ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

A STEMMING ALGORITHM DEVELOPMENT FOR TIGRIGNA LANGUAGE TEXT DOCUMENTS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

BY
GIRMA BERHE
JUNE 2001

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

A STEMMING ALGORITHM DEVELOPMENT FOR TIGRIGNA LANGUAGE TEXT DOCUMENTS

BY

GIRMA BERHE

Name and Signature of Members of the examining Board

Ato Getachew Jemaneh, Chairman, Examining Board

Ato Tesfaye Biru, Advisor

Dr. Gebre-Medhin Simon, Advisor

Dr. Kamal Bechkom, External Examiner

ACKNOWLEDGMENT

It gives me a great pleasure to use this opportunity to thank my advisors, colleagues, friends, institutions and others that have contributed in one or other way to the success of this research.

My deepest thanks are due to Ato Tesfaye Biru and Dr. Gebre-medhin Simon, my thesis advisors, for their invaluable advise. Their critical comments and supportive ideas were very helpful and important for the success of this work.

I would like to thank my colleagues and friends who have been giving me moral strength to work hard and complete my two-year study in School of Information Studies for Africa (SISA).

I would also like to thank those who have been providing me books and other materials on Tigrigna language.

Finally I am very grateful to my family and especially my parents.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS USED

Symbol	meaning	
CV	=	Consonant and vowel sequence
/ /	=	Affix (es)
1ps	=	1st person singular
1pp	=	1st person plural
2sm	=	2nd person singular masculine
2sf	=	2nd person singular feminine
2pm	=	2nd person plural masculine
2pf	=	2nd person plural feminine
3sm	=	3rd person singular masculine
3sf	=	3rd person singular feminine
3pm	=	3rd person plural masculine
3pf	=	3rd person plural feminine

CONSONANTS

b ቡ = voiced glottalized bilabial stop

p ፐ = central glottalized bilabial stop

P ጰ = voiceless glottalized bilabial stop

d ደ = voiced glottalized dental stop

t ተ = central glottalized dental stop

T ጠ = voiceless glottalized dental stop

g ገ = voiced velar stop

k ከ = central velar stop

Q ቐ = voiceless velar stop

? አ = voiceless laryngeal stop

f ፈ = voiced laryngeal stop

s ሰ = voiced glottalized dental fricative

z ዘ = central glottalized dental fricative

S ጸ = voiceless glottalized dental fricative

x ሸ = voiceless palatal fricative

Z ዠ = voiced palatal fricative

H ሐ = voiceless pharyngeal fricative

~ ዐ = voiced pharyngeal fricative

h ሀ = voiced laryngeal fricative

c ቸ = voiceless palatal affricate

j ጀ = voiced palatal affricates

C ጨ = voiceless glottalized palatal affricate

m መ = voiced nasal bilabial

n ኀ = voiced dental

N ኘ = voiced nasal palatal

l ለ = voiced dental flab

r ረ = voiced laryngeal fricative

w ወ = voiced glottalized bilabial

y የ = voiced glottalized palatal

VOWELS

i ኢ = high front unrounded

E ኤ = mid front unrounded

e እ = mid central unrounded

a አ = low central unrounded

u ኡ = high back rounded

o ኦ = mid back rounded

ɨ ኧ = high central unrounded

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Variant word forms that are likely to be encountered in indexing and retrieval are one of the causes of the problems that are involved in the use of free-text retrieval system. The variant word forms used in indexing and searching are likely to be of comparable importance in determining the relevance of a document to a user query that specifies just a single form. Reducing the variant words into one form improves performance of IR system and this can be achieved by a conflation technique, which is usually stemming that is established in this work. Stemmers are used in information retrieval to reduce as many related words and word forms as possible to a common form, which can then be used in the retrieval process.

This research explores the possibility of developing a stemmer to conflate variant words of Tigrigna language for use in IR of the language. Tigrigna belongs to the Semitic language group. These languages have a common grammatical system based on a root-pattern structure. Consonants bear the basic meanings while vowels form different patterns. Stems are built from consonantal roots before other words are built from stems. Tigrigna uses affixation to derive different word forms from stems. Common affixations are prefix, suffix, prefix-suffix pair and reduplication. Tigrigna uses extensive concatenation of affixes and can result in relatively long words, which often contain an amount of semantic information equivalent to a whole English phrase, clause or sentence. Due to this complex morphological structure, a single Tigrigna word can have thousand variants.

To design the stemmer, a sample text was collected from three different sources. The experiment in word-distribution on the sample data shows that words exist in their variants

across the text and singleton words constitute large percentage of the text. This resulted in low word-ratio and deviation from Zipf's law.

A stemmer is developed which is iterative and uses context-sensitive rules that removes prefix, suffix, prefix-suffix pair and reduplication of single and double letters. A semi-automated procedure was used to compile stopwords and affixes. The stemmer was tested on sample data of 1568 words, which were selected randomly from the sample texts. In this experiment the stripping procedures were applied in the order of prefix-suffix, double letter reduplication, prefix, suffix and single letter reduplication. The result of the experiment shows that, the stemmer performs at accuracy of 84% and brings a dictionary reduction of 32.40% and 54.6% for stem and root respectively.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The mass production of electronic information, the development of computerized library collections and increasing awareness of the importance of information by the society in day-to-day activity (business, research etc.), demands storing, maintaining and retrieving information in a systematic way. Systems that can store information in an organized manner and enable efficient access and retrieval are usually referred to as information retrieval (IR) systems. Information retrieval in its widest sense consists of the processes whereby information (or documents containing it) is stored and made available to users, and all the manual and automated systems for organizing documents, or descriptions of documents, and for recovering those relevant to a user's information needs (Nail, 1999).

Today, information-processing activities are carried out with the assistance of automatic equipment. Salton (1983) defines an automated information retrieval system as software/hardware package that allows users to query and receive information that is stored in a computer database.

As described by Chowdbhury (1999), an IR system is composed of different components, which includes documents (items of information), user's request (query), and matching of these queries with the document database. In information retrieval the stored information items and the incoming search request are normally represented by sets of content identifiers variously known as keywords, index terms, or simply terms (Salton et al., 1981).

The process of selecting keywords for representing a document is called indexing. Indexing can be done manually or automatically. In manual indexing the process of selecting the keywords is done by trained indexers, who are knowledgeable about the subject matter of the database, through scanning of the entire text or selected portions of the text, like titles, abstracts, or topic sentences (Tesfaye, 1987). On the other hand, in automatic indexing the operation is carried out with the aid of modern computing equipment.

In most IR systems, automatic indexing involves (Rijsbergen, 1975; Salton, 1983) selection of individual words, use of stopwords, use of conflation and weighting of the resulting term. After the individual words that constitute the document are identified, high frequency function words are eliminated using the stopword list. The function words are poor discriminators and cannot be used by themselves to identify document content. The next step, following the removal of stopwords is the identification of index terms and their assignment to the document. Since the same word could be represented in different forms, conflating the variant words to word stem form is useful. Reducing the variants produces a higher frequency of occurrence of the word stem in the document than any of the variant forms.

Different weighting mechanisms are proposed for measuring the usefulness of the remaining word stems for indexing purpose based on frequency technique. One of the measurements proposed by H.P. Luhn (Salton, 1983) assumes that the value or weight of a term assigned to a document is proportional to the term frequency. The rational of this approach is that a term, which occurs frequently in a document, is more likely to describe the content of the document. After a threshold value is determined, words with frequency above this threshold, those that are high frequency words, are removed. In the same way, low threshold value is determined and words below this are removed. These two operations leave words with

medium frequency and those are used to index the document. The weight (w_{ik}) of a term k in document i is calculated by,

$$w_{ik}=f_{ik}$$

Even though, the above weighting method enables to represent documents and so can be retrieved by those terms, it may not be good enough to distinguish or discriminate the documents in the collection. This leads us to another weighting system introduced by Spark Jones. Spark Jones introduced inverse document frequency measurement (Salton, 1983) which assigns high weigh to terms, which have substantial frequency in some individual documents of a collection but with relatively low overall collection frequency. These terms help in retrieving the documents to which they are assigned while also distinguishing them from the remainder of the collection. Such terms are considered as being potentially of greater importance for retrieval purpose. The weight (importance value) of a term k in a document collection of n is given as,

$$w_k=\log_2 \frac{n}{f_k}$$

Where n is the total number of documents in the collection, f_k is the number of documents in which k occurs, and w_k is the weight assigned to term k.

1.2 STATEMENT OF THE PROBLEM

According to Hetzron (1969), the EthioSemitic and South-Arabia languages are categorized under South-Semitic. The EthioSemitic languages are further classified as North-EthioSemitic and South- EthioSemitic. Tigrigna together with Geez and Tigre forms the North- EthioSemitic language and the South- EthioSemitic includes Amharic, Argoba, Harari etc. Tigrigna is spoken in the present day Tigray Region and the State of Eritrea.

According to the Office of Population and Housing Census Commission of Ethiopia (1999) there are about 3,371, 808 Tigrigna speakers of whom 3,224,875 speak the language as a mother tongue and 146,933 as a second language. However, before Eritrea became an independent country (since 1993), the total number of Tigrigna speakers in Tigray and Eritrea was 4,068,789 as stated in the 1984 census. At present, Tigrigna is the second most widely spoken Semitic language next to Amharic.

Literatures, books, newspapers, and magazines published in Tigrigna have been increasing over the past years. At the moment, Tigrigna is the medium of instruction in the primary and junior secondary schools in Tigray based on the new Educational and Training Policy of Ethiopia (MOE, 1994, sub-article 3.5.1). For this reason educational materials are produced in Tigrigna for Primary and Junior Secondary Schools. There are newspapers (e.g. ወይን and መቻልሕ ትግራይ), and magazines (e.g. ኣሰር) published in Tigrigna. There are also other newspapers and magazines published in Tigrigna, which are distributed in the Region. Since the introduction of Gee'z word processing, electronic documents published in Tigrigna have been produced for different purposes. CD-ROM's publication and Web-page development are also emerging.

Retrieval systems enable to get information easily and timely. In today's shrinking world, it is becoming evident that there is a large body of information and research available only in the language of the primary researcher, and much information cannot be shared between research communities without considerable translation and time delay (Hlava et al., 1997).

In order to access the information one needs to know the languages or build bilingual or multilingual retrieval systems. Nowadays, the question of multilingual access and multilingual information retrieval is becoming increasingly important for two reasons (ELISE II project 1999): first of all because of the recent rapid diffusion over the international computer networks of world-wide distributed document bases. Secondly, because multilingual digital libraries are becoming more common, for example, in countries with more than one national language, in countries where both the national language and English are commonly used for scientific and technical documentation.

The ultimate goal for systems of multilingual information retrieval is to offer users the opportunity to query in any language and retrieve a merged and ranked set of documents that match the query in whatever language they are stored (Peters, 2000). However, information access in multiple languages also implies an understanding of the issues involved in monolingual IR for different language types.

By developing retrieval tools for Tigrigna, one can design a retrieval system for the language and hence make the hard copy as well as the electronic documents published in Tigrigna accessible to users. A bilingual (eg. English-Tigrigna) retrieval system can also be designed that allows users to query using Tigrigna to access English documents or other languages.

Techniques for storing, maintaining, and retrieving data/information from English document databases have been studied, implemented, and tested for the last three decades (Al-Kharashi et al, 1994), but detailed works have not yet been done to see how well these techniques will work on Ethiopic documents in particular for Tigrigna documents. There are attempts made on IR of Ethiopic documents, which are basically developments of stemming algorithms for Amharic and Afaan Oromoo languages by Nega (1999) and Wakshum (2000) respectively.

In developing IR systems for a particular language, the study of automatic indexing technique for the language is necessary. Among the processes of automatic indexing, stemming and stopwords are highly dependent on the document language. The motivation for using stemming is the need to increase effectiveness of retrieval system since stem of a term represents a broader notion than the original term itself (Al-Kharashi et al, 1994). To my knowledge, so far no IR computational tools like stemming have been developed for Tigrigna language. Therefore, it is felt necessary to undertake a research on the development of stemming algorithm for the language, hence this study.

1.3 SIGNIFICANCE OF THE STUDY

One of the tools of IR is stemming. By developing a stemming algorithm for Tigrigna, variant words can be conflated. In Tigrigna a word can have thousand variants and conflating these variants increases retrieval effectiveness. It also reduces storage of index files. In addition to the above results, it can also give the following benefits:

- This research can be taken as a start for doing further research on computational work of the language;
- For developing computational tools for the language such as spell checher, grammar, thesauri etc.
- By developing stemming algorithm and other retrieval computation of the language, IR systems for documents of the language can be designed, which helps to support the cultural, economic, and social activities in Ethiopia in general and in Tigray Region in particular;
- For developing bilingual and multilingual retrieval systems which facilitate cross-cultural interaction among the societies in Ethiopia and others etc.

1.4 OBJECTIVES

1.4.1 General objective

The main objective of this research is to explore the possibility of developing a stemming algorithm for Tigrigna language to conflate word variants of the language.

1.4.2 Specific objectives

The specific objectives of the study are to:

- review the morphology of Tigrigna language;
- review techniques of stemming algorithms already developed for other languages;
- maintain a list of affixes used in Tigrigna texts;
- compile stopword list;
- write a program for stemming inflectional and derivational affixes; and
- test the stemmer on sample words.

1.5 METHODOLOGY

1.5.1 Review of related literature

Since stemming algorithm is language dependent, to develop stemming algorithm one has to know the morphology of the language or work with a knowledgeable person. To understand the morphology of Tigrigna language review of works on the language was done by consulting different sources such as books, theses, journals etc. Additional information was also collected through personal discussion with knowledgeable persons.

There are a number of stemming algorithms developed for different languages. The approaches and techniques used in these algorithms especially for English and Semitic languages (such as Arabic and Amharic) were studied from different sources such as journals, books, thesis, Internet and other related resources.

1.5.2 Programming Techniques

Some of the techniques used in the existing stemming algorithms developed for English and Semitic languages such as Arabic and Amharic were adopted in developing the Tigrigna stemmer.

To test the stemmer, a prototype program was written using C++ programming language. This language was selected for the following reasons:

- C++ is rich of string manipulation library functions, which is important feature of stemming algorithm;
- a program written in C++ is relatively easier to convert to other programming languages;
- experience of the researcher in writing programs using C++.

1.5.3 Test Data

For the purpose of producing stopwords and affixes, sample Tigrigna texts were collected from newspapers, magazines and fiction on social, political and culture. The selection of topics from these materials was done randomly.

The stemmer was tested on 1568 words selected randomly from the sample texts. Evaluation of the stemmer was done through error counting mechanism. The errors were basically overstemming, understemming, order and other. After investigation of the errors, the stemmer was modified.

1.6 SCOPE AND LIMITATION

The algorithm conflates only inflectional and derivational affixation. This algorithm does not conflate compounding and irregular word formations. To improve the performance of the stemmer, detail analysis on morphology of the language is important.

The stemmer has five stripping procedures: prefix-suffix pair, prefix, suffix, single-reduplication, and double-reduplication. Some of the errors were resulted from the order of the procedures. Due to limitation in time, different possible ordering of the stripping procedures could not be tested. Doing evaluation test on order of the procedures and selecting the best possible one could improve the performance of the stemmer.

