ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

Development of a Stemming Algorithm for *Afaan Oromoo* Text

A THESIS submitted in partial fulfillment of the requirement for the Degree of Master of Science in Information Science.

By **Wakshum Mekonnen Tucho**

June, 2000

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

DEVELOPMENT OF STEMMING ALGORITHM FOR AFAAN OROMO LANUGAGE TEXT

BY

WAKSHUM MKEONNEN

Name and Signature of Members of the Examining Board

Ato Getachew Birru, Chairman, Examining Board _____

Dr. Nega Alemayehu, Advisor _____

Dr. John Cowell, External Examiner _____

Dedicated to

My son, Natanaan,

Who was born during the first year of my study at the School of Graduate Studies.

ABSTRACT

This paper reports the design and development of a stemming algorithm for *Afaan Oromoo* language. Reviews of *Afaan Oromoo* morphology, stemming algorithms, and other relevant materials were made. In *Afaan Oromoo,* inflectional and derivational affixations are the major word formation processes. The initial design of the stemming algorithm was based on free-context conflation procedures following the longest-match suffix removal approach. An accuracy rate of 71% was obtained from this initial attempt. The improved algorithm incorporated suffix, context-sensitive, and recording rules in the procedures. Before stemming, functional and frequently occurring words, which were compiled as stoplist, are excluded from the input term(s) to increase the efficiency of the stemmer. Procedures for prefix removal and for conflation of words formed by reduplication of first syllable are also components of the modified algorithm. Using the modified stemmer an accuracy rate of 92% was gained from the test based on a sample of 1061 words. The percentage of errors recorded as understemming and overstemming were reduced to 4.58% and 2.5% respectively from 10.5% and 17.5% for the first version. A substantial decrease in size of sample text is achieved from this stemmer. The morphological complexity of the language is the main sources of errors for the resulting inaccuracies of the stemming algorithm. For further improvement of the stemmer therefore, detailed study of *afaan Oromoo* morphology is helpful. The result of this study in general shows the possibility of employing a stemming algorithm for conflating *Afaan Oromoo* words.

ABBREVIATIONS AND SYMBOLS USED IN THE STUDY

2sgm	=	2nd singular person masculine
3sgm	=	third singular masculine
2sg	=	2nd singular person
1pl	=	first person plural
msg	=	male singular
mpl	=	male plural
fsg	=	Female singular
fpl	=	female plural
perf	=	perfective marker
imperf	=	imperfective marker
→	=	becomes

TABLE OF CONTENTS

DECLARATION i
Dedicated to ii
ACKNOWLEDGEMENT iii
ABSTRACT iv
ABBREVIATIONS AND SYMBOLS USED IN THE STUDY v

CHAPTER 1 1

INTRODUCTION **1**
1.1 BACKGROUND OF THE STUDY 1
1.2 STATEMENT OF THE PROBLEM 6
1.3 JUSTIFICATION OF THE STUDY 8
1.4 OBJECTIVES 10
1.4.1 General Objective 10
1.4.2 Specific objectives 10
1.5 METHOD 10
1.5.1 Review of Related Literature 10
1.5.2 Programming Techniques 11
1.5.3 Testing Techniques 12
1.5.3.1 Developing and Testing the Stemming Algorithm for *Afaan Oromoo* 13
1.6 SCOPE AND LIMITATION OF THE STUDY 13
1.7 ORGANIZATION OF THE THESIS 14

CHAPTER 2 15

STEMMING ALGORITHM **15**
2.1 INTRODUCTION 15
2.2 AUTOMATIC WORD CONFLATION 16
2.2.1 Stemming Algorithm 18
2.2.1.1 Manner of Operation 19
2.2.1.2 Context Sensitive and Recording Rules 21
2.2.1.3 The Suffix Dictionary 23
2.2.2 Successor Variety 23
2.2.3 n – gram 26
2.2.4 Stemming Algorithm: A Review 27
2.3 STOP WORD LIST 32
2.4 CHAPTER SUMMARY 33

CHAPTER 3 34

THE AFAAN OROMOO LANGUAGE **34**
3.1 INTRODUCTION 34
3.2 THE *OROMO* ALPHABET: *QUBEE AFAAN OROMOO* 35
3.2.1 *Afaan Oromoo* Consonants 36

3.2.2 *Afaan Oromoo* Vowels 37
3.3 MORPHOLOGY 37
3.3.1 Kinds of Morphology 38
3.3.1.1 Inflectional Morphology 38
3.3.1.2 Derivational Morphology 38
3.4 THE *AFAAN OROMOO* MORPHOLOGY 39
3.5 HOW MORPHEMES OCCUR IN *AFAAN OROMOO* 41
3.5.1 Affixation 41
3.5.1.1 Inflectional affixes 42
3.5.1.1.1 Vowel Affixes 42
3.5.1.1.2 Consonantal Afffixes 44
3.5.1.2 Derivational Suffixes 50
3.5.1.2.1 Derived Verbs 50
3.5.1.2.2 Derived Nouns 56
3.5.1.2.3 Derived Adjectives 57
3.5.1.3 Case and relational concepts 57
3.5.2 Compounding 59
3.6 CONCLUSION 59

CHAPTER 4 61

EXPERIMENTING THE STEMMING ALGORITHM **61**
4.0 INTRODUCTION 61
4.1 THE TEST DATA 61
4.2 THE WORD DISTRIBUTION OF AFAAN OROMOO 63
4.3 STOPLIST 68
4.4 *AFAAN OROMOO* AFFIXES 70
4.4.1. Prefix 70
4.4.2 Suffixes 70
4.4.3 Prefix-Suffix Pairs 71
4.5 THE *AFAAN OROMOO* STEMMER 72
4.6 EVALUATION OF THE ALGORITHM 76
4.7 THE IMPROVED STEMMER 78

CHAPTER 5 85

CONCLUSIONS AND RECOMMENDATIONS **85**
5.1 CONCLUSIONS 85
5.2 RECOMMENDATIONS 86

BIBILIOGRAPHY 88

APPENDCES 93

LIST OF TABLES

TABLE 2.1: LIST OF POSSIBLE SUCCESSOR VARIETIES 24
TABLE 2.2: EXAMPLES OF MEASURES OF TERMS 30
TABLE 2.3: STEP1A RULES OF PORTER'S STEMMER 31
TABLE 3.1: EXAMPLES OF AFFIXATION PROCESSES IN *AFAAN OROMOO* 42
TABLE 3.2: INFLECTED ADJECTIVES 49
TABLE 3.3: EXAMPLES OF BENEFACTIVE/REFLEXIVE 51
TABLE 3.4: EXAMPLES OF BENEFACTIVE/REFLEXIVE STEM VERB DERIVATION 52
TABLE 3.5: STATIVE VERBS 55
TABLE 3.6: EXAMPLES OF PASSIVES 56
TABLE 4.1: WORD RATIOS CALCULATED FOR SAMPLE TEXTS OF AFAAN OROMOO 63
TABLE 4.2 : THE WORD DISTRIBUTION IN PERCENTAGE FOR FREQUENCY OF TWO OR ONE ... 64
TABLE 4.2: COMPARISON OF WORD RATIOS OF AFAAN OROMOO WITH ENGLISH AND ARABIC 67
TABLE 4.3: LIST OF THE MOST FREQUENTLY OCCURRING WORDS FROM SIX SAMPLE TEXTS 69
TABLE 4.4: LIST OF AFAAN OROMOO SUFFIXES 71
TABLE 4.5: EXAMPLES OF CONFLATED TERMS BY THE FIRST VERSION OF AFAAN OROMOO STEMMER 75

LIST OF FIGURES

FIG. 3.1: AFAAN OROMOO CONSTANTS 36
FIG. 3.2: AFAAN OROMOO VOWELS 37
FIG. 4.1: THE FIRST TRIAL OF AFAAN OROMOO STEMMING ALGORITHM 73
FIG 4.2: PREFIX REMOVAL ALGORITHM 79
FIG.4.3: ALGORTHM FOR SUFFIX REMOVAL 80
FIG. 4.4: ALGORITHM FOR THE MAIN PROGRAM OF AFAAN OROMOO STEMMER 82

LIST OF APPENDICES

APPENDIX I 93
A) PRONUNCIATION OF AFAAN OROMOO VOWELS 93
(B) PRONUNCIATION OF AFAAN OROMOO CONSONANTS 94
APPENDIX II 95
EXAMPLE OF A STEM AND ITS DIFFERENT VARIANTS 95
APPENDIX III 96
STOP WORDS COMPILED FROM AFAAN OROMOO SAMPLE TEXTS 96
APPENDIX IV 98
LIST OF AFAAN OROMOO WORD ENDINGS (SUFFIXES) 98
APPENDIX V 101
EXAMPLES OF WORDS AND THE RESULTING STEM BY THE STEMMING ALGORITHM. 101
APPENDIX VI 105
COMPARISON OF UNSTEMMED AND STEMMED TEXTS 105
APPENDIX VII 106
FREQUENCY OF WORDS AND ZIPF'S CONSTANT FOR SELECTED RANKS 106

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Timely and relevant information is one of the most important requirements for all kinds of activities in an information society. With no doubt, the information 'haves' are in a better position in all aspect than the 'have nots', since information now a days has become the lifeblood of every development activity. Therefore, we need to look for effective storage and retrieval system in order to store, process, and retrieve the information we need with minimal expense and time.

The effectiveness of an information retrieval (IR) system entirely depends on the indexing and searching techniques employed. In a very crude term, the search result from a given query can be an indication of how effective a given information retrieval is. It is common that in a textual database, words occur in various forms (either inflectional or derivational). For example, COMPUTER, COMPUTERS, COMPUTATION, COMPUTERIZATION, and so on, are morphological variants of the word COMPUTE. It is awkward to give the different forms of COMPUTE in order to retrieve related documents. In such a case, the system either

fails to retrieve relevant documents or the searcher has to use all variants.

In order to improve the retrieval effectiveness of an information retrieval system, a conflation procedure is used to reduce variants to the same form. Among the different methods used, stemming is popular in information retrieval systems. Lovins (1968) defines a stemming algorithm as "a computational procedure which reduces all words with the same root (or, if prefix are left untouched, the same stem) to a common form usually by stripping each word of its derivational and inflectional suffixes." Stemming is simply a process that involves striping off suffixes from terms so as to reduce terms with different endings or forms to a common stem. Stemming algorithm is one of the important tools of information retrieval.

The purpose of Information Storage and Retrieval is to extract relevant documents from a large collection in response to users' queries (Strazalkowski, 1995). The task of an information retrieval system thus involves the retrieval of relevant documents and at the same time suppressing the non-relevant ones from retrieval.

In an informational retrieval system (IRS), users generate a request, referred as a query, for their information need. The query may be in a form of Boolean expression or in a form of a natural language. The natural language expression can be a sentence or a set of sentences, a phrase, or a single word. In response to such queries, a typical IRS provides the user with a list of documents.

A method of representing (i.e., indexing) the content of a document and queries is an important activity in IR. Searching and retrieving a document in a database (or Information System, IS) could be effectively conducted if only the document has been effectively indexed (Tiziana, 1997). Indexing is therefore one of the important activities of IR system.

Salton (1989) defines indexing as a process of constructing document surrogates by assigning identifiers to text items. There are two techniques of indexing documents: manual indexing and automatic indexing. In manual indexing technique, it is the subject expert or trained person who assigns content description to a given document. Automatic indexing on the other hand is a process in which documents and queries are analyzed and described automatically using an automatically created index language for retrieval purpose.

As many writers argue, manual indexing has a number of drawbacks. Flynn (1987) for instance says, manual indexing is a highly labor intensive activity; inevitably subjective and inconsistent; requires highly qualified indexers in the respective field and it is expensive. In view of such problems with manual indexing, automatic indexing is usually preferred.

