUBMITTED FOR EXAMINATION WITH MY APPROVAL AS UNIVERSITY ADVISOR**

_____

**Ermias Abebe**

**Addis Ababa University**