1.7 ORGANIZATION OF THE THESIS

The thesis has five chapters. In Chapter 1 discussion about stemming and its importance in IR systems, the rational for need of stemming algorithm for Tigrigna documents and its importance are presented. As in other many studies, the chapter also contains objectives, methodology and scope and limitation of the study.

Discussion on conflation techniques, types and their mode of operation were presented in Chapter 2. Review on commonly used English stemmers such as Lovin's and Porter's are also discussed in this chapter. Chapter 3 deals with review on morphology of Tigrigna language with emphasis on inflectional and derivational affixes of the language. Examples and illustration on the usage of affixes are presented in the chapter.

Development and experiment of the stemmer are covered in Chapter 4. Finally the results obtained from the experiment, the conclusion reached and recommendations identified for future work are included in Chapter 5.

CHAPTER 2

CONFLATION TECHNIQUES

2.1 INTRODUCTION

One of the many characteristics of a natural language that must be taken into consideration when designing a free-text retrieval system is morphological variation of words (Popovič et al., 1992). The variant word forms are likely to be of comparable importance in determining the relevance of a document to a user query that specifies a single form, and this leads to the development of conflation techniques, which permit the matching of different forms of the same word.

As explained by Lennon et al. (1981), one of the main problems involved in the use of free text for indexing and retrieval is the variation of word forms that is likely to be encountered. The possible sources of variations are spelling errors, alternative spellings, multi-word concepts, transliteration, affixes and abbreviations. For example, the stem CONNECT can give rise to CONNECTION, CONNECTIONS, CONNECTIVE, CONNECTIVITY, CONNECTED, and CONNECTING. Reducing the variant words to the stem CONNECT resulted in retrieval of documents that are indexed with any of the variant words and hence increase retrieval effectiveness.

In terms of IR, conflation has two functions (Harman, 1991): reducing the total number of distinct terms with a consequent reduction in index storage required and updating problems; and bringing similar words, having similar meaning, to a common form with the aim of

increasing retrieval effectiveness. The later aspect has become more important as storage and processing cost have decreased over the past years.

Conflation can be achieved by either manual or automated means (Popovič et al., 1992). Manual conflation is normally effected by right-hand truncation at search time, with the truncation being carried out by the searcher. Extensive experience is needed if effective truncation is to be achieved. To apply truncation the searcher has to know where to truncate the word, other wise it may lead to over or under truncation.

If a word is over truncated then the stem becomes too short and may result in truncation of unrelated words to the same stem. For example, NEUTRON and NEUTRALIZE are retrieved by the same stem NEUTR*, even though they describe different things. Under-truncation on the other hand occurs when a word is not truncated enough and excludes words of the same root. For example the truncation of CONNECTIONS to CONNECTION rather than CONNECT* excludes CONNECTIVE and CONNECTING, which as far as retrieval is concerned may mean the same.

Another way of alleviating the problem of word variant is by using automatic conflation, which is an algorithm that permits the matching of different forms of the same word automatically. Automatic conflation can be broadly divided into two main classes (Ekmekçioglu et al., 1996): stemming algorithms and string similarity algorithms. Stemming algorithms are language dependent, which are designed to handle morphological variations where as string-similarity algorithms, which are (usually) language independent are designed to handle all types of variants (see Section 2.2 and 2.3 for detail). Manual truncation is applied

only to words in queries while stemming is additionally applied to words in queries and documents as they are added to the database.

2.2 STEMMING ALGORITHM

Lovins (1986) defines a stemming algorithm as a "procedure to reduce all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes". For example the words "reading", "readers", and "reads" are reduced to the stem "read". For languages such as English, the root of a word is obtained by removing both suffixes and prefixes and the stem is obtained by deleting only suffixes (Savoy, 1993). But in Semitic languages (Al-Kharashi et al., 1994), a root consists of consonants (radicals) and the stem is a combination of a root and derivational morpheme to which one or more affixes can be added.

Various stemming algorithms have been reported in the literature for languages such as English (Porter, 1980; Lovins, 1968), French (Savoy, 1993), Turkish (Ekmekçioglu, 1996), Slovene (quoted by Popovič, et al., 1992), Malay (quoted by Ahmad, et al., 1996) etc. Since recent times, attempts are also being made to develop stemming algorithm for Ethiopic languages, such as Amharic (Nega, 1999) and Afaan Oromoo (Wakshum, 2000).

Several works have been done to study the effectiveness of stemming for retrieval purposes (Popvič et al., 1992). Studies conducted by Lennon et al. (1981) indicated that better result could be obtained using stemming which is comparable to manual right-hand truncation. The experiment done by Al-Kharashi et al. (1994) to compare words, stems and roots as index terms in Arabic language text retrieval supports the above idea. The result of the experiment

shows superiority of root-and stem retrieval methods over word-retrieval methods for Arabic data. On the other hand works by Harman (1991), Walker et al (quoted by Popvič et al., 1992) indicated that stemming did not result in consistent improvement in the effectiveness of retrieval when compared with searches in which stemming was not used. Popovič et al. (1992) concluded that the effectiveness of a stemming algorithm is determined by the morphological complexity of the language that is designed to process. Al-Kharashi et al. (1994) and Nega(1999) found that using stemming gives better performance than without. Their works were on Arabic and Amharic (both are Semitic languages), which are morphologically complex languages compared to English.

The nature of stemming algorithms may vary considerably depending on whether a stem dictionary is being used, whether a suffix list is being used, and of course on the purpose for which the stemmer is designed, but most of them are based on certain principles and procedures (Lovins, 1968; Lennon et al, 1981). These procedures involve either the removal of the single longest matching suffix or the iterative removal of several suffixes. The rational behind iterative approach is the fact that suffixes are attached to stems one after the other. In the iterative approach, suffixes are removed from the stem in the order of their derivational rules. In this approach the stripping starts from the end of the word and working towards the beginning. For instance, if the word HOPEFULNESS is considered, the suffix –NESS will be removed in the first iteration and re-considering HOPEFUL, the suffix –FUL will be removed leaving HOPE as a final stem. According to Savoy (1993), suffixes are classified according to their derivational rules (e.g., the first class groups plural inflections together with the suffixes "-ed" and "-ing"). Stripping is done based on the class order that is defined by the programmer. Other approaches such as proposed by Paice (quoted by Savoy, 1993) may also exist, which are iterative but do not have their endings classified.

The longest-match approach, on the other hand, removes the longest suffix possible. If the same word HOPEFULNESS is considered, the suffixes in the word are: -NESS, -FUL, and –FULNESS. Therefore the algorithm removes –FULNESS from the word. The problems of using longest-match approach compared to iterative method are: need for generating all possible combinations of affixes and processing and storage space required (this may not be critical problem of today's system) and change of affixes during concatenation. Using the iterative approach has also its own problem (Wakshum, 2000): the need of examining large number of endings, which is a laborious activity and a preparation of list of order class.

A stemming algorithm might be context-free or context-sensitive. Context refers to any property that is attached to the remaining stem and the usage of the suffix (Tesfaye, 1987). In context-free algorithms, no restriction is applied on the stem and hence no additional operations are required to check restrictions. This leads to simplicity in developing context-free stemming algorithms and such algorithms may also be more efficient. But most of stemming works (Lovins, 1968; Porter, 1980; Savoy, 1993; Ahmad, 1996) indicated that better result could be obtained by adding constraints to stripping operation, that is, using context-sensitive, which are mainly language dependent. Context-sensitive rules specify particular circumstances in which each suffix may be stripped from an input word. Savoy (1993) described three general types of constraints: quantitative, qualitative, and recording rule.

In quantitative constraints minimum length for the remaining stem is set when a suffix is removed. This helps not to remove ending from a stem in which the ending is in the suffix list but actually not a suffix for that stem. For example for the word ABILITY and suffix –

ABILITY, as the remaining stem must not be zero, the suffix -ABILITY will not be stripped from the word.

Qualitative constraints define conditions to be satisfied by the ending of the remaining stem. For example in English, a stem does not end with that is, stripping "ize" from "seize" is not allowed.

Spelling corrections and adjustment rules must be used to conflate the words to exact stem. A recording rule is a transformation of the form AxC →AyC, where A and C specify the context transformation, x is the input word (string), and y is the transformation string. In the English language, instances of such conditions include:

- removal of one of the double consonants ('b', 'd', 'g', 'm', 'n', 'p', 'r', 's', or 't') at the end of the stem to conflate terms like HOPE, HOPES, HOPEFUL and HOPPING;
- turning terminal 'd', 'r', 't', 'z', into 's' at the end of the stem to conflate terms like ADMIT, ADMITTANCE AND ADMISSION; or
- changing '-rpt', into '-rb' to conflate terms like ABSORB, ABSORBING and ABSORPTION.

2.3 STRING-SIMILARITY

It is pointed out by Tesfaye(1987) that though stemming is easy to implement and provides a highly effective means of conflating words with different suffixes, there are other types of word variants, which are likely to occur in free-text databases. Other conflation mechanisms as indicated by Tesfaye have been suggested to handle such cases. One of these mechanisms is using string-similarity. This method is especially used during search session.

In IR system, string-similarity approach for conflation involves the system of calculating a measure of string similarity (similarity coefficient) between an input query term and each of the distinct terms in the database. In a typical search session those database terms with a similarity coefficient greater than some threshold value are then displayed to the user for the possible inclusion in the query. Freud & Willette (quoted by EkmekÇioglu et al., 1996) suggested the N-gram matching as one of the techniques in calculating the similarity coefficient. An N-gram is a set of n consecutive characters extracted from a word. The assumption of this approach is that, similar words will have a high proportion of n-grams in common. Typically values for n are 2 or 3, these corresponding to the use of digrams or trigrams, respectively (EkmekÇioglu et al., 1996).

For example, considering the two words "AGRICULTURE" and "AGRICULTURAL" the following sets of digrams can be extracted:

AGRICULTURE {AG, GR, RI, IC, CU, UL, LT, TU, UR, RE,}

AGRICULTURAL {AG, GR, RI, IC, CU, UL, LT, TU, UR, RA, AL}

Then similarity coefficient such as Dice's (see other formulas in Appendix VI) may be used to quantify the degree of similarity. Dice's formula of similarity coefficient (S) is given as follows (EkmekÇioglu et al., 1996):

$$S = \frac{2C}{(A+B)}$$

Where A is the number of elements in the first set, B is the number of elements in the second set, and C is the number of elements common to A and B.

Ekmekcioglu et al. (1996) proved from experiment that n-gram matching is not as such poor compared to stemming in conflating words in Turkish text.

2.4 STOPWORD LIST

Two related facts have been observed in the earliest days of IR. First, a relatively small number of words account for a very significant fraction of all text's bulk. Words like "it", "and", and "to" can be found in virtually every sentence in English. Second, these noise words (less important words) make very poor index terms. Users are unlikely to ask for documents about "to", "and" or "it". Hence it has been a tradition in setting up IR systems to discard these very common words of the language during indexing (Porter, 2000).

The removal of stopwords from indexing and query, results in effectiveness of retrieval (Savoy, 1993). This is because it reduces storage requirement and increases the matching of a query with index terms of a document. Therefore, compiling a stopword list (sometime referred to as stoplist or negative dictionary) is important in building IR system. Getting a list

of stopwords can be done by sorting a vocabulary of a text corpus for a language by frequency, and picking off high and low frequency words.

2.5 REVIEW OF STEMMING ALGORITHMS

Various stemming algorithms have been reported in the literature ranking from a weak stemmer, which removes only plural markers to a complex one that removes suffixes and prefixes. From the available literature, stemming algorithms for languages such as English (Lovins, 1968 and Porter 1980), Arabic (Al-Kharashi, 1991), Slovene (Popovič, 1992), French (Savoy, 1993), Turkish (Ekmekçioglu et al. 1996), Malay (Ahmed et al, 1996), Amharic (Nega, 1999), Affaan Oromoo (Wakshum, 2000) were reviewed. Basically, the stemming algorithms follow the principles and approaches employed by the two most common English stemmers; Lovins (1968) and Porter (1980). These two algorithms that form the bases of most algorithms employ suffix stripping. For the purpose of discussion the stemming algorithms of Lovins, Porter, Nega and Wakshum are presented.

Lovins (1968) algorithm is based on longest match principle, and uses a list of 260 endings sorted in decreasing order of length. For example, a word COMPUTATIONALITY would be stemmed to COMPUT if the suffix –ATIONALITY were included in the list. After a suffix is removed, the stem is compared with one of the rules (34 recoding rules) to consider spelling exceptions. The rules define, for instance, the treatment of a suffix preceded by double consonants (such as STEMMING to STEM), or minimal stem size must be retained (such as the removal of ING from WORKING but not from SING) etc.

The problem of using longest match approach is the need to have a larger affix list: basic and possible combinations. This requires storage cost and compiling and processing the list. Storage cost has been decreasing during the last decade so maintaining a large list is not a problem nowadays, but the process of producing the list may not be easy, especially for languages with complex morphology.

Porter's (1980) stripping procedure uses the iterative approach. It operates in five stages, using five different classes of suffixes (a total of about 60) to simulate the inflectional and derivational process of words. Some of the rules do not actually delete the suffix but they transform endings of the stem to new endings. For example, if we remove the suffixes –ing from absorbing and –ion from absorption, it gives us absorb and absorpt which are different stems. But using the recording rule that transforms endings from –rpt to –rb, the two words are conflated to the same stem that is absorb.

To my knowledge there are two attempts made in developing stemming algorithms on Ethiopian languages: Amharic (Nega, 1999) and Afaan Oromoo (Wakshum, 2000). Amahric belongs to the Ethiopian-Semitic language group. Nega discussed the complexity of the language and stressed the importance of stemming to achieve a high recall result in the retrieval system of the language. An Amharic word can have more than a thousand variants (Nega, 1999). Reducing those variants of the word into one stem increases the matching and as result a high recall. The stemmer used a context-sensitive iterative procedure that removes both prefixes and suffixes. To measure performance of the stemeer, it was tested on a sample data of 1221 words. The result of the experiment shows that, the stemmer performed at an accuracy of 95.9%.

As quoted by Wakshum (2000), Afaan Oromoo belongs to the Cushitic branch of the Afroasiatic language. Wakshum (2000) also says that the main word formation process of Afaan Oromoo is through suffixation and the only prefix in Afaan Oromoo is (hi) ni and occurs usually as hin or ni in texts. The Afaan Oromoo stemmer developed by Wakshum uses the longest-match approach. Wakshum used a semi-automated procedure to produce suffix list. The stemmer performed at an accuracy of 92.52% based on the sample data of 1061 word.

CHAPTER 3

MORPHOLOGY OF TIGRIGNA LANGUAGE

3.1 INTRODUCTION

Natural languages have intricate systems to create words and word forms from smaller units in a systematic way. The part of linguistics dealing with these phenomena is morphology.