The process of automatic indexing has three parts (Rijsbergen, 1975: 15): (1) removal of high frequency words, (2) suffix stripping, and (3) detecting equivalent terms. The first part involves the removal of frequently occurring words (stop words), such as articles, which have least significance in representing a document. According to Rijsbergen, the size of the document can be reduced by 30 to 50 percent by excluding stop words. After removing the

stop words the second process that further compresses the document is the suffix stripping (commonly done by stemming algorithm) process. Suffix stripping is a conflation procedure that reduces different variants to a single common term by removing suffixes. The last part of this process deals with selection of index terms. Terms in documents are given weights in the selection of terms for index terms.

There are several approaches of calculating the significance of terms in a document. One of these approaches is the one discussed by Frants et al. (1997) based on Luhn's proposal. According to this approach, the significance of the term *t* of document *i* denoted by IMP_{ti}, is calculated as,

(1.1) $$IMP_{ti} = f_{ti}/F_t$$

Where F_t is the total occurrence frequency of the word in *n* documents. It is given as

$F_t = \Sigma f_{ti}$, f_{ti} is individual occurrences in a single document.

According to (1.1), the value is low for very frequent words and is high for those rarely occurring words. Based on the values obtained the term is included in the document profile if it exceeds a certain preset threshold. Obviously, terms with high weights are the candidates. Frants et al (1997) stress that, stemming algorithm is an important activity that should precede the calculation of the frequencies of occurrence irrespective of any method used in this regard.

A number of stemming algorithms has been developed to enhance information storage and retrieval performance. Porter (1980) and Lovins (1968) are two of the classical English language stemmers. The existing stemmers for other languages include French (Savoy, 1993), Latin (Schinke et al, 1996) and Malay (Ahemed et al., 1996).

The evaluations for English show controversial results. Harman (1991), for instance, reported a weakly positive result on retrieval performance by evaluating the Lovins stemmer, S – stemmer and the Porter Stemmer. However, the results from some of the recent experiments (e.g., Hull, 1996) shows that stemming is generally beneficial for English.

On the other hand, stemming research on other morphologically complex languages indicate significant performance difference between stemmed and non-stemmed operations. Among the stemmers for which a substantial increase in performance reported include Slovene (Popovic & Willett, 1992) and Arabic (Al-kharashi & Evens, 1994).

Al-Kharashi & Evens (1994) show that the use of stemming increases the number of retrieved documents or the success of the matching of documents to a query. It is because, without stemming, the result of our search includes only a given term or the original term, so that there will be a chance for several relevant documents to be missed. Suppose the query term for searching in a given database is "moving". Without stemming, this would match only the records containing "moving" and misses other relevant terms that could represent the document. Moreover, following stemming, all records that are reducible to the

stem "move", such as "moves", "moved", "movement", "movable" and "movers" will be matched and retrieved.

The area of application of stemming other than IR includes (Dawson, 174): Indexing; Etymology –that functions with the help of conflated word index; Word frequency count, especially for complex morphological languages and Word parsing. The stemming operation however depends on language, such that we require separate stemmers for different languages.

Currently, huge information resources is becoming available electronically on the Internet, in CD-ROMs and on different other networks in various languages. The ease of retrieval of such information and knowledge is very important in today's world, where development in every aspect is entirely based on information. In spite of these, the development of computational tools for *Afaan Oromoo* text retrieval is in its infancy.

Afaan Oromoo is one of the most widely spoken languages in Africa. To the best of my knowledge however, yet there is no any stemming algorithms developed for the language to provide information-searching tools or to enhance the retrieval performance in general. In this work therefore, an attempt is made to develop a stemming algorithm for A*faan Oromoo*.

1.2 Statement of the Problem

Large numbers of people who live in Ethiopia speak *Afaan Oromoo*. There is also

significant number of speakers of the language out side Ethiopia. The non-natives who live in the *Oromo* areas also use the language. The role of the language for such large population in social, political and economic activities is unquestionable. It is believed that, the level of development of the language has great impacts on the effectiveness of information exchange and communication.

Afaan Oromoo is officially used in Oromyia Regional State since the end of 1991. It is used in education, public services, government offices, and other events including religious teachings. As a result, a significant number of people are able to read and write in *Afaan Oromoo* since then. More importantly, the language serves as a medium of instruction at the primary and secondary schools in this region. In these schools, textbooks and other reference material are available in *Afaan Oromoo*. Training is also given for teacher candidates in the Teachers Training Center in *Afaan Oromoo* in this Region. Furthermore, the language is offered as a minor course at the Institute of Language Studies (ILS), Addis Ababa University.

There are varieties of publication in hard copies in *Afaan Oromoo*, mainly on language, history, literature, and political issues. This includes books, newspapers, and other press products. There are also some electronic publications on the Internet in *Afaan Oromoo*. The major development however is in its linguistic aspects. The computational development of the language, especially in storage, processing and retrieval of information is in its infancy. There are some initiatives in this regard. Oromo Soft and Barissa are examples of word processors developed for *Afaan Oromoo* by some people who live abroad.

These days, computers are common in government offices, higher institutions, and in business enterprises in *Oromyia* as well as in other parts of Ethiopia used for word processing and for desktop publishing activities. Many of them have access to the Internet as well. In general, all sorts of information are becoming available in *Afaan Oromoo* electronically.

Much of the information we use appear in natural language compiled in hard copies – books, journals, reports, etc., and/or in soft copies – CDs, diskettes, cassettes, etc. Very recently, highly distributed, vastly diversified and heterogeneous information in electronic format is becoming available especially through the Internet. Clearly, the retrieval of relevant information to the user request from such huge source is not an easy task without the use of automatic information retrieval system.

1.3 Justification of the Study

Stemming algorithm for *Afaan Oromoo* is necessarily required to develop IR system to handle the retrieval activity efficiently. Stemming facilitates the search activity to locate relevant terms to users request that would otherwise be missed. Stemming is therefore, a prerequisite for search and other similar activities. In view of this, the result from this research is becoming a foundation for developing not only search tools but also for developing other computational tools, such as, indexing, word frequency counting, spell checkers, grammar, thesauri, etc.

In particular, the following points are taken into consideration.

- Currently there are a number of resources available in electronic form in *Afaan Oromoo*. The content of such information is of diversified in nature.
- *Afaan Oromoo* is the medium of instruction for Primary and Junior Secondary Schools in *Oromyia* Regional State. Textbooks as well as reference materials are available in *Afaan Oromoo* for use in the schools. In the future, there is a high possibility for the schools to be computerized and networked, thus all kinds of knowledge would be converted to electronic form and placed on the network or in digital libraries.
- There is a trend for the development of community information systems that gives emphasis to local contents in local languages. Early this year, a community-based Internet service funded by the British Council has been started at Walliso, a rural town located at about 130 kms. south west of the capital, Addis Ababa.
- The official language of *Oromiya* Region is *Afaan Oromoo*.
- Stemmers for other languages in the Cushitic family can be developed with less effort
- This research can be taken as a pioneer work to be used by other professionals in the field to serve as a base for further study for the development of IR tools for *Afaan Oromoo*.

Furthermore, it is believed that, the development of a stemmer for *Afaan Oromoo*, as a tool for IR would have contribution in developing the language computationally.

1.4 Objectives

1.4.1 General Objective

The main objective of this research is to investigate the possibility of developing a stemming algorithm for *Afaan Oromoo.*

1.4.2 Specific objectives

The specific objectives of the research are to:

- review related literature on morphology of *Afaan Oromoo* and stemming algorithms;
- compile a list of affixes used in *Afaan Oromoo*;
- compile a stop word list;
- develop stemming algorithm to experiment with *Afaan Oromoo* text;
- write computer programs for stemming inflectional and derivational affixes; and
- evaluate the stemmer how it performs on selected sample words.

1.5 Method

1.5.1 Review of Related Literature

Various literatures on the subject were consulted. Articles in journals, books, thesis and other related sources in hard copies reviewed. Other source of the literature reviewed includes soft copies in CD-ROMs and materials on the Internet web pages. Faculty and other appropriate individuals were consulted on the morphology of *Afaan Oromoo.*

1.4 Objectives

1.4.1 General Objective

The main objective of this research is to investigate the possibility of developing a stemming algorithm for *Afaan Oromoo.*

1.4.2 Specific objectives

The specific objectives of the research are to:

- review related literature on morphology of *Afaan Oromoo* and stemming algorithms;
- compile a list of affixes used in *Afaan Oromoo*;
- compile a stop word list;
- develop stemming algorithm to experiment with *Afaan Oromoo* text;
- write computer programs for stemming inflectional and derivational affixes; and
- evaluate the stemmer how it performs on selected sample words.

1.5 Method

1.5.1 Review of Related Literature

Various literatures on the subject were consulted. Articles in journals, books, thesis and other related sources in hard copies reviewed. Other source of the literature reviewed includes soft copies in CD-ROMs and materials on the Internet web pages. Faculty and other appropriate individuals were consulted on the morphology of *Afaan Oromoo.*

1.5.2 Programming Techniques

To develop the stemmer, the existing stemming algorithms were studied and tested for *Afaan Oromoo* texts. Java programming language was selected to write the source code for the stemmer for many of its advantages. The selections of Java for this study among other things are :

- Java automatically de-allocates memory, such that one need not spend too much of his/her time in debugging the inevitable errors that usually occur as in C\C++. The programmer therefore gets ample time to concentrate on the problems at hand rather than on the machine level problems.
- Java uses some of the syntaxes of C/C++ for one thing, thereby it is easier to switch to Java with less effort.
- The compiler is freely avialable.
- Java supports object-oriented programming approach.

Moreover, according to Ladd and O'Donnell (1996), Java provides the following advantages: Java,

- is architecturally neutral, i.e., platform independent;
- is a network aware and truly distributed ;
- supports all of the benefits of locally executed programs, apart from its intensive application on the Internet; and
- faster and efficient.

Furthermore, Java is platform independent such that it would be easier to extend the study to web applications.

1.5.3 Testing Techniques

The Sources of data

Six sample texts from different disciplines (e.g., history, science, literature and culture) in *Afaan Oromoo* were collected based on the availability of such documents in electronic formats. The samples were acquired from Education bureaus of *Oromyia* Regional State and from *Barisaa* Section of the Ethiopian Press Agency.

Suffix list compilation

The following sequence of operations were performed on the sample texts to compile suffix list:

- The words in the texts were first written in reverse order.
- The reversed list of the words was sorted and words with similar suffixes then come together
- The list was scanned for most frequently occuring endings
- The sub-string with high frequency of occurances were selected as suffixes/endings
- List of suffixes prepared
- The procedure was repeated for all other samples
- The separate list of suffixes merged and then compiled to a stoplist

1.5.3.1 Developing and Testing the Stemming Algorithm for *Afaan Oromoo*

The stemming algorithm approach used to develop the Afaan Oromoo stemmer is longest-match suffix stripping. During program development a prototype approach was adapted.

The algorithm was tested using the error counting technique. Errors due to inaccurate stemming are regarded as overstemming and understemming. Overstemming is when too longe of suffix is removed while understemming is when too short a suffix is removed. The extent of accuracy of the stemmer is determined by qualitative analysis of the output from the stemmer. The stemmer is also tested for compression performance. The result of the evaluation is quantitatively analyzed using basic statistics.

1.6 Scope and Limitation of the Study

The stemming algorithm developed for *Afaan Oromoo* text is limited to conflate only those inflectional and derivational variants formed in regular pattern. The stemmer does not conflate compounds and some variants that have irregular patterns of word formation due to time constraints. Therefore, detailed morphological analysis of *Afaan Oromoo* and development of more complex stemming algorithm will be required to improve the performance of the stemmer.

1.7 Organization of the Thesis

The thesis is divided into five chapters. The first chapter comprises background of the study that introduces stemming and its applications in information retrieval system and the rationale for developing the stemming algorithm. The chapter contains statement of the problem, justification, methodology, scope and limitation of the study.

The second chapter is review of stemming algorithms and its different techniques, methods of conflation and some related works in stemming algorithm for other languages. Chapter 3 entirely deals with *Afaan Oromoo* language in general and its morphology in particular. Word formation processes for different parts of speech will be discussed in this chapter.