The aim of morphology is to provide a theory that describes the word structure of the language. Morphology studies the internal structure of words. Lexical is one component of morphology which contains words and stems, the blocks of word formation rules, i.e. the rules of derivations and compounding, evaluative rules which include diminutive, inflectional rules, and readjustment rules (Scalise, 1984). As cited by Wakshum (2000), Silzer defined morpheme as the minimal linguistic unit of the language that carries meaning. According to Schiffman (cited by Wakshum, 2000) morphemes are of two types: free and bound. The free morphemes occur by themselves for example "come" in English language. Bound morphemes exist in attachment with other morphemes. They are either affix or root. Bound morphemes become complete after they pass through morphological process. The absence of inflectional elements in Latin and Italian makes word bound morpheme, for instance /lup-i/"wolves" becomes /lup-/, and /can-i/"dogs" becomes /can-/ and we can see that /lup-/and /can-/ are not free and cannot appear on the surface while inflected words can (Tesfai, 1993). As cited by Tesfai (1993), Anderson says that derivational rules are used to form words while inflection rules are used to complete words.

In Semitic language stems devoid of inflections can be formed from roots. For instance, /katab-/is formed from /ktb/. So, what is observed in inflectional languages such as Latin and Italian seem to comply with inflectional languages such as Tigrigna or Amharic (Tesfai, 1993). For instance ፈረድ /fered-/, that is derived from the root /frd/, will be complete when it is inflected (adding e at the end) and becomes ፈረደ /ferede/ "he judged". Semitic languages form words by derivation and compounding but that is not all. Semitic languages have word formation rules that are quite different from those of English, Italian etc. that are formed by affixation and compounding. This is because the most frequent way of word formation rules in Semitic languages is from roots (McCarthy, 1982). The common grammatical system of Semitic languages such as Arabic (Al-Kharashi et al., 1994), Tigrigna (Tesfai, 1993), and Amharic (Nega, 1999) is based on root-pattern and are morphologically complex compared to English. A root consists of two to four consonants conveys the basic semantic meaning. A vowel pattern marks information about voice and aspect.

Reduplication is a broader case of affixation. The form of the affix is a function of the stem to which it is attached, i.e., it copies (some portion of) the stem. Reduplication may be complete or partial. In the latter case it may be prefixal, infixial or suffixal. Reduplication can include phonological alteration on the copy or the original. In Tigrigna prefixal, መመጽሓፍና 'each of our book' and infixal, ተቀታተሉ 'killed each other' are common. Infixal reduplication is used for marking frequentative and reciprocal actions.

In this section a brief introduction on the morphology of the language is given. For the purpose of the thesis, the focus is on inflectional and derivational affixes of the language. Further and detail description of the language and its characteristics are given by Tesfai (1993), Asmeret (1983), Girmay (1991) and Tsegaye (1987).

3.2 TIGRIGNA SOUNDS AND ALPHABETS

Tigrigna has 29 consonants and 7 vowel phonemes (Girmay, 1991) and there are 244 alphabets with all the sounds and symbols. With the exception of two, all of the consonant alphabets have seven forms created by combining with vowels. The seven vowel phonemes of Tigrigna are grouped as front, central and back.

The word units of Tigrigna are: phoneme, morpheme, root, stem and word. A phoneme represents a basic sound or unit of sound. A Tigrigna root is a sequence of consonants usually called radicals, and is the basis for the derivation of words; a stem, on the other hand, is a consonant or consonant-vowel sequence. A stem can be free or bound: a free stem can stand as a word on its own while a bound stem has a bound morpheme affixed to it. A collection of phonemes or sounds creates a word, which can be as simple as a single morpheme or contain several of them.

The three major lexical categories in Tigrigna are nouns, verbs and adjectives (Tesfai, 1993). Each lexical item is inflected to express the type and situation. In Tigrigna inflecting lexicals for person, gender, number, and case is a common process.

3.3 TIGRGNA AFFIXES

An affix is a bound morph that is realized as a sequence of phonemes. The common types of affixes are prefixes and suffixes. Many languages have only these two types of affixes (Trost, 1993). English is among them.

A prefix is an affix that is attached in front of a stem. An example is the English negative marker un-attached to adjectives:

common uncommon

A suffix is an affix that is attached after a stem. Take, e.g., the English plural marker —s:

shoe shoes

In addition to prefix and suffix Tigrigna uses prefix-suffix pair and reduplication (see Section 3.3.3).

3.3.1 Inflectional affixes

Inflection is required in particular syntactic contexts. It does not change the part-of-speech category but the grammatical function. The different forms of a word produced by inflection form its paradigm.

Inflection is complete, i.e., with rare exceptions all the forms of its paradigm exist for a specific word. Regarding inflection, words can be categorized in three classes:

- Particles or not-inflecting words: they occur in just one form. In English, prepositions, adverbs, conjunctions and articles are particles;
- Verbs or words following conjugation;
- Nominals or words following declination, i.e., nouns, adjectives, and pronouns.

Conjugation is mainly concerned with defining tense and aspect and agreement features like person and number. Declination marks various agreement features like number (singular, plural, dual, etc.), case (as governed by verbs and prepositions, or to mark various kinds of semantic relations), gender (male, female, neuter), and comparison.

Nouns

Nouns in Tigrigna are inflected for number and gender and indicated by inflectional markers suffixed to the noun stem. There are two grammatical genders: masculine and feminine. In general nouns are masculine or feminine by nature. For example ሰብኣይ, ወዲ and ብዕራይ are masculine and ሰበይቲ, ጓል and ላሕሚ are feminine. There are few nouns that are masculine and become feminine by adding suffixes, such as ሓው "brother" ሓውቲ "sister", ንጉስ "king" ንግስቲ "queen" and ሓሙ "father-in law", ሓማት"mother-in law".

Numbers are expressed in singular or plural form. Pluralization is formed in two ways: external plural formation and internal (broken) plural formation (Asmeret, 1983). In external plural formation only suffix is added to the singular form. The common noun pluralizing suffixes are ኣት/-at/, ታት/-tat/, ቲ/-ti/, ኦት/-ot/, ኡት/-ut/, ኣን /-an/ and ኣውያን/-awyan/. To show how these suffixes are used, examples are given for ታት/-tat/ and ኣት/-at/ in (3.1) and (3.2).

(3.1)

Plural nouns having suffix ታት/-tat/

This suffix is usually used with nouns having ending letter of 1st, 2nd, 4th, 5th, and 7th order.

Singular		Plural		Gloss (plural)
ከበሮ	kebero	ከበሮታት	kebero-tat	Drums

ዓለባ	~aleba	ዓለባታት	~aleba-tat	Cloths
ሓሙ	hamu	ሓሙታት	hamu-tat	Mother-in-laws
ቅዳሴ	qdasE	ቅዳሴታት	qdasE-tat	Prayers

When ታት/-tat/ is added to nouns ending with 3rd order letter, it changes the letter to 6th order.

Singular		Plural		Gloss
ብርዒ	br?i	ብርዕታት	br?-tat	Pens
ሰልፊ	selfi	ሰልፍታት	self-tat	Demonstrations
ምኽሪ	skri	ምኽርታት	mkr-tat	Advices
ዓንዲ	~andi	ዓንድታት	~and-tat	Pillars

(3.2)

Plural nouns having suffix ኣት/-at/

This suffix is usually used with nouns having 6th order ending letter.

Singular		Plural		Gloss
ሰማይ	semay	ሰማያት	semay-at	Skies
ሓሶት	Hasot	ሓሶታት	Hasot-at	Lies
ሕማም	Hmam	ሕማማት	Hmam-at	Diseases

The second pluralization form is by employing internal broken. In this form the plural usually has different vowel and syllabic structure. It can also include affixes (prefix and suffix). As discussed by Asmeret(1983), there are eleven different classes of broken plurals based on the syllabic structure. Illustration examples on these classes are given from (3.3)-(3.13) below.

(3.3)

CVC1C2VC → CeC-a-CVC

Singular		Plural		Gloss
መንዲል	mendil	መናዲል	men-a-dil	Handkerchiefs
ሳንዱቅ	sanduK	ሰናዱቅ	sen-a-duK	Boxes
ድንግል	dngl	ደናግል	den-a-gl	Nuns

In this structure the plural infix is /-a-/

(3.4)

CeCeC → prefix-C1C2aC

Singular		Plural		Gloss
ገመል	gemel	ኣ–ግማል	?a-gmal	Camels
ዘመድ	zemed	ኣ–ዝማድ	?a-zmad	Relatives

The prefix /?a-/ is employed as a plural marker.

(3.5)

CC1C2i →prefix-C1C2aC

Singular		Plural		Gloss
ህዝቢ	HZbi	ኣ–ህዛብ	?a-Hz-a-b	Societies
እምኒ	?mni	ኣ–እማን	?a-?m-a-n	Stones

The plural marker is /?a-a-/

(3.6)

CVC → prefix-C1C2ac

Singular		Plural		Gloss
ዑፍ	~uf	ኣ–ዕዋፍ	?a-~w-a-f	Birds
ቤት	bet	ኣ–ብያት	?a-by-a-t	Houses

In the plural formation of biradical nouns a labio-velar sem-vowel 'w' replaces the back vowel and a palatal sem-vowel 'y' replaces the front vowel.

(3.7)

CVC1C2i/CVCVC → prefix-CaCC

Singular		Plural		Gloss
በትሪ	betri	ኣ–ባትር	?a-batr	sticks
ዝብኢ	zb?i	ኣ–ዛብእ	?a-zab?	hyenas

The plural marker is /?a-/

(3.8)

CVC1C2i/CVC1C2VC → prefix-C1C2UC

Singular		Plural		Gloss
ዓንዲ	~andi	ኣ–ዕኑድ	?a-~nud	pillars
ብዕራይ	b~ray	ኣ–ባዑር	?a-ba~ur	cows

The plural marker is /?a-u-/

(3.9)

CVC1C2i → prefix-C1C2C-C-i

Singular		Plural		Gloss
ቀርኒ	kerni	አ–ቀርንቲ	?a-krn-ti	horns
ዐጽሚ	~eSmi	አ–ዕጽምቲ	?a-~Sm-ti	bones
ከልቢ	kelbi	አ–ክልብቲ	?a-klb-ti	dogs

The plural marker is /?a-ti/

(3.10)

CVC1C2V→ CaC-a-C-suffix

Singular		Plural		Gloss
ባርያ	barya	ባራዩ	bar-a-y-u	slaves
ዕትሮ	~tro	ዓታሩ	~at-a-r-u	pots
ማዕጸ	ma~So	ማዓጹ	ma-~-a-S-u	doors

The plural marker is /-a-u/

(3.11)

CeC1C2VC → CeC-a-CiC-C-i

Singular		Plural		Gloss
መልኣክ	mel?ak	መላእክቲ	mel-a-?k-ti	angels
መንፈስ	menfes	መናፍስቲ	men-a-fs-ti	sprits

The plural marker is /-a-t-/

(3.12)

CVCVC →CVCa-suffix

Singular		Plural		Gloss
ሓማት	Hamat	ሓማውቲ	Hama-wti	mother-in-laws
ዕየት	~yet	ዕያውቲ	~ya-wti	lambs

The plural marker is the suffix /-wti/

(3.13)

CVCVC →CVCa-suffix

Singular		Plural		Gloss
ሓጺን	HaSin	ሓጻውንቲ	HaSa-wnti	irons
ክዳን	kdan	ክዳውንቲ	kda-wnti	clothes
ሕጻን	HSan	ሕጻውንቲ	Hsa-wnti	children

The plural marker is the suffix /-wnti/

There are very few nouns, which changed to other word form up on pluralization. One such example is ሰበይቲ "woman" አንስቲ "women". But irregular form plurals are not common in Tigrigna.

Pronominal suffixes

In Tigrigna as in other Semitic languages the nominal way of expressing someone's possession of something is to attach an abbreviation of the pronoun "his", "her" etc. to the thing possessed, e.g ከልቢ(kelbi) "dog", ከልብና(kelbna) "our dog". Thus a noun can have many suffixes, a different one for each person, gender and number. Mathewos (1951 EC) described three ways of suffixing pronominals depending on the form (order) of the end letter.

The first group for ending with 3rd order, the second group with 6th order and the third group with ending orders other than first and second group. For example for the word geza "house" see (3.14) how the suffixes are employed.

(3.14)

ገዛኡ	geza?u "his house"	3sm	ገዛኦም	"their house"	geza?om	3pm
ገዛኣ	geza?a "her house"	3sf	ገዛአን	"their house"	geza?en	3pf
ገዛኻ	gezaKa "your house"	2sm	ገዛኹም	"your house"	gezakum	2pm
ገዛኺ	gezaKi "your house"	2sf	ገዛክን	"your house"	gezakn	2pf
ገዛይ	gezay "my house"	1ps	ገዛና	"our house"	gezana	1pp

Adjectives

As explained by Bender (1976) Tigrigna adjectives agree with their noun in gender and number. In plural form they do not distinguish gender. Suffixes are used to change adjectives from masculine to feminine form. Mathewos (1951 EC) and Asmeret(1983) have identified ት /-t/, ቲ /-ti/ and ኣዊት /-awit/ as feminine markers. The suffix ኣዊ/-awi/ is used as masculine marker for names of country or place. In pluralizing adjectives suffixes such as ቲ/-ti/, ኦት/-ot/, ኣት/-at/, ታት/-tat/ and ኣውያን/-awyan/ are used. Illustration of how these suffixes are used in marking gender and number are given from (3.15) to (3.22).

(3.15)

Feminine adjectives having suffix /-ti/

Masculine		Feminine		Gloss
ፅቡቕ	SbuQ	ፅብቕቲ	SbQti	Handsome/ Beauty
ቅዱስ	qdus	ቅድስቲ	qdsti	Blessed

ንጉስ	ngus	ንግስቲ	ngsti	King /Queen
ክፉእ	kfu?	ክፍእቲ	kf?ti	Bad

The suffix /-ti/ is usually added to adjectives of ..$C_{k-1}uC_k$ form and deletes the back vowel /u/.

(3.16)

Feminine adjectives having suffix /-t/

Masculine		Feminine		Gloss
ቀታሊ	ketali	ቀታሊት	ketalit	Killer
በላአዊ	bela~i	በላዊት	bela~it	Eater
መራሒ	meraHi	መራሒት	meraHit	Leader
ተቐባሊ	teQebali	ተቐባሊት	teQebalit	Receiver

The suffix /-t/ is added to adjectives of. .$C_{k-1}aC_ki$ form.

(3.17)

The suffix /-awi/ and /-awit/

Masculine		Feminine		Gloss
ካናዳዊ	kanadawi	ካናዳዊት	kanadawit	A Canadian
አሜሪካዊ	?emerikawi	አሜሪካዊት	?emerikawit	An American
ኢትዮጵያዊ	?tyopyawi	ኢትዮጵያዊት	?tyopyawit	Ethiopian
ግብጻዊ	gbSawi	ግብጻዊት	gbSawit	Egyptian

(3.18)

Plural adjectives having the suffix /-awyan/

Singular				Plural		Gloss
Masculine		Feminine				
ካናዳዊ	kanadawi	ካናዳዊት	kanadawit	ካናዳውያን	kanadawyan	Canadians
አሜሪካዊ	?emerikawi	አሜሪካዊት	?emerikawit	አሜሪካውያን	?emerikawyan	Americans
ኢትዮጵያዊ	?tyoPyawi	ኢትዮጵያዊት	?tyoPyawit	ኢትዮጵያውያን	?tyoPyawyan	Ethiopians
ግብጻዊ	gbSawi	ግብጻዊት	gbSawit	ግብጻውያን	gbSawyan	Egyptian

(3.19)

Plural adjectives having suffix /-ti/

Singular				Plural		Gloss
Masculine		Feminine				
መጺጽ	meSiS	መጻጽ	meSaS	መጸጽቲ	meSeSti	sour (pl.)
ጸቢብ	Sebib	ጸባብ	Sebab	ጸበብቲ	Sebebti	narrow (pl.)
ጸሊም	Selim	ጸላም	Selam	ጸለምቲ	Selemti	black (pl.)
ቀይሕ	qeyH	ቀያሕ	qeyaH	ቀየሕቲ	qeyeHti	red (pl.)

(3.20)

Plural adjectives having suffix /-ot/

Singular				Plural		Gloss
Masculine		Feminine				
ቀዳማይ	qedamay	ቀዳመይቲ	qedameyti	ቀዳሞት	qedamot	first
ሕያዋይ	Hyaway	ሕያወይቲ	Hyaweyti	ሕያዎት	Hyawot	kind

ትግራዋይ	tgraway	ትግራወይቲ	tgraweyti	ትግራዎት	tgrawot	Tigrians
ዳሕራዋይ	daHraway	ዳሕራወይቲ	daHraweyti	ዳሕራዎት	daHrawot	last

(3.21)

Plural adjectives having suffix /-at/

Singular				Plural		Gloss
Masculine		Feminine				
ጽቡቕ	SbuQ	ጽብቕቲ	SbQti	ጽቡቓት	SbuQat	nice
ቅዱስ	qdus	ቅድስቲ	qdsti	ቅዱሳት	qdusat	holy
ርሑቕ	rhuQ	ርሕቕቲ	rhQti	ርሑቓት	rhuQt	far
ጽሩይ	Sruy	ጽርይቲ	Sryti	ጽሩያት	Sruyat	clean

(3.22)

Plural adjectives having suffix /-tat/

Singular		Plural		Gloss
ዓመጸኛ	~ameSeNa	ዓመጸኛታት	~ameSeNatat	"one who uses force"
ድኻ	dKa	ድኻታት	dKatat	Poor

There are few adjectives which takes the suffix /-u/ to change them to plural form. These words don't distinguish gender in their singular form. See (3.23) for illustration.

(3.23)

Masculine		Feminine		Gloss (plural)
ጻዕዳ	Sa~da	ጻዓዱ	Sa~adu	white
ሌባ	lEba	ሌያቡ	lEyabu	thieves
ዓሻ	~axa	ዓያሹ	~ayaxu	fools

Verbs

Tigrigna consists of biradical, triradical, and quadriradcal verbs with two, three and four consonants respectively.

As described by Tesfai (1993), biradical verbs are rare in Tigrigna and they are basically triradical which have original labio-velar semivowel 'w', palatal semi-vowel 'y' or a laryngeal consonant 'H' as their medial and the loss of 'w', 'y', or 'H' may result in biradical. Examples of biradical verbs are ከደ kede "he has gone", ሞተ mote "he died", በለ bele "he said".

Triradical verbs have sub-class Type A, Type B, and Type C. Type A verbs have non-geminated 2nd radical except in the imperfect aspect singular form.

eg. ጸረፈ Serefe "he insulted"

ደረፈ derefe "he sang"

ፈረደ ferede "he judged"

Type B verbs geminate the second radical.

eg. ዘከረ zekkere "he remembered"

ለመነ lemmene "he begged"

ሸመተ xemmete "he purchased"

Type C verbs have the vowel /a/ after the first radical and the second radical is a non-geminate.

eg. ናፈቐ nafeQe "he longed to see some one"

ማረኸ mareKe "he surrendered someone"

ላጸየ laSeye "he shaved"

Quadriradical verbs consist of four radicals.

eg.	መሰከረ	meskere	"he witnessed"
	ሰልጠነ	selTene	"he became civilized"
	መዝገበ	mezgebe	"he registered"

Verb inflection

Tigrigna verbs are conjugated in perfective, imperfective, gerundive, jussive and imperative. In conjugating the verbs affixes are employed.

Perfective

The simple perfective verb is inflected by suffixing person, gender and number morphemes to the perfect verb stem. Example (3.24) illustrates how these suffixes are employed.

(3.24)

ነገርኩ	neger-ku	' I told'	1ps
ነገርካ	neger-ka	'You told'	2sm
ነገርኪ	neger-ki	'You told'	2sf
ነገረ	neger-e	'He told'	3sm
ነገረት	neger-et	'She told'	3sf
ነገርና	neger-na	'We told'	1pp
ነገርኩም	neger-kum	'You told'	2pm
ነገርክን	neger-kn	'You told'	2pf
ነገሩ	neger-u	'They told'	3pm
ነገራ	neger-a	'They told'	3pf

The morphemes employed are ኩ /-ku/, ካ /-ka/, ኪ /-ki/, እ/-e/, እት /-et/, ና /-na/, ኩም /-kum/, ክን /-kn/, ኡ /-u/, and ኣ/-a/. Theses suffixes are used without any variation with verbs of Type A, B, C and quadriradicals. Actually there are no suffixes employed as perfect aspect marker. The vowels in the verb stem are the perfect aspect markers.

Imperfective

The simple imperfective verb is inflected by prefixing gender, person, and number morphemes to the imperfective verb stem. Example (3.25) illustrates how these suffixes are employed.

(3.25)

እነግር	?-negr	‘ I will tell’	1ps
ትነግር	t-negr	‘You will tell’	2sm
ትነግሪ	t-negr-i	‘You will tell’	2sf
ይነግር	y-negr	‘He will tell’	3sm
ትነግር	t-negr	‘She will tell’	3sf
ንነግር	n-negr	‘We will tell’	1pp
ትነግሩ	t-negr-u	‘You will tell’	2pm
ትነግራ	t-negr-a	‘You will tell’	2pf
ይነግሩ	y-negr-u	‘They will tell’	3pm
ይነግራ	y-negr-a	‘They will tell’	3pf

The morphemes employed are እ /?-/, ት /t-/, ት-ኢ/t-i/, ይ /y-/, ን /n-/, ት-ኡ/t-u/, ት-ኣ /t-a/, ይኡ /y-u/, and ይ-ኣ /y-a/. Just like the simple perfective, there is no prefix or suffix

employed as imperfective marker. The vowels in the verb stem are the imperfect aspect markers.

Gerund

The gerundive form is inflected by suffixing person, gender and number morphemes to the gerundive verb system.

(3.26)

ነገረ	negir-e	' I have told'	1ps
ነገርካ	negir-ka	'You have told	2sm
ነገርኪ	negir-ki	'You have told	2sf
ነገሩ	negir-u	'He has told'	3sm
ነገራ	negir-a	'She has told'	3sf
ነገርና	negir-na	'We have told'	1pp
ነገርኩም	negir-kum	'You have told'	2pm
ነገርክን	negir-kn	'You have told'	2pf
ነገሮም	negir-om	'They have told'	3pm
ነገረን	negir-en	'They have told'	3pf

Mood

Mood consists of the jussive and the imperative. The jussive expresses a command, but with less emphatic effect than the imperative mood, for 1st and 3rd persons. The imperative is used to express a command for the 2nd person in the singular and plural form. Verbs in the jussive form take prefixes or suffixes and in the imperative form they may or may not take suffixes. Example (3.27) illustrates how they are used.

(3.27)

ክነግር	k-negir	'let me tell'	1ps
ይንገር	y-nger	'let him tell	3sm
ትንገር	t-nger	'let her tell'	3sf
ንንጊር	n-nger	'let us tell'	1pp
ይንገሩ	y-nger-u	'let them tell'	3pm
ይንገራ	y-nger-a	'let them tell'	3pf

3.3.2 Derivational affixes

In contrast to inflection, which produces different forms of the same word, derivation and compounding are processes that create new words. In derivation, a different word often of, a different part of speech category, is produced by adding a bound morph to a stem. Derivation is incomplete, i.e., a derivational morph cannot be applied to all words of the appropriate class.

Noun derivation

In Tigrigna there are verbal nouns, adjectival nouns and nouns which designate abstractions. Verbal nouns which are infinitive, agentive and instrumental are derived from verbs by using affixes /m-/, /-i/, /-it/, and /me-i/ respectively. Example (3.28) illustrate how the affixes are employed.

(3.28)

Verb	**Gloss**		**Infinitive**	**Gloss**
ሰበረ sebere	'he broke'	ምስባር	m-sbar	to break

ቀተለ	qetele	'he killed'	ምቅታል	m-qtal	to kill
ሓረሰ	Harese	'he ploughed'	ምሕራስ	m-Hras	to plough

Verb		Gloss		Agentive	Gloss
ሰበረ	sebere	'he broke'	ሰባሪ	sebar-i	breaker
ቀተለ	qetele	'he killed'	ቀታሊ	ketal-i	killer
ሓረሰ	Harese	'he ploughed'	ሓራሲ	Haras-i	one who ploughs

Verb		Gloss		Instrumental	Gloss
ሰበረ	sebere	'he broke'	መስበሪ	me-sber-i	used for breaking
ቀተለ	qetele	'he killed'	መቅተሊ	me-qtel-i	used for killing
ሓረሰ	Harese	'he ploughed'	መሕረሲ	me-Hres-i	used for ploughing

Abstract nouns are derived from adjectives by suffixing /-i/, /-at/, /-et/ /-na/, /-eyna/, /-ay/ and /-net/. The suffix /-net/ is also used to derive abstract nouns from nouns and used to show of being something. See example (3.29) for illustration.

(3.29)

Adjective		Gloss		Abstract noun	Gloss
ቀጢን	qeTin	'thin'	ቅጥነት	QT-net	to be thin
ንፉእ	nifu?	'brave'	ንፍኣት	nf?-at	to be brave
ጎይታ	goyta	'lord'	ጎይትነት	goyt-net	to be lord

Noun		Gloss		Abstract noun	Gloss
ጅግና	jgna	'hero'	ጅግንነት	jgn-net	being hero

ቀሺ	qexi	'priest'	ቅስና	qs-na	being priest
ዘመድ	zemed	'relative'	ዝምድና	zmd-na	relationship
ፈረስ	feres	'horse'	ፈረሰይና	feres-yna	horse man
ተምቤን	tembEn	'name of a place'	ተምቤናይ	tembEn-ay	one from tembEn

Nouns can be derived from verbs using suffixes /-ay/ and /-tay/. See (3.30) for illustration.

(3.30)

Verb		**Gloss**		**Noun**	**Gloss**
ነደቀ	nedeqe	he built	ነዳቃይ	nedaqay	constructor
ሓረሰ	Harese	he plough	ሓረስታይ	Harestay	farmer
ጸሓፈ	Sehafe	he wrote	ጸሓፍይ	Sahafay	secretary

Adjective derivation

Adjectives can be derived from verbs and nouns through suffixes, prefixes or both. (3.31) Illustrates the situation.

(3.31)

Verb		Adjective	
ሰበረ	sebere "he broke"	ሰባሪ	sebar-i "one (masculine) who breaks"
በደለ	bedele "he committed crime"	በደለኛ	bedele-Na "one who commits crime"

Noun		Adjective	
ሓምሊ	Hamli "vegetable"	ሓምላይ	Haml-ay "green"

ኢትዮጵያ ?ityoPya "Ethiopia" ኢትዮጵያዊ ?ityoPy-awi "Ethiopian"

ሕንዚ Hnzi "poison" ሕንዛም Hnz-am "poisonous"

ዓፋር ~afar "a place in Ethiopia" ዓፋርኛ ~afar-Na "a language spoken by Afar people"

Tesfai (1993) noted that, the inputs of the suffix /-am/ ends in r, z, d and s. So the words /gud/, /merz/, /wez-/ can be inputs.

Verb derivation

Verbs can be derived from the active form of the verb by prefixing various elements. The derived forms can express different modes of actions such as passive, causative, reciprocal, causative-reciprocal and frequentative.

According to Tesfai(1993), generally there are two derivational prefixes /te-/ and /?a-/. The bases of /te-/ are perfective pattern /Ce(a)C(C)eC/ and the traditionally called gerundive pattern /Ce(a)C(C)iC/. The bases of /?a-/ are similar to the bases of /te-/. Example (3.32) demonstrates how the affixes are employed.

(3.32)

Active	**Gloss**
ረገመ regeme	"he cursed"
ሓገዘ Haggeze	"he helped"
ማለደ malede	"he meditated"
መዝገበ mezgebe	"he registered"

Passive		Gloss
ተረገመ	te-regeme	"he was cursed"
ተሓገዘ	te-Haggeze	"he was helped"
ተማለደ	te-malede	"he was meditated"
ተመዝገበ	te-mezgebe	"he was registered"

Reciprocal		Gloss
ተረጋገሙ	te-regagemu	"they cursed each other"
ተሓጋገዙ	te-Hagagezu	"they helped each other"
ተማላለዱ	te-malaledu	"the meditated each other"
ተማዛገቡ	te-mazagebu	"the registered each other"

Causative		Gloss
አርገመ	?a-rgeme	"he made some one cursed"
አሓገዘ	?a-Haggeze	"he made some one helped"
አማለደ	?a-malede	"he made some one meditated"
አመዝገበ	?a-mezgebe	"he made some one registered"

Causative Reciprocal		Gloss
አራገመ	?a-rageme	"he made others cursed each other"
አታሓጋገዘ	?ata-Hagageze	"he made others helped each other"
አማለደ	?a-malede	"he made others meditated each other"
አማዛገበ	?a-mazagebe	"he made others registered each other"

The derivational prefixes are:

ተ	/te-/	passive
ኣ	/ʔa-/	causative
ተ	/te-/ and reduplication of 2nd radicals	reciprocal
ኣት	/ʔat-/	causative reciprocal

Type A, B, and C verbs employ reduplication of the 2nd radicals in addition to the prefix /te-/ in the reciprocal form. In the quadriradical verbs, the reciprocal is formed by discontinues morpheme /te-a-/.

3.3.3 Reduplication

In addition to affixation and pattern change (insertion and deletion of vowels and/or consonants), Tigrigna uses reduplication. Reduplication is repeating part of a word/stem. For example in the word መመጽሓፍኩም፡ ገገንዘብናand ሰባበረ the letters መ፡ገ and በ are repeated. But if we see words such as ስብርብር፡ ገልጠምጠም and ዕንትርትር the repeating part consists of two letters which is ብር፡ ጠም and ትር.

In Tigrigna reduplication is used for marking actions such as repetitive, reciprocal, uncompleted, and ordered (one after the other) (Mathewos, 1951). To mark those actions the reduplication may come prefixal (ገገንዘብና) or infixal (ተረጋገሙ). The number of letters to be reduplicated may be one (መመጽሓፍኩም) or two (ስብርብር). See (3.33) for examples on reduplications.

(3.33)

Prefixal

መረት	merEt	መመረትኩም	memerEtkum
ደቅኽን	deqKn	ደደቅኽን	dedeqKn
ገዛኹም	gezaKum	ገገዛኹም	gegezaKum

Infixal

ቀተለ	qetele	ቀታተለ	qetatele
ላጺኹም	laSuKum	ላጻጺኹም	leSaSiKum
ጸሃፈ	Sehafe	ኣጻሃህፉ	?aSahahfa

3.4 Compounding

As cited by Tesfai(1993), compounding is the result of two or more words/stems combining into a single morphological unit. Tigrigna compounds are of two type; strict (lexicalized) and loosed. In loose compounds, the meaning of the whole compound is the meaning of the head qualified by the non-head member. For example መርፍኣረቂ merf~?areqi means "small sized needle".