The compilation of stopword lists and suffixes and techniques for developing stemmer are parts of chapter 4. The degree of complexity of *Afaan Oromoo* morphology in relation to other languages based on word frequency distribution is also discussed in chapter 4. Furthermore, the techniques for experimenting and results of the experiment from the test of the developed stemmer will also be presented in this chapter. Conclusions and recommendations constitute the last chapter of the report, chapter 5. And Bibliographies and appendixes are appended at the end.

CHAPTER 2

STEMMING ALGORITHM

2.1 Introduction

Among other things, stemming enhances the effectiveness of free-text retrieval system. The argument of Popovic and Willett (1992) in this regard is that, stemming allows the identification of matches between words in queries and documents. Al-Kharashi & Evens (1994) indicate that the motive for the use of stemming is to increase the number of retrieved documents since the stem of a term represents a broader notion than the original term itself. Similarly, Savoy (1993) reports that grouping of words that has the same root under the stem (or indexing term) will increase the success of matching of documents to a query in information retrieval. Such processes help to ensure that the search locates relevant terms that would otherwise be missed in the absence of stemming. Stemming is also used to reduce the size of index files in IR.

The general term that express the processing of morphological term variants is called conflation. It is the process of reducing words of similar semantics in to a single form. As it

will be shown in the next section, conflation could be manual or automatic operation. Stemming algorithm is the common automatic operation of conflation.

The operation of stemming algorithm mainly involves the removal of affixes from terms. As its name indicates, affix removal algorithms removes affixes from terms to obtain stem. The standard approach in this regard is to have a complete list of suffixes and then strip the longest possible match from the endings of a given term. Affixes are components (or parts) of terms that are required to suit grammatical requirements or other aspects of a language. In this regard, the morphological complexity of a language is one of the factors that is to be considered in developing a stemming algorithm.

The development of stemming algorithms based on affix removal for English could be easier than for other morphologically complex languages such as French (Savoy, 1993). As Savoy (1993) pointed out, for instance, the different categories of speech for French are gender and number –based. In addition, French has larger number of inflections than English. Obviously, more effort is required to develop stemming algorithm for languages with complex morphologies. In order to develop a stemmer therefore, a thorough understanding of the existing literature and the underlying techniques for developing a stemming algorithm is required.

2.2 Automatic Word Conflation

Conflation is the act of fusing or combining multiple words, which can be treated semantically equivalent and which do not necessarily be reduced to a common stem.

Ekmekcloglu et al (1996) define conflation as "a computational procedure that is designed to bring together words that are semantically related and to reduce them to a single form for retrieval purposes". This involves the process of the retrieval of approximately matching morphological variants. In terms of IR, word conflation has two functions. The summary of these functions is given by Willett (1988) and Lennon et al. (1981) (cited in Alemayehu, 1999: 18 – 19) as:

- Reducing the total number of distinct terms with a consequent reduction of dictionary size and updating problems; and
- Bringing similar words, having similar meanings to a common form with the aim of increasing retrieval effectiveness. This aspect has become much the more important as storage and processing costs have decreased over the year.

There are two main classes of conflation algorithm (Ekmekcioglu et al., 1996): stemming algorithm and string similarity algorithms. The first function stated above is handled by stemming algorithm and the second function could be handled by any of the two classes of conflation methods (Hendry et al, 1986, cited in Alemayehu, 1999: 19).

Stemming algorithm is a language dependent process that removes affixes from terms to reduce to a common stem. It is usually effected by suffix dictionaries that contain all possible term endings. By developing a stemming algorithm the process of truncation can be automated. The resulting stemmer from the algorithm takes in a word, breaks it down to its stem, and removes all the attached affixes. The function of a stemmer is therefore to conflate together all variants, which are semantically equivalent and share the same stem.

Unlike stemming algorithm, string similarity algorithm is language independent such that it can handle all types of variants. According to Ekmekcioglu et al., the most common of this approach of conflation are the *n-gram* match techniques. In an *n-gram*, a word is broken down to n-sized consecutive groups of characters. These sets of characters are then used to determine the similarities to other words.

Conflation can be performed manually (such as lexical look up) or automatically. Automatic conflation is performed mostly by stemming algorithm, which conflates morphological variants using different techniques like suffix removal. Lexical look up is a general method that can be used to conflate two words which are entirely different in form (Piarce, 1996), like "studying" and "learning" which are semantically identical. The manual process involves looking up of one of the words in the corresponding dictionary or thesauri. The approach is impractical due to the requirement of large table size to store large amount of terms.

The other approach of term conflation technique is the successor variety. In this approach a string is checked by adding a letter at a time to see for the probable successor words and also if there is a break or transition in the given word. A break found in this way implies that suffix might be attached to the word thus a stem is defined. The details of this and other conflation approaches are presented in the next sections.

2.2.1 Stemming Algorithm

The stemmer developed for one language may not function as it does for another.

Morphological analysis shows that languages in general differ from one another in their word formation process and thus in their affixes. Stemmers are therefore, language dependent, such that a stemmer for English does not work for Germany, or Arabic or *Afaan Oromoo*. Hence specific routines are required to develop a stemming algorithm for a specific language.

The existing stemmers have similar approaches although they differ in their practical implementations (Alemayehu, 1999). According to Alemayehu's argument, the most common approach in this regard is the use of dictionary of suffixes and the associated sets of rules on how to apply the suffixes. The manner of operation and context sensitivity (condition rules) are other characteristics of stemming algorithms (Alemayehu, 1999). These will be discussed in the next section.

2.2.1.1 Manner of Operation

A stemming algorithm can be either longest match or iterative. The longest match removes the longest suffixes that match the suffixes of the given term. In the case of iterative algorithm, suffixes are removed iteratively.

Basically, iteration depends on the order in which suffixes are attached to the stem. To perform this process suffixes are organized in certain order into classes. Savoy (1993) indicate that, suffixes are classified based on the deviation rules, like plural inflections and terms with suffixes such as "-ed" and "-ing" may constitute the last order class. In most

cases, the inner or previous order class is derivational suffix. For example the word ''EXCEPTIONS' is formed from the stem 'EXCEPT' and suffixes –ION and –S. In this example, the suffix –S constitutes the last order class that inflects the stem to plural form. Obviously, the suffix –ION forms the first order class and it is derivational suffix that transforms the verb 'EXCEPT' to a noun 'EXCEPTION'. But, this order may not always be assumed. There are exceptions in which inflectional suffix precedes the derivational ones. In DISINTERESTEDNESS, for instance, the ending, -EDNESS is composed of two suffixes: –ED and –NESS. The suffix –ED is an inflectional suffix while –NESS is a derivational suffix that changes the verb to adjective.

An iterative-stemming algorithm is simply a recursive procedure intended to remove strings in each order-class one at a time backward starting from the end of the word. In doing so, within a single order-class, only one match is allowed, but the algorithm of iterative stemming can have more than one order-class. In all cases it is the programmer who determines the order, the number of order-classes and contents of each order-class.

Regarding the longest-match principle, Lovins (1968) states that, "within a given class of endings, if more than one ending provides a match, the one which is longer should be removed." In order to implement this principle, endings in any class is stemmed in the order of decreasing length. The algorithm for the longest-match operation uses only one order-class. The affixes are therefore, ordered in the list in the order-class based on length. The concatenated suffix –EDNESS of the term DISINTERESTEDNESS is removed from the stem in a single process by the longest match algorithm. The removal of the same ending by iterative algorithm is done in two separate stages. The algorithm removes the outer

suffix –NESS first and then removes –ED in the second stage. In a longest match algorithm, the removal of the longest possible suffix (ending) always precedes other shorter ones, such that no extra effort is required.

These two manners of operations of stemming algorithms discussed above will come up with some disadvantages. In the case of the longest-match, one may require generating, all possible combinations of affixes, and consequently, a storage space (although it may not be a critical problem of today's computers) is required to store the respective endings (suffixes).

Similarly in applying iterative algorithm, one needs to examine a large number of endings, which is a laborious activity. Moreover, a number of complications are introduced into the preparations of the list of order classes and into the programming of the modules of the iterative algorithm. However, a shorter list of endings is required as compared to the longest match.

2.2.1.2 Context Sensitive and Recording Rules

The manner of operation of suffix removal discussed above can be augmented by rules that refer to context to improve accuracy of stemming procedure. A stemming algorithm can be context free or context-sensitive (Savoy, 1993). Context free implies the absence of qualitative or quantitative restrictions on the removal of endings. It is suggested that conflation processes can be improved by introducing context-sensitive rules to the stripping algorithm. Context-sensitive therefore, implies the addition of constraints to the

Another alternative to recording suggested by Lovins (1968) is the application of partial matching. The result from partial matching is similar to that of the recording rules, but it is not part of the stemming algorithm. According to Lovins, partial matching operates on the output from stemming routine during searching and therefore, it is a separate routine.

2.2.1.3 The Suffix Dictionary

Stemmers use dictionaries of suffixes together with a set of rules to to remove affixes from stems. In the suffix removal procedure, the ending of a term is checked for a match in the dictionary. If match is found, the stemmer strip off suffixes according to the corresponding rules. In the absence of a match for the ending in the dictionary, the input term is returned as a stem. Stemmers that have implemented suffix dictionaries include Lovins (1968), Dawson (1974), and Porter (1980) for English language. In addition to the suffix dictionary, some stemmers also use stem dictionaries to validate the result of stemming.

2.2.2 Successor Variety

For a given text, successor variety stemmer uses the frequencies of letter sequences as the basis of stemming. Hafer and Weiss (cited in Frakes, 1992: 134) define the technique of successor variety as:

> Let $\propto$ be a word of length n; $\propto i$ is a length i prefix of $\propto$. Let D be the corpus of words. D$\propto i$ is defined as the subset of D containing those terms whose first i

letters match $\propto i$ exactly. The successor variety of $\propto i$ denoted $S\propto i$ is then defined as the number of distinct letters that occupy the $i + 1$st position of words in $D\propto i$. A test word of length n has n successor varieties $S\propto 1$, $S_{\propto 2}$, ..., $S\propto n$.

The following example is considered to illustrate the technique of successor variety.

Text word: NATIONAL

Corpus: NATION, NATIONALIZE, INTERNATIONAL, INTERNATIONALIZE, SESSION, TOTAL, GENERAL, NATION, NORMALIZE.

In order to determine the successor variety for the given word NATIONAL, we have to examine for each letter in the word from the corpus. The successor variety of "N" of NATIONAL, for example is "A", and "O". Similarly, the next successor variety for NATION would be "T", for "NA." in the corpus. Using the same procedures, the successor variety for all characters can be determined. The complete list is given in Table 2.1.

TABLE 2.1: LIST OF POSSIBLE SUCCESSOR VARIETIES

Prefix	Successor variety	Letters
N	2	A, O
NA	1	T
NAT	1	I
NATI	1	O
NATIO	1	N
NATION	1	A
NATIONA	1	L
NATIONAL	1	I

Frakes (1992) indicate that, by using a large body of text, the successor variety sub-strings of

a term will decrease as more character is added until a segment boundary is reached. It is at this point where the successor variety will sharply increase. This information is then used to identify the relevant stems. In order to determine the stems therefore, segmentation is to be performed based on one of the following methods discussed by Hafer and Weiss (cited in Frakes, 1992: 134 – 135).

1. Cutoff: the boundary of the segment occurs when successor variety jumps or whenever cutoff value is reached. However, it is difficult to determine the cutoff value as too small value results in incorrect cuts and too large value misses correct cuts.
2. Peak and plateau: a segment break is made after character whose value is higher than that before or after it.
3. Complete word: a break is made after a segment if the segment is a complete word in a corpus or word list.
4. Entropy: this method uses advantage of the distribution of successor variety letters and complete change at each position and apply threshold to determine the cutoff.

Using complete word segementation method, NATION and AL are found to be segements of NATIONAL. In this case, NATION is a complete word in the corpus, but AL is not a complete one. The term NATION is thus selected as a stem from the segments.

In general, the successor variety stemming method has three parts (Frakes, 1992): (1) determine the successor variety stemming for a word, (2) use this information to segment the word using one of the above methods for segmentation, and (3) determine the stem using one of the appropriate segmentation methods. As Frakes (1992) discussed, the aim of the

method is to develop a fully automated stemming process that requires little or no human interaction. In contrast, the suffix removal operation requires human interaction to prepare suffix lists and removal rules, but more perfect.

2.2.3 *n – gram*

The *n-gram* method conflates terms based on the number of digram or *n-gram*. The *n-gram* stemmer is a procedure for comparing terms to find word similarity. The *n-gram* is a set of consecutive letters in a term, whose common value is two (*di-gram*) or three *(tri-gram)*. The basic principle behind this approach is that, similar words could have high proportion of *n-grams* in common. To compare two terms using *n-gram* technique, first, the terms are broken down to a pair of consecutive letters called *di-grams*. The similarity is then calculated between pairs of terms based on shared unique *digram*s. The approach may be better explained in the following illustration.

Nationalize ⇨ na at ti io on na al li iz ze

Unique *digram* = al at io iz li na ti ze

Nationality ⇨ na at ti io on na an ni it ty

Unique *digram* = al at io li na ti ty

In this example, out of the *d-igram*s for the 'nationalize', eight of them are unique. Similarly, seven of the *digram*s for the word 'nationality' are unique *di-gram*s. By inspection, the two words have six *digram*s in common to share: al, at, io, li, na and ti.

After determining the unique *digram* computation for the word pair, similarity between the words will be calculated using similarity coefficient. The commonly used similarity coefficient is that of Dice. It is stated as:

(2.4) $$S = 2C/(A + B)$$

Where A & B are the number of unique *digrams* in the first word and in the second word respectively, and C is the number of shared unique *digrams* by the first and the second words. Having obtained the similarity values for all pairs of terms in the database, we have to organize them into similarity matrix. The terms are now clustered using a single link clustering method from the similarity matrix.

Although, the *n-gram* stemmer is language-independent, it may fail to conflate words for some languages (Alemayehu, 1999). On one hand, two terms can have a common stem without having any shared *d-grams*. On the other hand, different words with different meanings may be conflated together because of similarities in their affixes.

n-gram matching is as effective as stemming algorithm in conflating terms for Turkish (agglutinative language)(Ekmekcioglu et al., 1996). But, stemming algorithm is commonly used for many other languages.

2.2.4 Stemming Algorithm: A Review

Lovin's (1968) stemmer and Porter's (1980) stemmer are the two commonly cited

English stemmers based on suffix removal algorithms. Lovin's stemmers use a longest match algorithm and exception list to remove large list of endings, which are over 260 different suffixes. The Porter stemmer on the other hand uses iterative algorithm that remove smaller numebr of suffixes (about 60) in a number of steps approach and a few context-sensetive recording rules. Each step performs either removing suffixes or transforming roots based on rules for truncation. In the course, short-suffixes are removed without exception.

Other languages for which stemmers have been developed include Malay (Ahmad et al., 1996), Sloven (Popovic &Willett, 1992) and Arabic (Al-Kharashi & Evens, 1994). The stemmer developed by Ahmad et al (1996) uses rules that define prefixes, suffixes, infixes, and prefix-suffix pairs. The input words to the stemmer were first checked against the root words in dictionary in order to avoid stemming operations on words which are already stems. After stemming the conflated terms are once more checked against the roots in the dictionary.

For a better understanding of stemming operation, Porter's stemmer is selected for detailed discussion. This stemmer provides a good average recall and precision values (Croft, 1999). It is based on a measure of vowel-consonant sequences and its algorithms work on a set of conditions rule. The condition of Porter's stemmer is divided into three classes: conditions on the stem, conditions on the suffix, and conditions on the rules. The corresponding examples of the conditions are given below.

1. The vowel-consonants sequence refers to the number of appearance of the pair in that order. Denoting C for a consonant sequence and V for a vowel sequence, the possible

combinations are:

(2.1) CVCV C

CVCV V

VCVC C

VCVC V

Porter (1997) generalizes these as:

(2.2) [C] VCVC...[V],

The square bracket indicates the arbitration of the enclosed character. With *m* stands for the number of times $(VC)^m$ is repeated in a word, the expression in (2.2) can be written as,

(2.3) $[C] (VC)^m [V]$

where *m* is called a measure of any word or part of a word in this form.

Conditional rules are set based on the values of *m*. Examples of measures for a term is given in Table 2.3 below.

TABLE 2.2: EXAMPLES OF MEASURES OF TERMS

Measure	Example
m = 0	TR, EE, TREE, BY
m = 1	TROUBLE, OATS, TREES, IVY
m = 2	TROUBLES, PRIVATE, DATEN, ORRERY

However, some of the outputs of the Porter stemmer lacks linguistic meanings, e.g., the variants INDUSTRIAL, INDUSTRY, INDUSTRIALIZED, etc., are conflated to INDUSTRI, which have no lingustic validation.

The following are 'conditions' set for Porter's stemmer.

2. * <S> - the stem ends with a given letter S.

3. *v* - the stem contains vowel

4. *d the stem ends in double consonant

5. * o – the stem ends with a constant vowel – constant sequence, where the final constant is not w, x, or y

6. the Boolean operators AND , OR and NOT are used in stem conditions, like (m>1) and (*S or *T) tests the stem with measure greater than 1 with ending of S or T.

The suffix conditions of the Porter's algorithm take the form:

(Condition) S1 → S2

This implies that, if a word that ends with the suffix S1 satisfies a given condition, S1 is automatically replaced by S2.

The rules are divided into a number of steps (five major steps) each contains a set of rules stated one after the other. These rules are obeyed in sequence, such that only one of

the rules is obeyed from a given set. Since the steps are ordered with suffixes of the longest match first, the longest possible match is always removed. The algorithm is given as follows (Frakes, 1992: 140):

```
{
step1a (word);
step1b (stem);
if (the second or third rule of the step1b1 (stem);
        step1c (stem);
step 2(stem);
step 3(stem);
step 4(stem);
step5a (stem);
step5b (stem);
}
```

The rules for step1a of this algorithm are given in Table 2.3 below.

TABLE 2.3: STEP1A RULES OF PORTER'S STEMMER

Conflation	Suffix	Replacement	examples
NULL	ssee	ss	caresses → caress
NULL	ies	i	ponic → poni
			ies → ie
NULL	ss	ss	caress → caress
NULL	s	null	cats → cat

Although Porter' stemmer is generally regarded as simple and effective, it has some limitations. The stemmer fails to conflate irregulated plurals such as women, teeth, etc. It does not also work much with words ending in 'y' or 'able', as in sleepy and doable. These words are stemmed to sleepi and doabl, respectively instead of sleep and doable.

2.3 Stop word List

Stoplist compilation is one of the schemes for improving the efficiency of stemming procedures. Stop words are words that are considered fewer worthies in representing documents or for index terms in a database. Commonly, the high proportions of most texts contain such non-relevant words. In English, for instance, definite article, conjunction, preposition, personal pronoun, possessive pronoun, etc. are considered non-significant words and are included in the stoplist in IR system.

Savoy (1999) gives two major reasons for developing stop word list. First, effective retrieval is based on the extent of the matches between a query and a document that in turn depends on good indexing terms. The exclusion of words such as, "the", "be", "in", "he", etc from representing a document during retrieval thus establishes an effective searching. Second, the resulting reduction of size of the inverted file after exclusion of stop words enhances retrieval effectiveness.

There is no theoretical standardized foundation or methodology for developing stop word list. Nevertheless, we need to follow certain guidelines for effective results. Rijbergen (1975) suggests Luhn's principle of measuring word significance as a general method for determining stop words. From the frequency analysis of words, Rijbberger generates a stop list. The higher the frequency of occurrence of a word the lower the significance of the word in representing a document. In this case, the most frequently and the least frequently occurring words are considered less significant and therefore included in the list of stop words.

This technique however introduces a risk of loss of performance. If an item is needed for a query, recall will be reduced. Similarly, precision can be also affected as a result of removing some words from queries. Salton (1983) (cited in Alemayehu, 1999), in this regard wrote that, the removal of high frequency words could reduce recall and that of low frequency words could bring losses in precision. As a result, alternative techniques of term weighing have been suggested as described briefly in Chapter 1.

The exclusion of stop words from a document results in substantial reduction of file size. In general, the exclusion of stop word from further processing reduces the size of the document by 30 to 50% (Rijsbergen, 1975; Savoy, 1993).

2.4 Chapter Summary

In this chapter, the basic principles of automatic word conflation, specially, stemming is discussed. In particular, the manner of operations: the longest-match suffix removal and iterative algorithm and the common conflation approaches: successor variety, affix removal and *n-gram* approaches are dealt in depth. Review of stemming algorithms and the techniques for compiling stoplist are also discussed. Stemming is a language dependent procedure. This is because the affixation rules are different for different languages. In the following chapter, the *Afaan Oromoo* language is discussed.

CHAPTER 3

THE AFAAN OROMOO LANGUAGE

3.1 Introduction

Afaan Oromoo is the language of the *Oromo* people who comprises the largest ethnic group in Ethiopia. The 1994 census report of the Central Statistics Office for instance, estimated *Oromos* to be 18,732,525[1] (about 32% of the total population of Ethiopia). Other than in Ethiopia, several clusters of *Oromo* communities live in Kenya, Sudan and Tanzania (Tilahun, 1994). A number of evidence (e.g. Bender et al, 1976: 130; Stroomer, 1987:1) indicate that in terms of the number of speakers and geographical size, *Afaan Oromoo* is one of the few important African languages. This was written by Grover Hudson (cited in Stroomer, 1987:1) as "*Afaan Oromoo* is one of the five or six most important languages in Africa." Likewise, according to Gragg (1976), *Afaan Oromoo* is the third largest language in size and number of speakers in Africa, next to Arabic and Haussa.

Linguistically, *Afaan Oromoo* belongs to the Cushitic branch of the Afroasiatic language family along with Somali, Afar and a number of other languages (Stroomer, 1987). Further,

[1] The census did not cover major areas of Misrak Hararge of the Oromyia Region

Afaan Oromoo is classified as a lowland east Cushitic sub-family (Bender (ed.), 1976: 3). Traditionally, the natives, especially the elders refer to the language as *Afaan Oromoo*, while the non-natives designate it simply as *Oromoo*. It is believed that, the former is more acceptable since it is indigenous and the people's self designation (Bender et al, 1976). Hence, *Afaan Oromoo* is preferred for use in this thesis.

There is a dialectical variation in *Afaan Oromoo*. According to previous studies in this regard (for example, Gragg (1976)), four major categories can be identified. These are: Western (Wellega, Iluababor, Kaffa and parts of Gojjam), Eastern (Harar, Eastern Showa, and parts of Arsi and Bale), Central (Central Showa, Western Showa and possibly Wollo) and Southern (parts of Arsi, Sidamo and Borena).

With regard to written *Oromoo*, the Roman (Latin) alphabet has become the official script for Afaan Oromoo since the end of 1991 (Tilahun, 1994). Currently, the language is being used as the official language of the Regional State of *Oromyia*. Furthermore, *Afaan Oromoo* is the medium of instruction for Primary Schools and Junior Secondary Schools of the Region.

3.2 The *Oromo* Alphabet: *Qubee Afaan Oromoo*

The alphabet of *Afaan Oromoo* is often called *Qubee Afaan Oromoo,* alphabet of the *Oromo* language. Like any natural language, *Qubee Afaan Oromoo* has consonants and vowels.

3.2.1 *Afaan Oromoo* Consonants

There are about 26 consonants in *Afaan Oromoo.* The simplified constant symbols as used in writing are given in Fig. 3.1 below.