On the other hand, in strict compound, since they do not have heads, the meaning of these words is not predictable from the constituent members of the compound. For example, one can not predict the meanings of ዓይኒምድሪ ~aynimdri "toilet" from the isolated meanings of the words ዓይኒ ~ayni "eye" and ምድሪ mdri "earth". This shows the words are amalgamated. Because of the complexity of compounding in Tigrigna especially the strict

ones and the time constraint, the stemming algorithm that will be developed will not handled compounding.

CHAPTER 4

STEMMING ALGORITHM FOR TIGRIGNA

4.1 INTRODUCTION

A review on the morphology of Tigrigna language has been presented in the preceding chapter. It has been shown that main word formation process in Tigrigna is done through affixation. The main classes of affix are: prefix, suffix, prefix-suffix pair and reduplication (single and double). Those affixes are used for inflecting and deriving words. Nouns and adjectives are inflected for gender, number and person. Verbs are inflected for gender, number, person, aspect and tense. Tigrigna uses extensive concatenation of affixes and resulted in a relatively long word that can represent semantic meaning of a phrase or sentence in English. As a result of those morphological structures of the language, a word can have thousand variants. In designing retrieval systems for the language, reducing these variants into one form, improves performance of the system. This can be achieved by a conflation technique, which is usually stemming. This chapter presents development of a stemming algorithm for the language. The compilation of stopwords and affixes and evaluation of the stemmer are also presented.

4.2 PURPOSE

This experiment is concerned with developing a stemming algorithm to conflate variant words in Tigrigna text documents and testing performance of the stemmer on sample data collection. The main tasks of the experiment are selecting sample data, compiling affixes and stopwords, developing the stemmer, and testing and evaluating the stemmer on the sample data. The detail discussions of each activity are presented in the subsequent sections.

4.3 TEST DATA

To experiment the algorithm developed, sample texts were prepared from three different sources: WOYN newspaper, ASER magazine and from a fiction titled STRUGGLE. These sources were selected in due consideration of their content (social, cultural and political), which is believed to represent the language, availability of the documents and standard of word use especially in the newspaper WOYN. The selection of topics and chapters from each sample texts was done randomly.

An attempt was made to create the test collection from the electronic versions of the sample texts that are in Ethiopic form by converting to text of Latin alphabets. When the Ethiopic ms-word document is saved in text type format (tfile), it consists of special characters, which are ASCII represents of the document. For the purpose of the experiment, a C++ program was written that accepts the text type format file (tfile) and produces text file. However, inspection of the result revealed that the conversion was not completely successful. The problems were: missed letters (eg. H, ri) and the same character representation for different letters (eg. ru, mE, Te were represented by the same character). These errors were the result of non-printable

special characters in text editor and some of the special characters were replaced by another character.

Because of the problems mentioned above and limitation on time, another approach was used to prepare test collection. In this approach, the sample texts were collected from hard copy and were typed using Latin alphabets. If the first approach were successful, it would have helped in experimenting the research. Not only this, but the procedure would have served for any one who is interested in doing experiments on Ethiopic documents. The size of the sample texts in terms of words, number of distinct words and word ratio is given in Table 4.1.

Table. 4.1 Number of Words

Name of text	Total words	Distinct words	Ratio of total words to distinct words	% of words with frequency of 1	% of words with frequency of 10 & above
WOYN	2542	1179	2.156	34.62	1.69
STRUGGLE	2997	1502	1.995	38	1.23
ASER	1491	739	2.017	36.61	1.27

4.4 WORD DISTRIBUTION OF TIGRIGNA TEXTS

It has been cited in many works (Hmeidi et al., 1997; Ekmekçioglu et, al., 1996; Nega, 1999) that, word-distribution in text documents of a language helps in studying behavior of the language. One way of measuring word-distribution is using word-ratio (total number of

words to distinct words). The word-ratio obtained for the sample texts is given in Table 4.1. The values are 2.156, 1.995 and 2.017 for WOYN, STRUGGLE and ASER respectively.

A text size of 1632 was considered from the text STRUGGLE to compare the word ratio with English and Arabic (adapted from Hmeidi et al, 1997). The ratio obtained for Tigrigna text indicated in Table 4.2 was almost similar with the Arabic text. The similarity in the ratio might be explained by the fact that both languages are Semitic.

Table 4.2 Comparison of Word Ratios of Tigrigna with English and Arabic

Language	Text	Length of Text	Distinct Words	Word Ratio
Tigrigna	STRUGGLE	1,632	918	1.777
English	Text 1	1,600	621	2.576
Arabic	Text 1	1,600	902	1.774

Based on frequency value, percentage of single words was calculated and the result is shown in Table 4.1. From the table we can see that singleton words constitute large portion in the documents. This shows existence of more variant words in Tigrigna documents.

Hmeidi et al (1997) argue that, the language with lesser values of the word ratio corresponds to more distinct words in a text and vise versa. This implies that a particular word appears less often for Tigrigna than for English. The variance of words and the word ratio (Hemeidi et al., 1997; Alkharashi et al., 1996) may be indications of the complexity of the morphology of the language. As such we can observe that morphology of Tigrigna is more complex than that of English language.

In addition to word ratio, the Zipfians law could also be used as an indication of the complexity in the morphology of a language. This law is based on frequency of words and given by:

$$f*r=k$$

Where f is the frequency of a word in the text, r is rank of the word when the words are listed down from the highest frequency to the lowest one, and k is a constant value. What the law says is that, the product of the frequency and the rank gives us constant value for all the words of a text.

The Zipf's constant for some selected ranks of the sample texts used in this experiment is given in APPENDIX I. According to Nega(1999) many languages do not obey the law. For instance, on his work on Amharic language he showed the deviation of Amharic documents from the law. This may also be shared by Tigrigna documents as can be seen in Table 4.3. The unevenly distribution of words in Tigrigna documents and the deviation of the language from Zipf's law may be an indication of the complexity in morphology of the language.

Table 4.3 The Zipf's constant for selected ranks of WOYN text

Word	f	r	f*r
?ab	67	1	67
?wn	25	10	250
sraH	16	20	320
?ayte	11	30	330

kllna	10	40	400
?zom	9	50	450
TeQlala	7	60	420
?aKEba	6	70	420
zelo	5	80	400
bSay	4	90	360
bmKanu	4	100	400
menbernet	2	200	400
mSrarom	1	300	300
bbQ~at	1	400	400
feSaminetu	1	500	500
mergeStat	1	600	600
zeytwaSa?	1	700	700
tegbarat	1	800	800
hwaHat	1	900	900
kadren	1	1000	1000
melsi	1	1100	1100
HSretat	1	1179	1179

4.5 COMPILATION OF STOPWORD LIST

The stopword list was compiled from the three sample texts by collecting the most frequently occurring words. Using a C++ program, frequency of words in each texts was generated. Table 4.4 lists the top 30 words from each text. As can be seen from the table, the stopword list consists of prepositions (such as አብ ?ab "in", ካብ kab "from", ምስ ms "with" and

ናብ nab "to"); demonstrative adjectives (እዚ ?zi "this" and እቲ ?ti "the(masculine singular) "); articles (እታ ?ta "the (feminine singular) and እቶም ?tom "the (masculine plural)") and conjunctions (such as ግን/ግና gn/gna "but", ድማ dma "and"). There are also non-function words (e.g bEt, mKri, guba?E). As indicated earlier, function words such as prepositions, conjunctions and articles in Tigrigna exist affixed to words. For this reason, the frequency of function words in Tigrigna does not seem to be as high as in the English language. That is, the frequency of function words in Tigrigna is low. Because of this reason, the use of frequency alone did not help to generate all the stopwords and many function words were added manually to the stopword list by consulting books and dictionaries (e.g. do, wxTi, ?zu, ?ten). The complete list of the stopword list is given in Appendix II.

Table 4.4 The first 30 words of the sample texts with high frequency

	WOYN		ASER		AGAZI		Total	
No	Word	Freq	Word	Freq	Word	Freq	Word	Freq
1	?ab	67	?yu	54	?ab	74	?ab	177
2	nay	60	qnE	53	?yu	61	?yu	147
3	?ti	46	?zi	38	meQele	48	nay	120
4	bEt	40	?ab	36	?ti	46	?ti	117
5	?zi	38	dma	29	Hayelom	44	?zi	91
6	mKri	33	nay	27	nay	33	?wn	69
7	?yu	32	?ti	25	kem	33	dma	68
8	kab	31	QanQa	16	?wn	29	bEt	65
9	guba?E	26	kab	16	nab	29	kab	64
10	?wn	25	meSHaf	16	neyru	29	qnE	53
11	dma	24	zbl	16	tegadelti	25	kem	52

12	hSuS	22	?wn	15	bEt	25	nab	49
13	medreK	19	tgrNa	15	ma?serti	24	meQele	48
14	dmSi	18	qal	14	?tom	22	Hayelom	44
15	neti	18	Hade	13	?surat	20	zbl	39
16	wdb	17	koynu	13	nti	19	mKri	33
17	koynu	17	trgum	13	Hayli	18	koynu	33
18	nayti	17	Hbre	10	kab	17	?tom	32
19	?abalat	17	Tbeb	10	dma	15	dmSi	30
20	sraH	16	nab	9	?zi	15	?abalat	30
21	zbl	15	wey	9	komandotat	14	neyru	29
22	kem	15	?wan	8	ms	14	Hade	29
23	hzbi	15	klte	8	Hade	14	guba?E	26
24	Halafnet	14	?a?mere	8	?abalat	13	tegadelti	25
25	kunetat	14	?Ka	7	dmSi	12	nayti	25
26	feSamit	13	zelewo	7	hweHat	12	ma?serti	24
27	lebewa	12	gna	7	gizE	12	?wan	24
28	tgray	12	kal?ay	7	derg	12	hSuS	22
29	menber	11	malet	7	se~at	11	mKanu	22
30	?ayte	11	meskot	7	gujle	11	hweHat	21

4.6 COMPILATION OF PREFIX

Tigrigna uses prefixes for marking prepositions (eg. ብ "with"), accusative markers (e.g ን"to") and plurals (e.g ኦ). It is also used for gender, person and number inflection, marking jussive and imperfective, and deriving nouns from verbs.

In this work, semi-automated procedure was used to produce prefix list. The sorted list of words were used to identify the prefixes. To extract the sub-strings and their frequency, a C++ program was written. After generating the sub-strings, their frequencies were compiled. Then, those sub-strings with high frequency were taken as prefix. A string of three radicals (consonants) was considered in identifying the remaining string of the word. If a sub-string is followed by vowel then the vowel is taken as part of the sub-string since words in Tigrigna do not begin with a vowel. That means, for the example given below instead of ?, ?a is considered. The most frequent leading strings identified as a result of this process are given in Table 4.5.

For example, for the word **ኣይበልናንዶ** (?aybelnando), the following sub-strings are generated

ኣ	?a
ኣይ	?ay
ኣይበ	?aybe
ኣይበል	?aybel

Table 4.5 Highly frequent leading strings

Sub-string	Frequency
?a	275
z	273
me	221
b	216

?	184
n	143
te	133
m	125
ze	89
k	88
t	80
?n	73
zte	66
?ay	60
ha	60
ke	51
y	49
h	45
?na	39
bm	32
w	30

Except three (ሃ ha, ህ h and ው w) all the sub-strings in Table 4.3 are considered as genuine prefixes (consultation of literature).

In addition to quantitative method described above, consultation was made to books and dictionaries to check the correctness of the prefixes. Additional prefixes were also included manually from the dictionary (e.g ናይ nay, ከምዚ kemzi, እንካብ ?nkab) . The complete list of prefix is given in Appendix III.

4.7 COMPILATION OF SUFFIX

A similar approach was used to generate suffixes but this time, first a word is reversed and the sub-strings are extracted. For example, to extract the sub-strings from the word ሰባበርናዮ sebabernayo, first the word is reversed and gives us oyanrebabes. Then, the same approach of prefix compilation was followed. But the sub-strings to be considered as a suffix, consideration of the following vowel is not important since a Tigrigna word can end in a vowel. Table 4.6 shows sample of the sub-strings produced as a result of the process. All the sub-strings given in the table are genuine suffixes (consultation of literature). Complete list of the suffixes is given in Appendix IV.

Table 4.6 Highly frequent ending strings

String	Frequency
i	637
u	606
t	360
n	351
e	350
a	333
m	280
at	252
om	248
o	120
ti	113

tat	54
wi	53
awi	52
ay	51
Na	50

4.8 PREFIX-SUFFIX PAIRS

In addition to prefix and suffix, Tigrigna also uses prefix-suffix pair. Some of the frequently used prefix-suffix pairs are መ-ቲ me-ti, መ-ያ me-ya, መ-ኢ me-i, መ-ታ me-ta and መ-ት me-t. Prefix-suffix pairs are usually used to derive nouns from verbs. To illustrate on the usage of prefix-suffix pairs, the following examples are given.

መቕበሪ	me-qber (bury)-i	meqberi	coffin
መንግስቲ	me-ngs (Kingship) -ti	mengsti	government
መወርወሪ	me-werwere (he threw)-ya	mewerwerya	something used for throwing
መድሓኒት	me-dehane (he has recovered)-t	medhanit	medicine
መጀመርታ	me-jemere (he has started)-ta	mejemerta	beginning

4.9 THE STEMMER

Techniques developed for English and Semitic languages such as Arabic and Amharic were studied. Some of the techniques used in these algorithms were incorporated in developing the Tigrigna stemmer. The algorithm uses iterative approach but when it finds two affixes that match with the word, the longest one is removed.

As indicated in the preceding chapter the process of inflection and derivation in Tigrigna is done through one or combination of the following processes:

- pure external affix (without modifying the stem)
- external affix and with modification of the letter (s) of the stem at the beginning, end or other positions
- pattern change (insertion or deletion of consonants and vowels)
- reduplication (prefixal and infixal)

Accordingly, theses characteristics were taken into consideration in developing the stemmer.

Of the two approaches (context free and context sensitive) discussed earlier, a context-sensitive was considered appropriate, as Tigrigna is morphologically complex language (see Chapter 3). Though Tigrigna uses external affixation like plural marker as –s in English, most of the time affixes in Tigrigna modify the stem. Therefore, a context-sensitive approach must be used to get better conflation results.

To this end, each affix is accompanied by a context-sensitive rule. The techniques for describing the rules are adopted from Porter (1980). The rule for removing an affix is given in the form:

Condition A⇒ SP|S

This means if a word has affix A and the Condition that accompanies it is true; the word will be changed to a stem with pattern SP (it could be the pattern of the whole stem or substring) or the affix is replaced by the string S which could be null (removed) or more. The Condition may be applied on the word or the stem and can also have expressions with logical operators; and, or, and not. For example, (L>n and (*s or *t)) tests for a stem with L > n and ending with s or t.