FIG. 3.1: AFAAN OROMOO CONSTANTS

Stop	Labial	Alveolar Dent	Palatal	Velar	Glottal
Voiceless	(p)	t	ch	k	?
Voiced	b	d	j	g	
Glotalized	ph	x	c	q	
Implosive		dh			h
Spirant (voiceless)	f	s	sh		
Voiced	(v)	(z)			
Nasal	m	n	ny		
Sonant		r l			
Glide	w		y		

Note:

(a) The consonant p, v, and z only occur in loan words.

(b) Consonants (*dubbifamaa*) may occur as ungeminated (*laafaa*) or geminated (*jabaa*) to form different words: cf. *baruu* 'to learn' and *barruu* 'palm of hand'.

3.2.2 *Afaan Oromoo* Vowels

Afaan Oromoo has ten vowels, five short and five long.

FIG. 3.2: AFAAN OROMOO VOWELS

short				long			
i			u	ii			uu
	e		o		ee		oo
		a				aa	

The corresponding equivalent pronunciation of English as used in *Afaan Oromoo* is given in Appendix I.

3.3 Morphology

Morphology studies how words are formed in a language. Silzer (1999) defines morphology as "the study of the structure of words". These definitions show that words have different structures that may be derived from other words by changing the basic class or category of words or by inflecting to indicate number, time (past, present or future), gender, etc.

Each language has its own morphological structure that defines rules used for combining the different components the language may have. The English language for instance is basically different in its morphological structure from French, Arabic, or *Afaan Oromoo*. In view of this, studying the morphology of *Afaan Oromoo* is important in order to deal with the computational aspects of the language.

3.3.1 Kinds of Morphology

Morphology can be classified as inflectional or derivational.

3.3.1.1 Inflectional Morphology

Inflectional morphology studies the inflectional changes in words that generally do not result in changing the classes of words. Rather, the inflectional changes indicate tense (present, past, far past, future), number (singular, plural), gender/class (masculine, feminine, neuter), person (first, second, third), etc. Illustration is given in (3.1) below.

(3.1) present → past : look → looked

singular → plural: car → cars

male → female: actor → actoress

1st/2nd person → 3rd person: give-s → gives (e.g., he/she gives, they give, you give)

As indicated in (3.1), inflectional morphology deals with the combination of stems with grammatical markers of suffixes such as –s, -ess, -ed and –ing, in English for example. Generally inflectional morpholgy is very productive as all nouns have singular/plural distinctions, most verbs have tense distinctions, etc.

3.3.1.2 Derivational Morphology

Derivational morphology results in changing classes of words, for example, a noun may be derived from a verb, or an adjective can be derived from a verb (see examples in 3.2).

(3.2)

(i) create (v) + ive (adjective marker) → "creative" + ity (noun marker) → "creativity" (n)

(ii) nation (n) + al (makes adjective) → "national"(adj.) + ize (makes a verb) → "nationalize"(v) + ation (makes a noun) → de + "nationalization" (n) → "denationalization"

(3.3) (a) nouns to adjectives:

(i) affection-ate → affectionate, (ii) girl – ish → girlish

(b) verbs to nouns:

(i) sing – er → singer, (ii) clear → clearance (iii), locate – ion → location

(c) nouns to verbs:

(i) vaccine – ate → vaccinate

(ii) brand – ish → brandish

In the next sections, the morphology of *Afaan Oromoo* will be introduced. As we will see, the inflectional and derivational morphologies in the language are productive. [For the detailed treatment of the subject, see Gragg, 1976; Temesgen, 1995; Bender et al., 1976; Kebede, 1996 and Heine, 1981.]

3.4 The *Afaan Oromoo* Morphology

A morpheme is the minimal linguistics unit of a language that carries a meaning (Silzer, 1999). Hence, a morpheme can not be further decomposed into a meaningful unit. There are two broad categories of morphemes (Schiffman, 1999): free and bound

morphemes. A free morpheme can stand as a word on its own likes: "work", "live", and "man". Those morpheme, which do not occur as a word on their own can be affixes or a roots to which an affix or affixes can be attached (Wardhaugh, 1977). In *Afaan Oromoo* roots are bound as they can not occur on their own like in *dhug-* 'drink' and *beek-* 'know', which are pronounceable only when other completing affixes are added to them. In other words, these roots serve as base stems in *Afaan Oromoo* since they posses non-verbalized glosses (meanings).

Like the root, an affix is also a morpheme that can not occur independently. It is attached in some manner to the root, which serves as a base. These affixes are of three types – prefix, suffix, and infix. The first and second types of affixes occur at the beginning and at the end of a root respectively in forming a word. In *beekumsa* 'knowledge', for instance, - *umsa* is a suffix and *beek-* 'know' is a stem. An infix is a morpheme that is inserted within another morpheme. Like English (Wardhaugh, 1977), *Afaan Oromoo* does not have infixes as far as I could ascertain from the existing literature.

The only prefix in *Afaan Oromoo* is *(hi) ni.* It occurs usually as *hin* or *ni* in texts. This prefix is used to change a root to a positive, negative or to interrogative based on falling and rising of stresses on the vowel. Examples of this kind are *hín dèèmnè* 'we went' and *hìn dèèmnè* 'didn't go'. As it is indicated in the examples, and used in other texts, *(hi)ni* is not attached to the word they express unlike other suffixes. However, on the other hand there are evidences indicating the assimilation of *(hi)ni* with stems. Examples are *hingala* 'he will be returned' *and himbara* 'he learns'. Further study is thus required to give detailed explanation for the variations in the usage of the prefix.

TABLE 3.1: EXAMPLES OF AFFIXATION PROCESSES IN *AFAAN OROMOO*

Word	affix	Stem
dhug-aatii 'drink'	*- aatii-*	*dhug-* 'drink'
kuf-aatii 'falling down'	*- aatii-*	*kuf-* 'fall down'
keenn-aa 'gift'	*- aa-*	*keenn-* 'give'
kadh-aa 'begging'	*- aa-*	*kadh-* 'beg'
haam-tuu 'harvesting tool'	*- tuu-*	*haam-* 'harvest'
nyaat-tuu 'eater'	*- tuu-*	*nyaat-* 'eat'
deem-uu 'go or to go'	*- uu-*	*deem-* 'go'
bar-siis-uu 'to teach'	*- sis- uu-*	*barsiis-* 'teach'
har-ata 'broom'	*- ata-*	*har-* 'sweep'
hidh-ata 'armament'	*-ata-*	*hidh-* 'tie'

Broadly speaking affixes are of two types: inflectional and derivational. As we shall see latter on, stems/roots in *Afaan Oromoo* are bound morphemes, such that they require inflectional or derivational suffixes to form stand-alone words.

3.5.1.1 Inflectional affixes

Inflectional affixes can be a vowel or a consonant.

3.5.1.1.1 Vowel Affixes

There are three common inflectional affixes in *Afaan Oromoo*: *–a, e-* and *–i.* According to Kebede (1996), the suffixes *–a, e-* and *–i* are markers for imperfective (imperf), perfective (perf) and singular imperfective (imperf) respectively. These vowel suffixes are added at the final position of the verbal word. In addition to these three suffixes, Kebede also introduced a zero *(-ϕ-)* vowel suffix as a first or third person marker. However, such marker is not

normally used in writing system and therefore it will not be considered for developing the respective stemming algorithm. Examples given in (3.6) by Kebede may illustrate the inflectional suffixes. (Note: the zero (*-ϕ-)* vowel suffix is ignored in (3.6), (3.7), and (3.8)).

(3.6) a. *deem-a* → *deema*

'go' – 3sgm –imperfect 'he/it (will) go/goes'

b. *deem-e* → *deeme*

'go' – 3sgm – perf 'he/it went"

c. *deem-i* → *deemi*

'go' – 2sgm – imerf 'go'

Other examples of vowel suffixes are given in (3.7) and (3.8) below.

(3.7) a. *beek-a* → *beeka*

'know' –3sgm – imperfect 'he knows'

b. *beek-e* → *beeke*

'know' – 2sgm – perf 'he knew'

c. *beek-i* → *beeki*

'know' – 2sg – imperf 'know'

(3.8) a. *dhug*-a → *dhuga*

'drink' – 3gm – imperf 'he drinks'

b. *dhug-e* → *dhuge*

'drink' – 3sgm – perf 'he drunk'

c. *dhug-i* → *dhugi*

'drink' – 2sg – imperf 'drink'

3.5.1.1.2 Consonantal Afffixes

A) Verb

The inflectional consonantal afffixes are used to mark person in a verb. Kebede (1996) identified the suffix *–n*, for the first person plural marker and the suffix *–t* for the second person singular marker. The consonat suffixes generally preceds the vowel suffixes when they occur in a single word. Illustrative examples are given in (3.9) and (3.10) below. In these examples, the vowel *'a'* is (an imperfect tense marker) affixed to the terms to give a stand alone words.

(3.9) a. *deem-n-a* → *deemna*
'go' – 1pl – imperf 'we (will) go'

b. *deem-t-a* → *deemta*
'go' – 2sg- imperf 'you (will) go!'

c. *beek-t-a, bee-tta* → *beekta*
'know' – 2sg – imperf 'you know'

d. *beek-n-a* → *beekna*
'know' – 1pl – imperf 'we (will) know'

e. *kuf-n-a* → *kufna*
'fall' – 1pl – imperf 'we (will) fall'

As Kebede argues in (3.10) below, when *-n* follows a root that finishes in palatal glide, *-n* becomes *-nny* or *–nn* depending on dialectical variation. Further analysis of (3.10) shows that, some or all of the middle vowels are also changed (e.g., *-a* is replaced by *-e*) in the

inflectional process.

(3.10) a. *taa-ny-a* → *tennya/tenna*

'sit' – 1pl – imperf 'we (will) sit'

b. *dhagay-n-a* → *dhagennya/dgagenna*

'hear' – 1pl – imperf 'we (will) hear'

c. *kaay-n-a* → *keenny*

'put' – 2sg – imperf 'we (will) put'

B) Noun

In *Afaan Oromoo*, there are two genders (feminine and masculine) and two numbers (singular and plural). Examples include *nama* 'man' (singular) *nam-oota* 'men' (plural), and *harree* 'donkey' (singular) *harr-oota* 'donkeys' (plural). For gender we may have, *jaar-saa* 'old man' (masculine) and *jaar-ttii* 'old woman' (feminine).

In addition to masculine and feminine genders, Bender et al (1976) identifies another special form of number indicator called singulative gender that refers to 'exactly one'. Such variants are usually formed by addition of the suffix *-icha* to a noun for masculine and *-itti* for feminine as in *leenc-icha* 'that lion' and *leenca-tti*i 'the lioness'.

In most cases, there is a clear distinction between singular and plural in *Afaan Oromoo*. The following suffixes are used to form plurals.

(3.11)

a) *-oota, -ota*

The suffix *-oota* is used when the vowel that precedes the last consonant is short and *-ota*

when the vowel is long as in the following examples.

man-a	'house'	*man-oota*	'houses'
fard-a	'horse'	*fard-oota*	'horses'
wuc-ii	'chicken'	*wuc-oota*	'chickens'
ilkaa-n	'tooth'	*ilkaan-ota*	'teeth'

b) *-oolii, -olii, -oolee, -olee*

There is a general indication that the conditions/rules in (a) above is true for cases in (b) too. See the following examples.

	farda 'horse'		*fard-oolii* 'horses'
	jabbii 'calf'		*jabb-oolii* 'calves'
	wucii 'chicken'		*wuc-oolii* 'chickens'
	gaangee 'mule'		*gaang-olii* 'mule' (plural)

c*) –een*

Examples:	*mana* 'house'		*mann-een* 'houses'
	muka 'tree'		*mukk-een* 'trees'

Note that, the last vowel in these examples is removed and the preceding consonant are duplicated in the process of suffixation.

d) *(-awwaa)n:*

akkoo 'grandmother' + *-awwaa(n)* → *akkoo-wwaan* 'grandmothers'

obboleessa 'brother' + *awwaa(n)* → *obbolee-wwaan* ' brothers'

rakkoo 'problem' + *awwaa(n)* → *rakkoo- wwaan* 'problems

e) *–eetii: muka* 'wood' → *muk-eetii* 'woods'

f) *–eyyii: sooressa* 'rich man' → *soor-eyyii* 'rich men/people'

g) *–ii:* This type of suffix occurs with change of stem as in *korma* (sg.) → *koromm-ii* (pl.). (The term *korma* is usually used to designate male cow and male bull)

h) *–oo: simbira* 'bird' *simbir-oo* 'birds' (*simbirr-oota* is also possible)

The first two alternate suffixes *–ota* and *-oota* are most commonly used for changing nouns to their plural forms. The former occurs when the vowel of the preceding syllable is short and the later occurs when such vowel is long. Only few nouns use the other suffixes and they are not much productive. According to Gragg (1976), these suffixes are limited to inflect only few nouns in their respective group.

Bender et al (1976) treat the noun plurals and case-forms in *Afaan Oroomo* as the accusative (direct object form). The following examples illustrate the noun, *saree* 'dog' with its different forms in this regard.

(3.12)

	Accusative (citation form)	nominative (subject form)
singular	*saree* 'dog'	'*saree-n(i)* 'the dog'
plural	*sar-oota* 'dogs'	*sar-oonni 'those dogs'*
singulative	*sar-icha* 'that dog'	sa*a-ittii 'that (bad) dog'*

Note that the final vowels in the above examples are removed before the term is inflected. In this way *–a* from *nama* 'man' is dropped before *–oota* is affixed to the term.

Definiteness

In *Afaan Oromoo*, the indefinite meaning is often indicated by using the numerical *tokko* 'one' as in many other languages (Bender et al, 1975). Examples: *Sanga tokko* 'an ox', *naadheen tokko* or simply *naadheen* 'a woman'. There are demonstratives that impart definiteness to the nouns. In *nittiin kun* 'this woman', the demonstrative *kun* is used to address that particular woman. Although such demonstratives and the numerical *tokko* indicate numbers, they can not be treated as affixes. Therefore, the stemmer that will be developed for *Afaan Oromoo* will not handle them.

The suffixes, *-icha* and *–ttii* are also used to indicate the notion of definiteness:

nama 'man' → *nam-icha* 'the man' and *intaala* 'girl' → *intala-ttii* 'that girl'.

Basically, adding *-ni* to a noun forms a subject. Examples are given below.

(3.13)

Citation form	subject form	meaning
sangaa	*sangaa-ni*	'ox'
muka	*muk-ni*	'wood'
miila	*miilla-ni*	'leg'
boobaa'a	*booba'aa-ni*	'fuel'

C) Adjectives

Similar to the noun formation, adjectives in *Afaan Oromoo* are inflected for gender and number. It can be shown that *–sa, -aa, -tuu, - aawaa, ooftu,-* and *–oo* are gender sensitive

adjectival endings.

Gender

The gender in *Afaan Oromoo* is more consistently marked in adjectives than in nouns. The most usual patterns of this kind are presented in Table3.2 below.

TABLE 3.2: INFLECTED ADJECTIVES

Suffix	Example	Meaning
-sa	*gog-eesa(m)*	dry
eettii	*gog-eettii(f)*	
-aa	*jiidh-aa (m)*	wet
-tuu	*jii-tuu (f)*	
	bareed-aa(m)	beautiful
	bareed-uu (f)	
-aawaa	*mach-aa (m)*	drunk
ooftuu	*mach-ooftuu (f)*	
-aa	*xiin-aa (m)*	small
-oo	*xiin-oo (f)*	

It should be noted that there are adjectives which are invariant for gender (Gragg, 1976). These include adjectives ending in *-ii, -ee, -a, -oo, -uu* and *-aa*. Examples: *dheedhii* 'raw', *gadhee* 'bad', *janna* 'brave', *dhiyoo* 'near', and *haaraa* 'new'.

Number

A significant number of the plural forms of adjectives are formed by reduplication of first syllable. The following are examples of adjectives formed in this way.

(3.14)

M.sg.	M.pl.	F.sg.	F.pl.
xinnaa	*xix̲innaa* 'small'	*xinoo*	*xix̲innoo*
jabaa	*jajjabaa* 'strong'	*jabduu*	*jaj̲abduu*
gudda	*guguda* 'big'	*guddoo*	*gugudoo*

In these examples, the patterns used in the plural formation for each of the adjectives are different from one another that mark the irregularity of adjective formation process.

Definiteness

Like nouns the suffixes *-(t)ihca* and *-ittii* are used to express definiteness in adjectives. *E*xamples are *mucca, mucc-icha* 'that baby boy' and *mucca-ttii* 'that baby girl'. These suffixes are also used to express definiteness in nouns as shown in the previous sections.

3.5.1.2 Derivational Suffixes

Different classes of words can be formed by adding derivational suffixes to a root or stem and to other classes of words.

3.5.1.2.1 Derived Verbs

A verb can be formed by adding derivational suffixes to another root or stem verb, noun or adjective. Different classes of derived verbs inclde benefactive/reflexives, causative, statives, or passives.

A) Benefactive/reflexive

The suffix *-at* is added to a root to give reflexive-middle or benefactive/reflexive. Basically,

-at incorporates meaning, which is "to do something for oneself" to the derived benefactive/reflexive root or stem verb. The suffix *–at* is used to derive a verb stem from roots of all major categories of speech for all persons except for first person singular (Kebede, 1996). In this case, *–at* becomes *–addh* for the first person singular. Common examples of this type of derivations are given in the Table 3.3 below.

TABLE 3.3: EXAMPLES OF BENEFACTIVE/REFLEXIVE

Base	Suffix	Derived word
kadh- 'beg'	*-at*	*kadh-at* 'beg for oneself'
dubb-'speak'	*-at*	*dhubb-at* 'speak for oneself'
ajjes- 'kill'	*-at*	*ajjees-at* 'kill for oneself'
bit- 'buy'	*-at*	*bit-at-* 'buy for oneself'
diq- 'wash'	*-at*	*diq-at* 'wash oneself'
haad- 'shave' *-at-*	*-at*	*hadd-at-* 'shave for self"

The *t* in *dubbat-* of the above example is often palatalized to *–ch* and *–uu* by adding to the root as indicated by Kebede (1996). [see (3.15) below]

(3.15) *dubbat-uu* → *dubbachuu* or *dubbatuu*

'speak' 'speaking / to speak'

Furthermore, Kebede (1996) illustrates the different forms of the stem dubb- 'speak' for different types of persons as shown in Table 3.4

In the following examples the causative is derived from base of adjectives or nominals.

(3.17) a. *malaa-s-* → *malaas-*

'pus' 'discharge pus'

b. *gudd-is-* → *guddis-*

'big' 'make big'

c. *furd -s* → *furdis*

'fat' 'make fat'

d. *dheer -s-* → *dheeres-*

'tall' 'make tall'

e. *jab -s-* → *jabees-*

'strong' 'make strong'

(i) As it is shown, *-a* and *-at-* are being lost when causative suffix is attached to a base. Further inspection reveals that, vowels that appear at the end of the adjectival base are removed to form causative, as in c, d, and e of (3.17) above.

In (3.18) below, the constant *t* is changed to *–at*

(3.18) a. *nyaat-* 'eat'*-sis* → *nyaachis* 'make eat'

b. *tabat-*'play' – *sis* → *tabachis* 'make to play'

Other examples of causatives which are derived from verbal roots and nominals are given in (3.19) and (3.20).

(3.19) a. *fid* - 'bring'– *sis* → *fichissiis* 'make to bring'

b. *fix*- 'finish' – *sis* → *ficcisiis* 'make to finish'

c. *lix-'enter'* – *sis* → *liccisiis* 'make enter'

The last consonant of the stem in (3.18) and (3.19) is replaced by *–chi* or *–cci* based on the respective syllable.

(3.20) a. *dubb*-'talk'-*at-siis* → *dubbachiis* 'cause to talk'

b. *but* -'snatch'– *at – siis* → *buttachiis* 'to snatch'

c. *gaaf*-'request' *–at – siis* → *gaafachiis* 'make to request'

As it is illustrated in (3.20), the vowel(s) after the last consonant in the stems are replaced by the vowel *a* and the suffix *–siis* becomes *-chiis*.

C) Stative/Benefactive

The suffixes *-a*? and *–om* are added to an adjectival or nominal bases to derive a stative verbs in *Afaan Oromoo*. Temesgen (1995) gives the following examples.

(3.19) a. *fid* - 'bring'– *sis* → *fichissiis* 'make to bring'

b. *fix*- 'finish' – *sis* → *ficcisiis* 'make to finish'

c. *lix-'enter'* – *sis* → *liccisiis* 'make enter'

The last consonant of the stem in (3.18) and (3.19) is replaced by *–chi* or *–cci* based on the respective syllable.

(3.20) a. *dubb*-'talk'-*at-siis* → *dubbachiis* 'cause to talk'

b. *but* -'snatch'– *at* – *siis* → *buttachiis* 'to snatch'

c. *gaaf*-'request' *–at* – *siis* → *gaafachiis* 'make to request'

As it is illustrated in (3.20), the vowel(s) after the last consonant in the stems are replaced by the vowel *a* and the suffix *–siis* becomes *-chiis*.

C) Stative/Benefactive

The suffixes *-a*? and *–om* are added to an adjectival or nominal bases to derive a stative verbs in *Afaan Oromoo*. Temesgen (1995) gives the following examples.

TABLE 3.5: STATIVE VERBS

Base	Affix	Derived Statives
qaanii 'shame'	*-a*	*qaan-a* 'get ashamed'
gadhee 'bad'	*-a*	*gadh-aa* 'become bad'
fira 'relative'	*-om*	*fir-oom* 'become relative'
gamna 'wise'	*-om*	*gamn-oom* 'become wise'

As a result of assimilation of '*t*' to '*D*, the suffix *-at* is changed to *–aDD* in the first and second person imperative to derive stative from the base (Temesgen, 1995). Examples of such type of stative verb is *yaad-* 'think' + -at → *yaadaDD* 'remember'. It is however adapted to use *–addh* in place of *–aDD* primarily for ease of writing.

Kebede (1996), on the other hand, regard *–(a)aw* as the suffix that is added to the base of a noun and adjectives to derive stative verb stems. He illustrated using examples in (3.21) below.

(3.21) a. *beela-* 'hunger'*-aw* → *beelaw* 'be hungry'

b. *dukkana-*'darkness'*-aw* → *dukkanaaw* 'be dark'

c. *booruu-* 'muddy'*-aw* → *booraw* 'be muddy'

D) Passives

In *Afaan Oromoo*, adding the suffix *-am* to transitive stems reults in passive forms (Temesgen, 1995). Examples given below may illustrate such type of verbs.

TABLE 3.6: EXAMPLES OF PASSIVES

Stem	Suffix	Passives
abboom- 'command'	*-am*	*abboom-am* 'be obeyed'
nyaadhu- 'eat'	*-am*	*nyaata-am* 'be eaten'
mur-'cut'	*-am*	*mur-am* 'be cut'
gurgur- 'sell'	*-am*	*gurgur -am* 'be sold'

3.5.1.2.2 Derived Nouns

Gragg (1976) identifies a number of productive suffixes from which nouns are derived.

(i) *-eenna* and *(u)uma-* or *–(o)oma-* and *–eenya* are used to produce adjectival nouns.

(3.22) Examples: *jabaa* 'strong *jaab-eenya* 'strength'

nama 'man' *nam-ooma* 'humanity'

gamna 'wise' *gamn-ooma* 'wisdom'

(ii*) –oma(a)* produces as a demonstrative noun (abstract) when suffixed to the following nouns.

(3.23)

mootii 'rule *moot-oma(a)* 'governor ship'

michuu 'friend' *mich-ooma* 'friendship'

gaarii 'fine' *gaar-omaa* 'fineness'

(iii) *-aa, -tuu*: these suffixes are mostly used to distinguish between feminine and masculine. Consider the word *bar-* 'teach' and *duul-* 'campaign'. The resulting derived forms by

affixing the words are *bar-aa/-tuu* 'learner (m/f)' and *duul-aa/-tuu* 'campaigner (m/f)'. Nouns which are derived from such suffixes are called deverbative nouns (Gragg, 1976).