The Condition part may contain the following:

L > n	Length of the remaining stem is greater than n.
*s	The stem ends with s (and similarly for other letters).
*uC	The second from the last of the word is u (and similarly for other letters).
WP	The word has pattern WP. Pattern is described as sequences of CV or CVC because Tigrigna syllable consists of CV or CVC.
SP	The stem will have pattern SP. For example, SP can have the form *iC which means the letter before the last consonant will be changed to 'i'.

In setting minimum stem length which is denoted by n in the above expression, the number of radicals (consonants) in the word are considered. Tigrigna words are usually three to five

radicals (Tesfai, 1993). Therefore, a minimum stem length of three radicals (consonants) is considered.

In the stemmer developed, five step rules were used for the purpose of removing affixes. The purpose of each step is given as follows:

The first step takes the word to be stemmed as an input and removes double letter reduplication. For instance, ገልጠምጠም gelTemTem "messing" consists of repeated sub-string ጠም "Tem". In removing such form, first the radical (sequence of consonants) of the word was extracted and checked for repeating double sequences. In this case ግልጥምጥም glTmTm is the radical and has repeated double sequence that is ጥም "Tm". Therefore, the first sub-string which is Tem is removed from the string and leaving the word as ገልጠም gelTem.

The second step removes prefix-suffix pair. This step takes the output of the previous step as an input and checks if the word contains match with any of the prefix-suffix pair. If the word contains a match and the remaining string has a length greater than three, then the prefix and the suffix are removed from the word. For example, the word መጀመርያ mejemerya contains the prefix-suffix pair መ–ያ me-ya and the remaining string after extracting the pair is ጀመር jemer, which has length of three radicals. Therefore, the prefix and the suffix are removed from the word and gives ጀመር jemer as an output.

The third step removes prefixes. This step takes the output of prefix-suffix stripping. In removing a prefix, checking for match in the prefix list and counting length of the remaining string is done. The prefix should not be followed by a vowel (as discussed above) so checking

of this also been done. If the word satisfies all those conditions, the prefix is removed from the word.

In the fourth step, removal of suffixes is done. After accepting the output of step 4, this step checks if the word contains any match from the list of suffixes. If the word has a match and the remaining string is three the suffix is removed from the word.

The last step is used to stem reduplication of single letter. This has the same approach as step 1, but it checks for reduplication of single letter. For example, the word ሰባበረ sebabere "he broke into pieces" contains single reduplication that is babe. After removing the first reduplicated consonant together with the vowel, it gives ሰበረ sebere as an output.

Except the removal of reduplication (single and double), after each step is applied, the word is submitted to a recording step, which checks for some spelling exception. For instance, when the prefix ?ana is removed from the word ?anakese, it gives kese but the correct stem is nekese. So this step checks for spelling exception and made readjustment.

4.10 IMPLEMENTATION OF THE STEMMER

Affixes are actually removed through the process of matching the input word to the affixes in the rules. The algorithm uses iterative approach in removing concatenated affixes. The general operation of the algorithm is given in flow chart as follows:

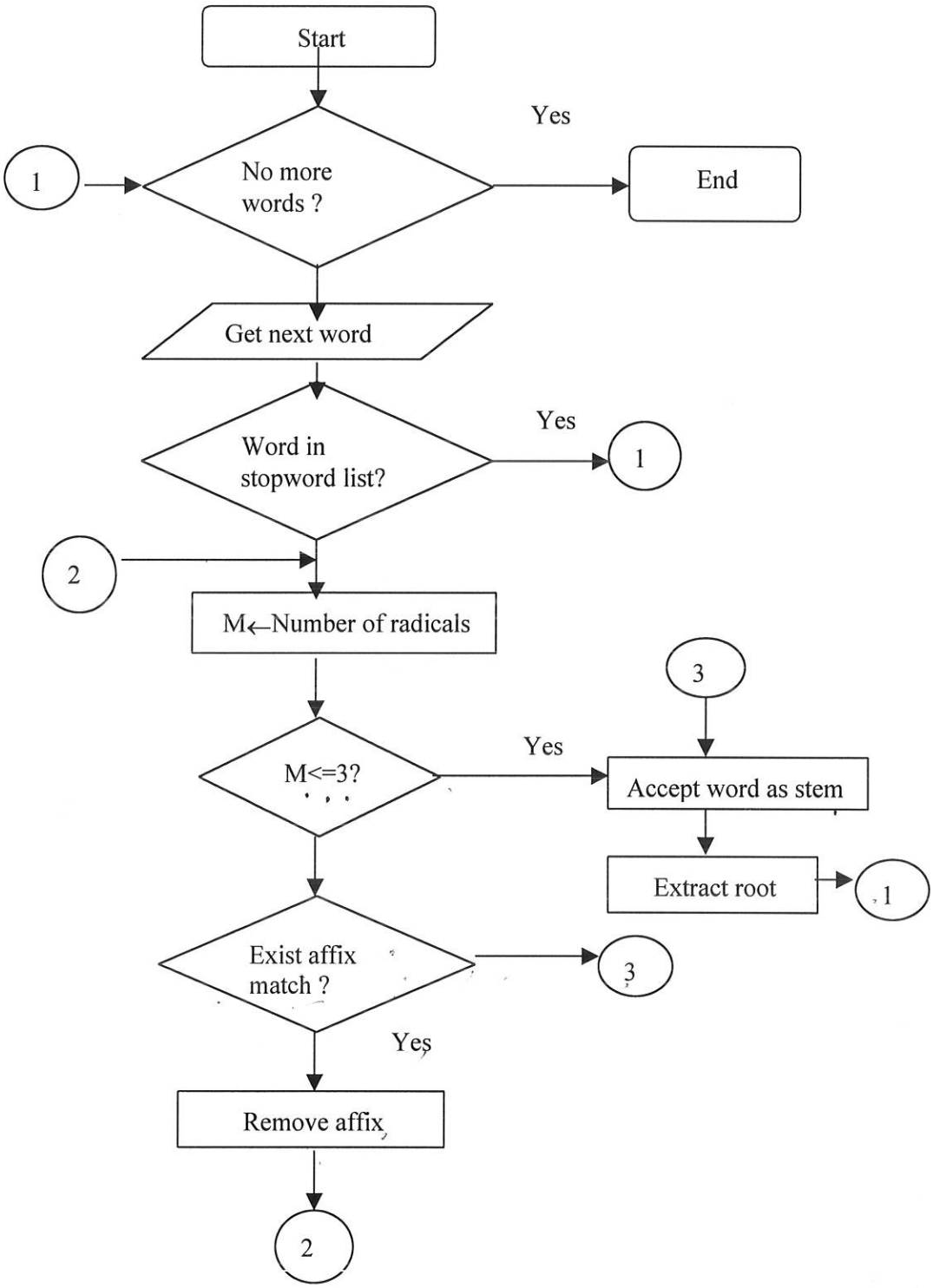


Figure 4.1 Flowchart of the Stemmer

The stemming program has six main procedures:

- Prefix-suffix stripping
- Removing double letter reduplication
- Prefix stripping
- Suffix stripping
- Removing single letter reduplication
- Extracting root

Detail description of these procedures is given below.

4.10.1 Prefix-suffix stripping

The procedure takes a word as an input. It checks the existence of both the prefix and the suffix and it removes both. For example from the word መወርወርያ (mewerwerya), the prefix /me-/ and the suffix /-ya/ are removed and the word becomes ወርወር (werwer).

1. Get WORD
2. Count number of radicals of WORD (nw)
3. **If** nw <=3 **then** stop and return WORD
4. **If** no PREFIX-SUFFIX pair **then** stop and return WORD
5. **If** PREFIX-SUFFIX pair match **then**

   Count number of radicals of PREFIX-SUFFIX (nps)

   If (nw-nps) >=3 **then**

   Remove PREFIX and copy substring to PWORD

   Remove SUFFIX from PWORD and copy substring to SWORD

   Copy SWORD to WORD

Go to Step 2

Else Go to Step 4

Else Go to Step 4

Figure 4.2 Algorithm for removing prefix-suffix pair

4.10.2 Removing double letter reduplication

This procedure is used to remove double letter reduplication. The procedure takes as an input form the pervious procedure. It checks the occurrence of the reduplication and removes the first two letters. If the letters removed have vowels then the vowels are also removed. For example, the word ገልጠምጠም gelTemTem contains reduplicated letters ጠም Tem and ጠም Tem. Therefore, the first two letters Tem are removed from the word and leaving the word ገልጠም gelTem.

1. Get WORD
2. Count number of radicals of WORD (n)
3. **If** n < 5 **then** stop and return WORD
4. Extract root of the WORD (C1C2C3...Cn)
5. **If** Ci=Ci+2 and Ci+1=Ci+3 **then**

   Remove Ci and Ci+1 with their vowel and copy the remaining to DWORD

   Return DWORD

Else return WORD

Figure 4.3 Algorithm for removing double reduplication

4.10.3 Prefix stripping

In removing the prefix from the word, which is the output of the previous procedure, the prefix and the rules are checked. As some of the prefixes should not be followed by a vowel the procedure checks this. For example, to remove the prefix ን /n-/ "for/to" the procedure first checks whether the prefix is followed by a vowel or not. The prefix /-n/ is removed from ንሰላም (nselam) "for peace" but not from ነገርና (negirna) because it is followed by a vowel.

1. Get WORD
2. Count number of radicals of WORD (n)
3. **If** n <=3 **then** stop and return WORD
4. **If** PLIST empty **then** stop and return WORD
5. **If** PREFIX does not match with WORD **then** Go to Step 4

   Count radicals of the prefix (np)

   If (n-np) >=3 **then**

   Remove PREFIX and copy sub-string to PWORD

   Copy PWORD to WORD

   Go to Step 3

   Else

   Go to Step 4

Figure 4.4 Algorithm for removing prefix

4.10.4 Suffix-stripping

After a prefix stripping operation is applied to a word, the next procedure is suffix stripping. This procedure checks the existence of a suffix, and if there is a rule attached to the suffix and it is satisfied the suffix is stripped from the word.

1. Get WORD
2. Count number of radicals of WORD (n)
3. **If** n <=3 **then** stop and return WORD
4. **If** SLIST empty **then** stop and return WORD
5. **If** SUFFIX does not match with WORD **then** Go to Step 4

   Count number of radicals of SUFFIX (ns)

   If (n-ns) >=3 **then**

   Remove PREFIX and copy sub-string to PWORD

   Copy PWORD to WORD

   Go to Step 3

   Else

   Go to Step 4

Figure 4.5 Algorithm for removing suffix

4.10.5 Removing single letter reduplication

This procedure is used to remove single letter reduplication. The procedure takes as an input the output of the previous procedure. It checks the occurrence of the reduplication and removes the first letter if the word contains single reduplication. If the letter to be removed is followed by a vowel, then the vowel is also stripped. For example the word

መመጽሃፍኩም memeShafkum, contains reduplicated letters መ me and መ me. Therefore the first letter me is removed from the word, leaving the word መጽሃፍኩም meShafkum.

1. Get WORD
2. Count number of radicals of WORD (n)
3. **If** n < 4 **then** stop and return WORD
4. Extract root of the WORD (C1C2C3...Cn)
5. **If** Ci=Ci+1 **then**

    Remove Ci with following vowel and copy the remaining sub-string to DWORD

    Return DWORD

Else

    Return WORD

Figure 4.6 Algorithm for removing single reduplication

The following example demonstrates how the algorithm works:

If we take the word ዝሰባበርናዮ(zsebabernayo) "what we have broken into pieces", first the prefix ዝ /-z/ is removed and leave the word ሰባበርናዮ (sebabernayo), then the suffix ናዮ /-nayo/ is removed and the word becomes ሰባበር sebaber. This word contains reduplicated consonant which is b (babe), and the single reduplication removing procedure removes ባ /-ba-/ and it becomes ሰበር (seber-) which is the stem of ሰበረ sebere "he broke".

4.11 EVALUATING AND IMPROVING THE STEMMER

For the purpose of evaluating the stemmer, a sample data of 1568 unique words were collected from the sample texts randomly. Quantitative approach was used to measure the performance of the stemmer. Basically it is on number of errors, that is, words that are not conflated correctly. Table 4.7 shows sample of the errors and their type. Four types of errors were found: order, understemming, overstemming and other.

Table 4.7 Examples of Stemming Error

Word	Resulting Stem	Expected Stem	Error Type
?abzgeberelun	bezg	geber	order
z?amenklu	nkl	?amen	order
zbetatn	betat	beten	order
korarmtu	korar	kormt	order
?afelalay	felal	felal	order
n?abalat	belat	?abal	order
msfaHn	faHn	sfaH	order
?ntrHrHu	ntH	rHrH	order
?agESe	?agES	gES	understemmed
bebiHade	beHad	Had	understemmed
slezKone	zKon	Kon	understemmed
qnyawi	qny	qn	understemmed
mekelaKeli	kelaK	kelaKel	Overstemmed
?anfetat	fetat	?anfet	Overstemmed
msTir	sTir	msTir	Overstemmed
~blela	~bl	~blel	Overstemmed
bebime~altu	~elt	me~alt	Overstemmed
deliKn	deliK	deley	other

From the manual assessment done on the stems, it showed that the stemmer performs at accuracy of 74%. The errors constitute: 6.9% overstemmed, 11.2% understemmed, 5.2% order and 2.6% other. Order indicates errors produce as result of order of the stripping procedures. For example, if we take the word ?afelaly "difference", it has prefix ?a, single letter reduplication lala (ll) and suffix y. If the order was single letter reduplication stripping, prefix stripping and suffix stripping, it would have resulted in the correct stem felay. The "other" errors represent errors, which were the results of combination. The stemmer was also evaluated in terms of degree of compression for stem and root. For calculating percentage of compression, the expression used by Popoviç (1992) for Slovene language is used. The compression C is defined by,

$$C = \frac{100 * (W - S)}{W}$$

Where W is the total word of the text and S is the stem or root.

The dictionary size (text size) and compression figures obtained for stem and root are given as follows:

Size of the data	1568
Number of stems	1166 (27% reduction)
Number of roots	799 (50% reduction)

Improving the Stemmer

To improve the stemmer, the errors were studied. The possible sources of the errors were: an affix not in the list, minimum stem length for some words and order of operation. The algorithm has five procedures: prefix-suffix pair stripping (ps), single letter reduplication

stripping (r1), prefix stripping (p), suffix stripping (s) and double letter reduplication stripping (r2). The order of operation was ps, then r2, then p, then s and finally r1. There was no enough time to test the possible ordering of the procedures to obtain the best result.

Modification was made on the stemmer by including some more affixes (e.g bebi-, ke- key-) and considering minimum stem length of two radicals for some of the words. For instance, the word ሃበ habe "he gave" with a consonant sequence ህ·ብ hb has two radicals. On the other hand this word has inflectional and derivational forms such as ሂበ hibe "I gave", ሂባ hiba "she gave", ሂበን hiben "they (f) gave", ሂቦም hibom "they (m) gave", ክህ·ብ khb "I will give", ምሃ·ብ mhab "to give", ክንህ·ብ knhb "we will give" to mention a few, that contain common radical ህ·ብ hb. These particular words have the same stem with a consonant sequence (root structure) ህ·ብ hb, if the prefix (k,m, kn) and suffix (en, om, na) are stripped. To conflate these words to the same stem hb, a minimum stem length of two radicals should be considered. Hence, when the remaining string has two radicals the algorithm checks if the sub-string has radical ህ·ብ hb. As discussed in chapter 3, two radical words are rare in Tigrigna. Therefore, in modifying the stemmer a list of two radical words was compiled by consulting dictionary. The list contains radical (root) representation of the words.

The new version stemmer included those modifications and was tested on the sample data. The result shows an increase of accuracy by 10%, raising the percentage of accuracy to 84%. Regarding the compression the following results were obtained:

Size of the data	1568
Number of stems	1059 (32.4% reduction)
Number of roots	717 (54.6% reduction)

While the understemmed errors were reduced by 9.4%, the overstemmed errors were increased by 0.5%.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Tigrigna is one of the Semitic languages. These languages have common grammatical system based on root-pattern structure. The main word formation process in Tigrigna is done through affixation. Tigrigna uses prefix, prefix-suffix pair and suffix. It also uses reduplication of part of a word, which is single and double letters. In Tigrigna the processes of adding one suffix to another can result in relatively long words, which often contain an amount of semantic information equivalent to a whole English phrase, clause or sentence. A Tigrigna word can give rise to a very large number of variants, with a consequent need for effective conflation procedures if high recall is to be achieved in searches of Tigrigna text databases. In this research, the possibility of developing stemming algorithm for the language was investigated.

The quantitative analysis done on the sample texts, which were colletced from three different sources showed that words are distributed throughout the texts in their morphological variants and singleton words constitute 35%-38% of the sample texts. Lower word ratio and deviation from the Zipf's law that was shown on the sample texts could be indications of the complexity of the language's morphology.

Stemmers developed for other languages could not be applied for this language because of the morphological complexity and difference in features of the language as discussed in Chapter 3. However, commonly used methods of stemmers such as using affix dictionary, stopword list and context sensitive rules are employed. Also some techniques are adopted from Porter (1980), Ahmad (1996) and Nega (1999) in developing the stemmer. Striping suffix is not

enough to conflate variant words of Tigrigna to one form. Hence the stemmer also includes procedures to remove prefix-suffix pair, prefix and reduplication of single and double letters.

To experiment the stemmer developed, test data of size 1568 words were slelected randomly from the sample texts. The experiment showed that the stemmer performs at accuracy of 84% and reduced the dictionary size by 32.4% and 54.6% for stems and roots respectively. This shows that using a stemming for Tigrigna brings a significant reduction in dictionary size as a result of conflating variant words to the same stem. The results obtained from the experiment are promising and using the stemmer in IR system of the language could improve the performance of the system.

The stemmer conflates only inflectional and derivational affixes. It does not conflate compounding and irregular forms. There are different ordering possibilities of applying the procedures. In this experiment the stripping order was prefix-suffix, double letter reduplication, prefix, suffix and single letter reduplication. Other possibilities could not be tested due to limitation in time.

5.2 RECOMMENDATION

This research demonstrated the possibility of developing a stemmer to conflate word variants of Tigrigna language, which has complex morphology and where a word can have thousand variants. As indicated above, however the study was based on a limited size of sample texts and not tested in IR environment within the time constraint. The precise mode of operation of the algorithm depends on the order of the stripping procedures, despite the fact that it is not clear in what order the procedures should be applied to an input word to obtain correct stem. The ordering could be based on arbitrary criteria or on a linguistic analysis. Due to limitation in time the merits of the different possible ordering could not be studied in this experiment.

In view of the importance of stemming in IR of Tigrigna and the encouraging results obtained in this research, the following recommendation are identified for further work in order to make the result useful in operational retrieval environment:

- developing a mechanism for producing electronic test collection of the language in Ethiopic format for the purpose of experimenting the stemmer and other related works;
- experimenting the stemmer on text collection of large size collected from different sources;
- studying the effect of ordering the stripping procedures on the performance of the stemmer and selecting the best possible ordering;
- experimenting the stemmer in IR environment to measure its performance in actual retrieval session;

Further more by doing additional researches on the language, this work can help to develop application tools such as spell checker, parser, thesaurus and dictionary

BIBLIOGRAPHY

1. Aba Mathewos Hagos (1951 EC[1]).

   ሰዋስው ትግርኛ ቀዳማይ መጽሓፍ ናይ ቃላት ሰዋስው ክሳብ ስሩዕ ረባሕታ. ናይ ማሕተም ትምህርቲ ፍራንቸስካና አሥመራ.

2. ብናይ ኢትዮጵያ ቋንቋታት ኣካዳማ (1989 EC[1]).

   መዝገበ ቃላት ትግርኛ ብትግርኛ.ኣዲስ አበባ.

3. Ahmad F., et al. (1996). Experiments with Stemming Algorithm for Malay Words. *Journal of the American society for Information Science,* 47(12), 909-918.

4. Al-Kharashi, I.A. and Evens, M. W. (1994). "Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System." *Journal of the American society for Information Science,* 45(8), 546-560.

5. Asmeret K/mariam. (1983). "The Morphomomemics of Noun and Verb in Tigrigna". BA Thesis. Addis Ababa University.

6. Bates, Marcia. (1998). Indexing and access for digital libraries and the Internet: Human, database, and domain factors". *Journal of the American society for Information Science,* 49(13), 1185-1205.

7. Bender, M.L. et al. (1976). Language in Ethiopia. Oxford University Press, London.

8. Central Statistic Office. (1991). *The 1984 Population and Housing Census of Ethiopia: Analytical Report at National Level*, Addis Ababa.

9. Central Statistic Office. (1999). *The 1994 Population and Housing Census of Ethiopia, Results at Country Level* Vol. II Analytical Report, Addis Ababa.

10. Chowdhury, G.G. (1999). *Introduction to modern information retrieval.* London: Library Association Publishing.

[1] EC Ethiopian Calander

11. Ekmekçioglu C.F., Lynch M. F. and Willett P. (1996). "Stemming and N-gram Matching for Term Conflation in Turkish Texts. " *at URL http://www.shef.ac.uk/~is/publications/infers/paper13.html.*

12. ELISE II (1999). Report on the feasibility of adding multilingual functionality to the search and browsing facilities of the ELISE system at URL http://nile.dmu.ac.uk/elise/el2_dels/d42_7d.htm

13. Girmay Berhane. (1991). "Issues on the Phonology and Morphology of Tigrinya", Doctoral Dissertaion. University du Quebec, A Montreal.

14. Harman, D. (1991). "How effective is suffixing?" *Journal of the American society for Information Science.* 42, 7-15.

15. Hetzron, R. (1969). "The Classification of Ethiopian Semitic Languages". University of California, California.

16. Hlava, M.K, et al. (1997). Cross Language Retrieval-English / Russian / French at URL http://www.accessinn.com/aaai.htm

17. Hmeidi I., Kanaan G. and Evens M. (1997). Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. Journal of the American Society for Information System, 48(10): 867 - 881.

18. Lennon, M., Peirce, D., Tarry, B., and Willett, P. (1981). "An evaluation of some conflation algorithms for information retrieval." *Journal of Information Science*, 3, 177-183.

19. Lovins, J.B.(1968). Development of Stemming Algorithm. Cambridge: *Electronic System Laboratory*, MIT.

20. McCarthy, J.J. (1982). "Formal Problems in Semitic Phonology and Morphology." University of Texas, Austin.

21. Ministry of Education.(1994). Education and Training Policy, Addis Ababa.

22. Nail, Martin. (1999) Information retrieval research at URL http://www.lic.gov.uk/research/information_retrieval/index.html

23. Nega Alemayehu. (1999). *Development of a Stemming Algorithm for Amharic Language Text Retrieval.* Ph.D Thesis. University of Sheffiled (unpublished).

24. Peters, Carol. (2000). "Multilingual Information Retrieval". Presented at Cross Language Evaluation Forum at URL *http://www.ercim.org/publication/Ercim_News/enw40/peters.html*

25. Popovič M. and Willett P. (1992). "The Effectiveness of Stemming for Natural-Language access to Slovene Textual Data." *Journal of the American society for Information Science,* 43(5), 391-395.

26. Porter, M.F.(1980). An Algorithm for Suffix Stripping. *Program, 14*, 130-137.

27. Porter, Martin. (2000). How to make stemming algorithms at URL http://open.muscat.com/developer/docs/stemming.html

28. Rijsbergen C.J.V.(1979). *Information Retrieval.* London: Butterworths.

29. Salton, Gerard, et al. (1981)."The measurement of term importance in automatic indexing." *Journal of the American society for Information Science,* 32(1-6), 175-186.

30. Salton, Gerard. (1983). "Introduction to modern information retrieval." New York: McGraw-Hill, Inc.

31. Savoy, J. (1993). "Stemming of French Words Based on grammatical Categories. "*Journal of the American society for Information Science,* 44(1), 1-9.

32. Scalise, S.(1984). "Generative Morphology". Foris Publications, Dodrecht-Holland/Cinnaminson-USA.

33. Selkirk, E.O.(1982). 'The Syntax of Words". The MIT press, Cambridge, Massachusetts.

34. Tesfai Tewolde. (1993). "Word Formation in Tigrinya." M.Sc. Thesis, Addis Ababa University (Unpublished).

35. Tesfaye Biru. (1987). "Incorporation of Relevance Data in The Term Discrimination Value." M.Sc. Dissertation, University of Sheffield (unpublished).

36. Trost, Harald (1993) Computational Morphology at URL http://www.ai.univie.ac.at/~harald/handbook.html

37. Tsegaye Taffere. (1987). "Derivations of Nouns in Tigrinya ".BA thesis. Addis Ababa University.

38. Wakshum Mekonnen. (2000). "Development of Stemming Algorithm for Afaan Oromoo Text." M.Sc. Thesis. Addis Ababa University (unpublished).

APPENDCES

APPENDIX I

The Zipf's constant for selected ranks of the sample texts

a) WOYN

Word	f	r	f*r
?ab	67	1	67
?wn	25	10	250
sraH	16	20	320
?ayte	11	30	330
kllna	10	40	400
?zom	9	50	450
TeQlala	7	60	420
?aKEba	6	70	420
zelo	5	80	400
bSay	4	90	360
bmKanu	4	100	400
menbernet	2	200	400
mSrarom	1	300	300
bbQ~at	1	400	400
feSaminetu	1	500	500
mergeStat	1	600	600
zeytwaSa?	1	700	700
tegbarat	1	800	800

hwaHat	1	900	900
kadren	1	1000	1000
melsi	1	1100	1100
HSretat	1	1179	1179

b) AGAZI

Word	f	r	f*r
?ab	74	1	74
?wn	29	10	290
?zi	15	20	300
neyrom	11	30	330
kulu	9	40	360
seb	8	50	400
qeSele	7	60	420
?ntay	6	70	420
~aserte	5	80	400
neSa	5	90	450
?ilu	5	100	500
mKnyat	2	200	400
?iKum	2	300	600
?mba	1	400	400
?abeban	1	500	500
z?atwulu	1	600	600
bQelilu	1	700	700

mdlay	1	800	800
qlTuf	1	900	900
?nazemeru	1	1000	1000
SeniHe	1	1100	1100
bHawi	1	1200	1200
bergiga	1	1300	1300
yre?ay	1	1400	1400
negeru	1	1500	1500
zHz	1	1502	1502

c) KNE

Word	f	r	f*r
?yu	54	1	54
meSHaf	16	10	160
nab	9	20	180
meskot	7	30	210
?ayneberen	4	40	160
derasi	4	50	200
QanQana	3	60	180
seb	3	70	210
?alo	3	80	240
qewami	3	90	270
?aqalilka	2	100	200
msfaHn	1	200	200

?nabela	1	300	300
?ntrd?on	1	400	400
wsedu	1	500	500
~aynet	1	600	600
kebabina	1	700	700
?afe	1	739	739

APPENDIX II

List of stopwords compiled from the sample texts

?ab	neyru	wdb	bmbal	se~at
?yu	Hade	gujle	gu~zo	ntom
nay	guba?E	l~li	?mber	Tgray
?ti	tegadelti	tgrNa	gna	seb
?zi	nayti	sraH	trgum	?abo
?wn	ma?serti	kunetat	lebewa	?abal
dma	?wan	kulu	gizE	kllna
bEt	hSuS	neyrom	derg	kllawi
kab	mKanu	qal	?alo	bmeseret
qnE	hweHat	QanQa	?ayneberen	bnay
kem	?abti	hzbi	wey	mengedi
nab	?surat	tgray	menber	bota
meQele	?yom	Halafnet	?ayte	?ta
Hayelom	medreK	gn	wdbn	?ntay

zbl	nti	komandotat	Hadega	bza~ba
mKri	ms	nezi	kll	zelewo
koynu	neti	klte	mengsti	?Ka
?tom	Hayli	feSamit	qalsi	Hbre
dmSi	?abzi	yKun	nabti	Tbeb
?abalat	meSHaf	?ilu	ksab	
do	wxTi	?zu	?ten	

APPENDIX III

List of prefix compiled from the sample texts

?	?n	k	m	ye
?a	?na	K	me	z
?ab	?ne	kE	ms	ze
?ake	?nkab	ke	n	zey
?an	?nt	kem	nay	zte
?ana	?t	Kem	sle	
?ane	?te	kemzi	sne	
?at	b	key	t	
?ate	bebi	keyte	te	
?ay	bzom	ki	y	

APPENDIX IV

List of suffix compiled from the sample texts

zu	te	nun	le	kana	eret	atni
zi	tatn	nu	la	ka	eren	atna
yu	tat	net	kyom	K	ere	atn
yn	ta	ne	kyo	iyawi	er	atkum
yda	t	nayom	kyen	it	eQu	atkn
yawi	su	nayo	kya	ir	enu	atki
yad	sti	nayen	kuwom	in	ena	atka
ya	ste	naya	kuwo	i~om	en	aten
y	siyawi	nani	kuwen	i?un	em	atat
xn	si	nana	kuwa	i?u	elun	at
wu	Selu	nan	kumwom	i	elu	asiyawi
wti	Se	naKum	kumwo	Hn	elti	anu
won	ru	nakum	kumwen	Hat	elom	an
wom	rti	naKn	kumwa	eyom	ele	amTa
wo	rn	nakn	kumni	eyen	eKat	am
wnti	ri	naKi	kumna	ey	eK	alu
wn	rHu	naki	kum	exn	edi	altu
wi	rHa	naKa	Kum	ewu	ebu	ale
way	ret	naka	kuKum	ewn	eberen	abn
uwom	ren	Na	kuKn	etun	ebe	a~ti
uwo	re	na	kuKi	etu	eato	a?u
uwen	rasiyawi	n?om	kuKa	etom	eata	a?om

uwa	Qu	n?o	ku	etni	e~altu	a?o
uni	otat	n?an	kni	etna	e	a?en
un	ot	n?a	knani	etn	dotat	a?a
um	one	n	knana	etkn	dom	a
ulu	on	mwo	kna?om	etki	do	~ti
uKum	omwom	mTa	kna	eten	bn	~om
uKn	omwo	mn	kn?o	etat	bat	~altu
uKi	omwen	mi	kn?an	Eta	azu	?ya
uka	omwa	m	kn?a	et	azi	?un
uKa	omni	lun	kn	esu	ayda	?om
u	omna	lu	ki	este	aya	?n
tun	omn	ltu	Kewn	esi	ay	
tu	omKum	lti	kayom	eselu	awyan	
tom	omKn	lom	kayo	eSe	awn	
to	omki	ln	kayen	es	awit	
tna	omKa	li	kaya	eru	awi	
tn	om	ley	Kat	erti	away	
ti	o	lesu	kani	erHu	atom	