3.5.1.2.3 Derived Adjectives

Adjectives can be derived from different word categories (Heine, 1981). Examples for adjectives derived from verbs are given in (3.24) below.

(3.24)

a) adjectives derived from verbs

(i) *bareed-* 'be beautiful' + *tuu* → *bareeduu* (f) 'beautiful'

bareed- 'be beautiful' + *aa* → *bareedaa* (m) 'handsome'

(ii) *hat-* 'to steal' + *tuu* → *hattuu* 'thief'

(iii) *saam-* 'to rob' + *tuu* → *saamtuu* 'burglar'

Note that both *hattuu* and *saamtuu* have no gender distinctions.

3.5.1.3 Case and relational concepts

The case is handled broadly in *Afaan Oromoo* with a wide range of formal devices (Gragg, 1976). It can be shown that the base form of the noun (e.g. adjective) is used for direct object, to predicate nominal and isolated citation. Gragg (1976) discusses the following case suffixes.

Subject: The suffix n(i) is used to change nouns to subjects. As the following

examples illustrate, we do not necessarily change a noun that ends in long vowel, but the short vowel endings should be removed for the change to take place.

(3.25)

saree 'dog' → *sareen(i) 'the dog'*

garaa ''stomach' → *garaan(i)'the stomach*

harka 'hand' → *harki hark 'hand'*

The same argument above applies for definite and plural nouns. Examples in this case are:

(3.26)

nama 'man' → *nam-icha* → *nam-oo-ni*

nadheen-ittii 'you the (bad) woman' → *nadheen-ittii-n* 'that (bad) woman'

Possessive: The last vowel of a possessed noun is lengthened to express possession. In this case, the last vowel of the noun phrase is lengthened if it is short. Examples: *mana* 'house' *qoottuu* 'farmer' become *mana qottuu* 'a farmer house', and *mana namaa* 'a man's house'. The suffixes *-ti* and *-if* mark location and 'benefactive' respectively as used in *mana nam-aa (-ti)* 'it is somebody's house' and *mana nam-aa-ti-if* ' for somebody's house',

Dative: *-(dhaa)fi*, optionally appears after base forms. Examples of this sort are *keesuumaa* 'guest' *keesuumaa-dhaafi* 'for the guest' and *gurbaa* 'boy' *gurba-dhaaf* 'for the boy'.

Instrumental: *(dhaa)n: harka* 'hand' + *(dhaa)n* → *hark-aan* 'with hand'

From the suffix *–ni* words of this category can be formed. Example:

Afaani 'mouth' → *afaan-nini* 'with mouth'

3.5.2 Compounding

Compounding is the joining together of two linguistic forms, which functions independently (Wardhaugh, 1977). Typical examples of compound nouns include: *abbaa-buddenaa* 'step father' from *abbaa-* 'father' and *buddena* 'food' and *sanabata-guddaa* 'Sunday' from *sanabata-* 'Sabbath' *and guddaa* 'big'. Likewise, the compound *dafqee bulaa* 'proletariat' is formed from the verb *dafqee* 'toil' and the noun *bulaa* 'he who lives'. According to Bender et al. (1976), compound morphemes are rare in *Afaan Oromoo*. Nevertheless, *Afaan Oromoo* is very rich in compounds as well (for details see Temesgen, 1995 for example).

Although a large number of compounds occur in *Afaan Oromoo*, the underlying formation process is very irregular. It is therefore, a very difficult task for instance to determine the stem of compounds from which the words are made. As a result, it requires more efforts and time to develop a stemming algorithm that conflates such words for *Afaan Oromoo*.

3.6 Conclusion

It has been shown that for many cases the word formation process in *Afaan Oromoo* by affixation is regular. In *kus-uu* 'to store', *cims-uu* 'to make strong', *bar-uu* 'to learn'

CHAPTER 4

EXPERIMENTING THE STEMMING ALGORITHM

4.0 Introduction

In this chapter, the compilation of stoplist and suffixes, development of stemming algorithm for *Afaan Oromoo* and its evaluation is discussed. In addition to the base suffixes, the compiled list of affixes contains concatenated affixes. The stoplist is compiled from frequently occurring words of the sample texts. Prior to stemming, stop words are excluded to increase the degree of accuracy of conflation. The stemmer developed follows the longest match approach.

4.1 The Test Data

To study the behavior of *Afaan Oromoo* and to evaluate the degree of accuracy of the respective stemming algorithm, six sample texts were selected. The texts were obtained from the Government's Weekly Newsletter, *Barissa* in *Afaan Oromoo* language and from the Curriculum Department of *Oromyia* Regional State Education Bureau on textbooks of Junior and Secondary Schools. The major contents of the texts were: social studies, physical

education, language, handcraft and short articles.

The selection of the texts was primarily based on the availability of the document in electronic form. The contents of the sample texts selected for experimenting the development of the stemming algorithm were:

- News from one of *Barissa's* February 2000 edition (AON).
- A short fiction article obtained from *Oromyia* Education Bureau (FIC).
- The other four sample texts were obtained from Curriculum Department of *Oromyia* Education Bureau. These samples were textbooks and training materials in different disciplines for Elementary and High Schools of the Region. These textbooks were in:
  - ⇒ Physical Education for Grade 4 (SPOR)
  - ⇒ Art of making cloths (HR)
  - ⇒ Social Studies (contains history and geography) for Grade 8 (SOS) and,
  - ⇒ *Afaan Oromoo* Language for Grade 10 (AOL).

The size of the sample texts in terms of words is given in Table 4.1. These samples are from different disciplines and it is believed to represent the language. Although textbooks in Biology, Physics and Chemistry were obtained, they contain large number of loan words in their respective area and thus they are domain-dependent that do not serve for generic purpose. Therefore, they were not selected for the sample text.

4.2 The Word Distribution of Afaan Oromoo

The behavior of a language could be studied by analyzing the distribution of words in texts. Table 4.1 shows the size of the sample texts. The total running words and the corresponding distinct words are 106,863 and 22,227 respectively for the sample texts used in this study. The distribution of the words in the first 10% for each of the texts indicates that, few words constitute the major portions of the samples. As shown in Table 4.2, three of the samples (SOS, SPOR and HR) have six words each, AON has 7 words and the remaining two, AOL and FIC contain 16 and 13 words each respectively for the respective 10% of the occurrences. Furthermore, the first 30% of the occurrences in each of the samples are 36 for SOS, 46 for HR, 103 for FIC, 123 for AOL, 51 for AON, and 65 for SPOR.

TABLE 4.1: WORD RATIOS CALCULATED FOR SAMPLE TEXTS OF AFAAN OROMOO

Types of text	*Title of text*	*Size of text (total words)*	*Unique words*	*Ratio of Total words to Unique words*
News from *Barissaa* weekly *Afaan Oromoo* News letter	AON	1,555	856	1.817
Fictions	FIC	4,141	2,222	1.864
Hand Craft (How to make cultural cloths)	HR	15,661	3,755	4.171
Social studies, text book for grade 8	SOS	18,638	2,521	7.393
Afaan Oromoo Language, text book for grade 10	AOL	30,898	7,073	4.368
Cultural Sport, text book for Grade 4	SPOR	35,970	5,800	6.202
Total		106,863	22227	4.808

The percentage of the first 10 most frequently occurring words are also given for each

sample texts in Table 4.2. Except two of them (AOL and FIC), the samples have percentages above 10% for their respective 10 occurrences. The values computed by Alemayehu (1999) for Amharic sample texts for instance shows smaller than the one obtained for *Afaan Oromoo* samples. For the samples AGRI (3927 words) and NEWSP (22430 words), Alemayehu has obtained 7.8% and 5.9% respectively. The corresponding values of samples of similar sizes for *Afaan Oromoo* are 8.52 for FIC (4141) and 13.53 for HR (15, 661), respectively. In both cases the values of *Afaan Oromoo* are greater than that of Amharic, even with less running words in the latter case.

TABLE 4.2 : THE WORD DISTRIBUTION IN PERCENTAGE FOR FREQUENCY OF TWO OR ONE

Data	**No of words in the first 10% of the whole text**	**Percentage of occurrences of the first 10 top words**	**Frequency of two or one**	**Frequency of one**
SOS	6	15.04	11.6	9.86
SPOR	6	12.67	14.05	2.7
AON	7	13.24	56.02	29.1
AOL	16	8.31	23.17	15.43
FIC	13	8.52	44.1	29.5
HR	6	13.53	20.9	14.1

The frequency of occurrence of two or one for four of the samples are below 30, and only two of the samples (AON and FIC) have values greater than 30. These two samples with high percentages of occurrences are far smaller in sizes than the other samples. The values for the frequency of one also show higher for these two samples.

The frequency and rank table is also generated to see whether the distribution of

Afaan Oromoo words obey Zipf's law or not. Zipf's Law is stated (Rijsbergen, 1979: 13) as, "the product of the frequency of use of words and the rank order are approximately constant." The law is expressed as,

$$f*r = C$$

Where f is the frequency of occurrence of a word and r is the corresponding rank.

Alemayehu (1999) discusses that, many languages do not fully obey the law. His review of Zipf's work indicates that, the words in German and Norwegian exhibit a concave shape in the beginning for the log-log curve of the rank-frequency data. In another case, he has shown that, the law holds for Czech language only for those words with medium frequency.

Based on computed values for Amharic data, Alemayehu also argues that, Amharic words generally exhibit a high deviation from Zipf's law. The justification he gives for such deviation is the appearance of high proportion of singletons in Amharic texts. He also argues that, most of the low frequency words are the inflected or derived forms of words in the respective texts.

The Zipf's constants for some selected points of the sample texts in *Afaan Oromoo* are given in Appendix VII from a) through f). The product $f*r$ for five of the samples generally increases with an increase in rank and decrease in frequency with the exception of SOS, that shows some irregularity in this regard. Although, proper statistical testing is not done due to time constraint, this shows that, *Afaan Oromoo* also deviates from Zipf's law like Amharic.

A word to type ratio gives additional information about the behavior of a language in a text. It is the ratio of the total words to the distinct words of a text as shown in Table 4.1. There is a general indication from Hmeidi et al. (1997) that the ratio increases proportionally with the size of the tokens both for English and Arabic. The data obtained from this report however show some irregularity in this regard. From Table 4.1, the ratio for SOS is greater than that of AOL although it is less in size. In SOS, for instance, the first 28 occurrences constitute 27.69% of the entire text. The corresponding value for AOL constitutes about 15.1%, which is less by nearly 13%. In SOS, a word may be repeated several times than it does in AOL. However, since the remaining five sample texts follow the general trend of other languages (Eg., English and Arabic – see Hmeidi (1997)), this irregularity do not affect the validity of the data.

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. In other words, a particular word appears less often for Arabic than for English. The rationale given by Hmeidi et al., in this regard is that, the formation of verbs and derived nouns are person, number and gender dependent such that large number of words are formed by affixation in Arabic. Verbs in Arabic therefore appear in so many forms than it does in English. In addition, definite articles and some common prepositions in Arabic are attached to words with no space in between. This implies that affixing the existing words for Arabic forms large numbers of variants. In contrast, more new roots exist or occur in English than in Arabic. The analysis of the word ratio may be an indication for degree of complexity of morphology of a language. According to the values of the word ratio, the morphology of Arabic is far more complex than the morphology of English (Hemeidi et al., 1997; Alkharashi & Evans, 1996).

TABLE 4.2: COMPARISON OF WORD RATIOS OF AFAAN OROMOO WITH ENGLISH AND ARABIC

Language	*Text*	*Length of Text*	*Distinct Words*	*Word Ratio*
Afaan Oromoo	AON	1,555	856	1.816
	HR	15,661	3,755	4.171
English	Text 1	1,600	621	2.576
	Text 2	16,000	2,699	5.926
Arabic	Text 1	1,600	902	1.774
	Text 2	16,000	5,775	2.771

Note: In this table,

1. the data for English and Arabic is adapted from Hmeidi et al, 1997.
2. the sample texts AON and HR are compared with Text1 and Text2 respectively.
3. the contents of the texts are different except their similarities in size (that is, the Arabic, English and *Afaan Oromoo* texts are different).