APPENDIX V

Formulas for calculating similarity coefficient

Cosine

$$S = \frac{C}{\sqrt{A} \bullet \sqrt{B}}$$

Jaccard

$$S = \frac{C}{A + B - C}$$

APPENDIX VII

Comparison of unstemmed and stemmed texts

a) Unstemmed

ቤት ምኽሪ ብሄራዊ ክልላዊ መንግስቲ ትግራይ ኣባላት ናይቲ ቤት ምኽሪ ብመሰረት ዝሓ
ተትዎን ዘተኣኻኽብዎን ናይ ለበዋ ፊርማ ኣብ መወዳእታ ወርሒ መጋቢት ኣብ ህልው ኩ
ነታት ክልላዊ መንግስትን ዝጠመተ ህጹጽ ጉባኤ ኣካይዱኡሉ እዚ 30 መጋቢት 1993 ዝተጸ
ወዐ ጉባኤ ብቤት ጽሕፈት ናይቲ ክልል ዝተጸወዐ ስሩዕ ጉባኤ ኣይነበረን ኣብ ዓንቀጽ 5
0/40 ዝሰፈረ ሕገ መንግስቲ ብሄራዊ ክልላዊ መንግስቲ ትግራይ ብመሰረት ዝፈቀዶ ህጹጽ
ጉባኤ ንኽጽዋዕ ዝደሊ ደገፍ ንምትእኽኻብ ዝተወሰኑ ኣባላት ናይቲ ቤት ምኽሪ ዝገበር
ዎ ብምኻኑን ኣግባብነት ብዘለዎ ናብ ህዝቢ ክፍቲ ተገይሩ ብዕሊ ክካየድ ስለዝተወሰነ ዕዩ
ጉባኤ ክካየድ ዝኽኣለ
ቅድሚ ኩሉ ናይ ቤት ምኽሪ ክልላዊ መንግስትና ህጹጽ ጉባኤ ንክካየድ ዓርሰ ተበግሶ ወ
ሲዶም ኣብ ምትእኽኻብ ደገፍ ዝተዋፈሩ ኣባላት ናይቲ ቤት ምኽሪ ነዚ ለበዋ ንምግባር
ዘንቀሳቐሶም ምኽንያታት ናብቲ ጉባኤ ኣቕሪቦም እዮም
ህወሓት ኣመኔታ ዝሰኣነሎም ኣባላት ቤት ምኽሪ ብሄራዊ ክልላዊ መንግስቲ ትግራይ ዝኾኑ
ውልቀሰባት ብምህላዎም ናይ እዚ መግለጺ ድማ ደሞክራሳዊ ኣሰራርሓ ተጻሪሮም ጠጠው
ብምባሉም ኣብ ክልል እዚ ደሞክራሲያዊ ሕጋውን ዝኾኑ መድረኻት ብምርጋጽ ዘይሕጋ
ውን ንስርኣትና ዘናግዕን ምንቕስቓሳት እናካየዱ ብምህላዎም ብተወሳኺ ኣብ ሃገርና ሓፈሻዊ
ዘይምርግጋእ ዝፈጥር ኣዲ ዝበታትንን ንኢህወደግ ብምስንጣቕ ህልውናኡ ዘስእንን ተግባራ
ት ይፍጽሙ ስለዘተወ ኮሚቴ ስራሕ ኣፈጻሚት ክልልና ብዞምዝተጠቐሱ ምኽንያታት ኣብዞ
ም ውልቀሰባት ኣመኔታ ከሕድር ስለዘይከኣለን ብቑንድነቱ እውን ኣብ ክልልና ዘሎ መን
ግስታውን ህዝባውን ስራሕቲ ብሰንኪ እዚ ጠጠው ኢሉ ከምዘሎ ብተግባር ብምርግጹ እዩ
ብመሰረት እዚ እዮም ጠለብ ዘቕረቡ ኣባላት ቤት ምኽሪ ናብቲ ህጹጽ ቤት ምኽሪ ዘቕረ
ብዎ ለበዋ ኣይተ ገብሩ ኣስራት ኣቦ መንበር ክልልናን ኣባል ስራሕ ፈጻሚት ቤት ምኽ
ርን ነዞም ኣብ ላዕሊ ዝተዘርዘሩ ጸገማት ተሓታታይ ብምኻኖም ካብ ኣቦ መንበርነትን ካ
ብ ናይ ክልልና ስራሕ ፈጻሚትን እቲ ቤት ምኽሪ ዘትዩ ንኽወርድ ብኻልኣይ ደረጃ ኣ
ይተ ገብረመስቀል ሃይሉ ኣባል ስራሕ ፈጻሚት ቤት ምኽሪ ክልል ትግራይ ህዝብን ቤት
ምኽርን ዘንበረሉም ሓላፍነት መሰረት ገይሮም ዘይሰርሑ ምህላዎም ሓላፍነት መሰረት ተ

ጠቒሙን ንከወግዶም ነዚ ለበዋ እዚ ድማ እቲ ቤት ምኽሪ ተዛትዩ ንኽድግፎ ዘተሓሳስብ እዩ ነይሩ

ኣብ ሕገ መንግስቲ ክልልና ዓንቀጽ 50 ንኡስ ዓንቀስ 4 ሰፊሩ ከምዝርከብ ስሩዕ ኣኼባ ቤት ምኽሪ ኣብዘይህልውሉ እዋን ልዕሊ ፍርቂ ኣባል ክጽዋዕ እንትትድለ ርዕሰ ምምሕዳር ናይቲ ክልል ሀጹጽ ኣኼባ ክጽውዕ ግድነት ኣለዎ ይብል ይኹን እምበር ርዕሰ ምምሕዳር ናይቲ ክልል ነዚ ሀጹጽ ጉባኤ ዝጠልብ መድረኽ ክጥቀመሉ ብዘይምኽኣሉ ኣባላት ናይቲ ቤት ምኽሪ ዓርሰ ተበግሶ ወሲዶም ፐቲሽን ኣተኣኻኺቦም ሀጹጽ ጉባኤ ክጽዋዕ ሓቶቶም ብመሰረት ሕጊ ሕገ መንግስቲ ክልል ትግራይ ካብ እቶም ጠቕላላ ኣባላት ቤት ምኽሪ ልዕሊ ፍርቂ እንተሓቲቶም እቲ ሀጹጽ ጉባኤ ክጽዋዕ ስለዝፈቕድ ኣባላት ቤት ምኽሪ ትግራይ ካብ ዝኾኑ 152 ተወከልቲ ህዝቢ ድማ እቶም 128 ኣባላት ናይቲ ቤት ምኽሪ ዝፈረሙሉ ናይ ፐቲሽን ፊርማ ሀጹጽ ጉባኤ ክካየድ ከምዘለዎ ዝሓትት ብምንባሩ ንቤት ጽሕፈት ምምሕድዳር ክልል ትግራይ ቀሪቡ እዩ እዚ ኮይኑ ናይቲ ክልል ኣቦ መንበር ነዚ ጉባኤ ክጽዋዕ ትጽቢት ኣብዝገበረሉን ነቲ ሀጹጽ ጉባኤ ክመርሕ ብኣባል ይኹን ብጽሑፍ ተደጋጊሙ ጻዊኢት እንትበጽሖ ሰናያ ፍቓድ ከየርኣየ ተሪፉ እዚ ምኻኑ ምስተረጋገጸ ድማ እዩ ኣብ ዓንቀጽ 59 ህገ መንግስቲ ክልልና ርዕሰ ምምሕድዳር ኣብ ዘይህልወሉ ምክትል ርዕሰ ምምሕድዳር ተኪኡ ከምዝሰርሕ ብመሰረት ዝገልጾ እቲ ሀጹጽ ጉባኤ ሕፋውነት ብዝተኽተለ

ሕጋውነት ናይዚ ሀጹጽ ጉባኤ ካብ ኣጸዋውዕኡ ክሳብ ናይ ኣካያይዳ ስርዓት ሕጋውን ንህን ጻ ህገ መንግስቲ ክልላዊ መንግስትና መሰረት ዝገበረን ምኻኑ ብናይ ኣባላት ድምጺ ምስተረጋገጸ እቲ ቤት ምኽሪ ስርሑ ብምጅማር ኣባላት ናይ እዚ ቤት ምኽሪ ነቲ ኣብ ላዕሊ ዝሰፈረን ናብቲ ሀጹጽ ጉባኤ ቀሪቡ ዘሎን ለበዋ ርእይቶኦም ክህብሉ ብምዕዳም ነቲ መድረኽ ከፊቱ ኣብ ህልዊ ኩነታትውድብናን ክልልላዊ መንግስትናን ተደሪኾም ኣብቲ መድረኽ ካብ ዝቐረቡ ስፍሓት ዘለዎም ርእይቶታት ድማ ነቶም ዝተወሰኑ ብዝተጸሞቐ ኣገላልጻ ምርኣይ ይከኣል እዞም ንህልዊ ኩነታት ውድብናን መንግስትናን ክንፈልጥን ኣብ ሽግራቱ መኺርና ዕልባት ንክንእልሸሉ መድረኽ ክኸፍቱ ለበዋ ዘቕረብናሉም ሰባት ንኣረኣእያን ን እምነታትን ህዋሓት ሒዞም ምሳና እናተቓለሱን እናቃለሱን መጺኦም እዮም መምርሕታትን ፖሊሲታትን ውድብና እናተግበሩ ብምንባሮም ኣብኦም ብዘይምንም ጥርጥር ዝህቡና መደባት ክንፍጽም ጸኒሕና እዚ ኮይኑ ቅድሚኦም ከም ውድብን ከም ህዝብን ብዙሕ ተሞክሮታት ነይሩና ብጣዕሚ እንኣምኖምን መራሕትና ኢልና እንፈትዎምን ብዙሓት ተቓለስቲ ኣብ ጎንና ነይሮም ይኹን እምበር ሽኡኡ እውን እንተኾነ ኣብ ዕላማን መስመርን እንተዘይኮኑ ኣን ሰባት ዝተደረኽ ውድባዊ እምነት

b) Stemmed (stopwords excluded)

ሄር ሓተት ኽኸብ ፌርም ወዳእ ወርሕ ገቢት ህልው ነግስ ጠመት ዱኣል ገቢት ጸወዕ ብ ቤት ጽሕፍ ጸውዕ ስሩዕ ዓንቀጽ ሰፈር ሕግ ሄር ፍቀድ ጽዋዕ ዘድል ደገፍ ኽኻብ ወሰን ገበር ቅስቓስ ፍርቅ ደገፍ ርኻብ ነግስ ህንጽ ደግፍ ኻኑን ገባብ ዘለ ክፍት ገይር ብዕ ል ካየድ ወስ ካየድ ኽኣል ቅድም ነግስ ካየድ ዓርስ በግስ ወሰድ ኽኻብ ደገፍ ዋፈር ግ ባር ነቀሳቕ ያት ቐረብ መኔት ሰኣን ሄር ዝኾኑ ውልቅ ህላው ገለስ ዴሞክር ሰረሕ ውድ ብ ነግስ ጥሃስ ከል ውህድ በዝሕ ውሳን ግዝእ ሰረሕ ጻረር ጠጠው ባል ዴሞክር ሕግ ዝኾን ደር ርጋጽ ሕግ ሰርእ ናቅዕ ቓስ ካየድ ህላው ወሳኽ ሃገር ሓፈሽ ርገእ ፈጥር ኣ ድ በት ኢህወደግ ንጣቕ ህልው ስእን ግባር ፍጽም ዘል ኮሚት ፈጻም ጠቐስ ያት በዝ ውልቅ መኔት ከሕድር ከኣል ቐንድ ዘል ነግስ ህዝብ ስራሕ ሰንክ ጠጠው ከምዝ ግባር ርገጽ ቕረብ ቕረብ ገብር ሰር ልልን ኽርን ነዝ ተዘር ጸገም ሓት ኻን እብ ነበር ፈጻም ዝትይ ወርድ ኻልእ ደረጅ ህዝብ ኽርን በረል ሰረት ገይር ሰርሕ ህላው ሰረት ጠቂም ከወግ ልበው ዛይት ድግፍ ሓሰብ ሕግ ዓንቀጽ ዕንቅ ሰፈር ረህብ ስሩዕ ኣኼብ ህልው ፍርቅ ጽዋዕ ትድል ርዕስ ሕደር ኣኼብ ጽዋዕ ግድን ኣል ይብል ርዕስ ሕደር ጠልብ ጥቀም ኻል ዓርስ በግስ ወሰድ ፐቲሽ ኽኸብ ጽዋዕ ሓተት ሕግ ሕግ ጠቕል ፍርቕ ሓተ ት ጽዋዕ ፈቅድ ዝኾን ወከል ፈረም ፐቲሽ ፌርም ካየድ ዘል ሓትት ንባር ንብ ጽሕፍ ሕደር ቀረብ ጽዋዕ ጽቢት ገበር መርሕ ኣባል ጽሑፍ ደጋግ ጻዊዕ በጽሕ ሰን ፍቓድ ርኣ ይ ተረፍ ረጋግ ዓንቀጽ ህግ ርዕስ ሕደር ህልው ክትል ርዕስ ሕደር ትከእ ሰርሕ ገልጽ ሕጋው ኽተል ሕጋው ናይዝ ጸወዕ ከየድ ስርዓ ሕግ ህንጽ ህግ ነገስ ሰረት ገበር ረጋግ ስርሕ ጅማር ሰፈር ቀረብ ዘሎን ርእይ ህብል ዕዳም ከፈት ህልው ውድብ ነግስ ደረኽ ቐረብ ጽፍሕ ለው ርእይ ነት ወሰን ጸሞቕ ገለጽ ርኣይ ከኣል እዝ ህልው ውድብ ነግስ ፈልጥ ሸግር ኽርን ዕልብ ለሸል ኽፍት ቕረብ ሰብ ረኣእ ምነት ሒዝ ምሳን ቓለስ ቃ ከስ ጸእ ረሕት ፖሊስ ውድብ ግበር ባር በእ ምሉእ ምነት ሓዲር ህዝብ ዓበይ ረሕት በል ምንም ጥርጥር ሃቡን ደብ ፍጽም ጸኒሕ ቅድም ህዝብ ሞክር ነይር ጣዕም ኣኦምን ረሕት ኢልን ፈትው ዙሕ ቓለስ ጎን ሸኡእ ተኾን ዕላም ሰመር ኮይን ኣን ሰብ ደርኽ ውድብ ምነት

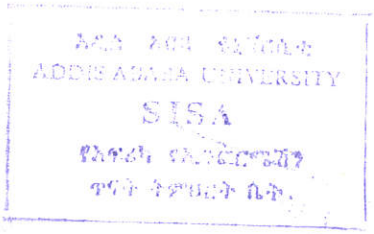
DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

GIRMA BERHE

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

DR. GEBREMEDHIN SIMON



ATO TESFAYE BIRU