The morphology of *Afaan Oromoo* can also be explained in terms of the reasons given for Arabic. Take for instance, the term 'rich', which is the same for both male and female in English, but different for *Afaan Oromoo* (*soressa* and *soretti*, respectively). Similarly, different forms are required when referring to persons. The equivalent terms for the term 'learner', for example is *baraa* for male and *bartuu* for female in *Afaan Oromoo*. This makes the number of distinct words higher in a text.

4.3 Stoplist

The list of stop words is compiled by merging the most frequently occurring words in six lists of the sample texts. Based on the results of the frequency of occurrences, the first 30 words from each text were compiled to stop list as shown in Table 4.3 below. Out of the total 180 words in Table 4.3, the unique words compiled to a stoplist are 106.

As shown in the list, few words constitute high percentage of their respective sample texts. The majority of such most frequently occurring words are function words (e.g., pronouns, definite articles, etc.,) and few of them are non-function words. Examples of the functional words included in this list are *fi* 'and', *kana* 'this', *gara* 'to', *irra* 'on' *garuu* 'but' and *yk* (short form of *yookiin*) 'or'. Nouns, pronouns, numerals, and other non-functional words, such as *barsiisu* 'to teach' are also found in this list. Unlike English, pronouns are inflected or derived in *Afaan Oromoo*. The pronoun *isa* 'he', for example exists in the sample text as *isaa* 'is he!', *isatu* 'he who' and *isaaf* 'for him'. As a result, a single word may appear in different variants in *Afaan Oromoo*.

TABLE 4.3: LIST OF THE MOST FREQUENTLY OCCURRING WORDS FROM SIX SAMPLE TEXTS

No	**SOS** (hawas)		**SPOR** (esport)		**AOL** (afaan)		**HR** (uffata)		**FIC**(qorsa)		**AON** (durdurii)	
	term	Freq	term	Freq	term	Freq	term	Freq	Term	Freq	term	Freq
1	fi	537	akka	1127	fi	597	fi	496	koo	73	gadaa	36
2	akka	287	fi	1019	hin	402	akka	244	kan	46	akka	26
3	barannoo	281	kan	530	kan	330	qal'aa	238	natti	38	kan	25
4	ni	262	gochuu.	331	akka	252	wayyaa	231	isaa	33	sadarkaa	24
5	barattoo	229	isaanii	323	tokko	184	ni	186	akka	31	sirna	18
6	mata	207	adda	310	keessatt	182	kan	153	kana	30	foollee	17
7	irratti	190	ni	278	ni	159	hin	152	ture	28	tokko	17
8	kan	190	dha.	250	itti	158	bBoffee	148	tokko	27	ni	15
9	barnoota	175	irraa	246	irratti	154	dha'iins	144	garuu	26	gadaan	14
10	duree	175	barattoo	240	isaa	151	ykn	128	isa	21	hin	14
11	dha	172	irratti	237	gilgaala	149	dha	126	malee	21	kun	14
12	fayyadam	172	itti	226	kana	144	barbaach	121	of	21	itti	12
13	barbaach	168	tokko	210	keessaa	142	irraa	115	ishee	18	ammoo	10
14	dhiyeess	150	addaa	209	barreess	135	adda	101	inni	17	foolleen	10
15	adda	125	gara	185	yoo	133	oomisham	92	keessa	17	kana	10
16	kana	123	keessatt	185	armaan	122	oomishuu	90	keessaa	17	keessatt	10
17	barsiisu	120	sochii	180	waan	119	fayyadam	89	marshaan	17	ykn	9
18	keessatt	115	aadaa	174	irraa	113	meeshaal	86	nama	17	ta'uusaa	8
19	kaartaa	108	hin	171	isaanii	107	dheedhii	82	jedhee	16	waggaa	8
20	barsiisa	97	kana	166	dha	103	fo'aa	81	har'a	15	ittiin	7
21	itti	93	wal	163	gara	102	meeshaal	79	waan	15	jiru	7
22	agarsiis	89	irra	161	wal	101	manii	78	harmee	14	kabajamu	7
23	kanaa	87	isaa	154	kun	99	qabu	78	marshaa	14	mirkanee	7
24	danda'am	80	ittiin	153	malee	94	qal'aa	76	naan	14	namoonni	7
25	kanneen	79	muuziqaa	145	maal	92	yemmuu	76	yeroo	14	shan	7
26	addaa	78	naannoo	144	nama	83	keessa	72	dhufee	13	yoo	7
27	lafaa	75	barattoo	142	barnoota	81	dha'iins	70	jedheen	13	yaa	7
28	gargaars	70	barbaach	120	irra	81	ittiin	70	jira	13	gara	6
29	isaanii	69	fayyadam	120	yeroo	81	tokko	70	kee	13	waa'ee	6
30	madaaluu	67	kanaan	120	booda	80	qajjisoo	67	abdiin	12	yaada	6

Further manual inspection of the texts was conducted to extract other non-content bearing words that are not included in the first 30 occurrences. In this way, another 182 additional words are extracted and the number of words in the list reaches 288. The complete list is given in appendix III.

occurring endings. From the sorted list of the reversed words, those suffixes with high frequencies were extracted (all the six sample texts were used). These suffixes were merged with suffixes obtained from literatures and then compiled. About 70 of these suffixes (endings) are base or lingustically valid suffixes (see Table 4.4).

TABLE 4.4: LIST OF AFAAN OROMOO SUFFIXES

a	een	mma	oolee	ttu
aa	eeny	n	oolii	tu
aadh	eess	na	oolii	tuu
aatii	eettii	ne	ooma	u
aawaa	ettii	ni	oota	umaa
aawaan	eyyii	nne	ota	umma
adh	f	nni	s	umsa
am	i	o	siis	umsi
at	icha	olee	sis	uu
ata	ii	olii	t	uud
aw	iitti	om	ta	uudh
aw	itti	oma	te	wwa
cha	ittii	oo	ticha	yyu
e	me	ooftuu	ttii	yyuu

The complete list of *Afaan Oromoo* endings compiled in this process is given in Appendix IV. The total number of endings compiled in this list is 342.

4.4.3 Prefix-Suffix Pairs

The occurrence of prefix-suffix pair is common in *Afaan Oromoo*. Terms such as *hinbeeku*, 'I do not know' and *hinbarbadu* ' do they want? contain both of the affixes. In these examples, *hin* is a prefix and *-u* and *uu* are suffixes added to the stem *beek-* and *barbad-* respectively. The removal of suffixes alone therefore does not give correct stems. It is thus necessary to include a procedure for stripping the prefix.

4.5 The *Afaan Oromoo* Stemmer

For the reasons discussed in chapter 2, it is not possible to apply the stemming algorithm developed for English or other languages to *Afaan Oromoo* due to the differences in the patterns of word formations and differences in their morphologies in general. Some of the approaches from these stemmers are however adapted to develop a stemmer for *Afaan Oromoo*.

In this regard, the rules applied by Ahmad et al. (1996) for Malay stemmer was adopted for *Afaan Oromoo*. The algorithm was based on longest-match approach. Two versions of the algorithm were developed. The first version is based on context-free suffix removal of words.

The rules for developing the algorithm were generated using the list of endings and prefix *given* in the preceding section. It is encoded as follows:

Suffix rules format: +Suffix e.g., + *tuu*

Other examples of the rules include: + *ota,* +*umsa,* + *aawaan,* +*yyuu*, +*e* and +*icha*.

The algorithm developed for the first version has the following general procedures.

FIG. 4.1: THE FIRST TRIAL OF AFAAN OROMOO STEMMING ALGORITHM

1. ***get*** the ***input*** word
2. ***if*** (word is stopword), ***goto*** 5
3. ***if*** (word is not stopword) and **if** (wordlength >= 3)

    if word ***endsWith*** any of the possible longest suffixes in the rules

        if (wordlength – suffixlength >= 3)

            remove suffix, ***goto*** 5

    else

        do the next step
4. ***return*** word,
5. ***stop, goto*** 1, if there is more word to be conflated

This algorithm performs the tasks in the following manner. First, it checks if a word is in the stoplist or not. If found in the list, the word is excluded from further processing and nothing returned to the calling routine; stop and process the next word if any. If the word is not in the stoplist, its length is checked in the next step. Any word whose length is less than or equal to three characters is returned as stem. A minimum stem length set for Afaan Oromoo is 3. Except few function words (e.g., ol 'up', of 'self'), no word is identified with two or less characters long. If a word is more than three characters long, control passses to the next step. In this step, the ending of the word is checked for any match in the rules for suffixes. If match is not found the word is returned as a stem. If a match is found, the suffix is removed from the word. The truncuated term is now taken as a stem.

For example *namoota* 'men' is 7 characters long, the term is also not included in the stoplist and it has an ending that matches with one in the stoplist *-oota*. The stemming operation in this case removes *oota* from the term and only the stem part *nam-* 'man' remains. Similarly, the term *hinbeeku* 'I do not know' is conflated to *hinbeek-* after the suffix *–u* is

removed. But, the conflated term is not the minimum stem expected since the prefix *hin-* is not removed. Words such as *xixina* 'the small ones', *gugudda* 'the big ones' and *kakkaas* 'pick them up' are also not reduced to their respective stems by this algorithm as discussed earlier in this section.

In this stemmer, procedures with the longest suffixes precede other consecutive shorter suffixes. Therefore, the longest possible suffix in the rules is removed before any shorter ones. The longest suffix compiled as rules in this case has 6 characters (e.g., *-aawaan, -eettii)* and the least is one character long. Examples of the outputs from the stemmer are given in Table 4.5.

TABLE 4.5: EXAMPLES OF CONFLATED TERMS BY THE FIRST VERSION OF AFAAN OROMOO STEMMER

Unstemmed term	expected stem	Result	error type
dhiiraa	dhiir	dhiir	
guntuta	guntut	guntu	overstemmed
dubraa	dubr	dubr	
tuttuquu	tuq	tuq	
jaallatan	jaallat	jaall	overstemmed
fayisaa	fay	fayis	understemmed
kootu	koot	koo	overstemmed
hime	him	him	
fayisaan	fay	fay	
hiriyaadha	hiriy	hiriy	
gandi	gandi	gandi	
keenyas	keeny	keeny	
walitti	wal	wal	
aana	aan	aan	
jabbii	jabb	jabb	
tiksinutti	tiks	tiksinu	understemmed
argincc	arg	arg	
walitti	wal	wal	
dhihaanne	dhih	dhih	
barree	bar	barr	understemmed
jaallanne	jaalat	jaall	overstemmed
ijoollee	ijooll	ijooll	
tiksituun	tiks	tiksit	understemmed
biratti	bir	bir	
hunda	hund	hund	
nu'argiti	arg	nu'arg	understemmed
Fayisaafi	fay	fayis	understemmed
Jedhanii	jedh	jedh	

4.6 Evaluation of the Algorithm

In this report, the error counting approach was adapted to evaluate the stemming algorithm that tastes the accuracy of the conflation results. The number of correctly conflated words and incorrectly confalted ones are counted for analysis. Accordingly, the selected terms were considered for the corresponding stemming equivalence. The output from the stemmer was then checked against the respective expected valid stem. These errors were then described in terms of understemming and overstemming. Understemming occurs when too much of the term is removed and understemming occurs when too little of the term removed. There are also some terms, which are not, stemmed. Examples are given in appendix V.

To evaluate the algorithm a test data of 1061 words of an article from fiction collection was selected. The words in the text contain base suffixes, prefixes, prefix-suffix pair, as well as concatenated suffixes. Out of the total words selected for stemming, 232 of them are stop words. As a result, the test was made on 829 of the words.

The output from the stemmer indicates 87 (10.5%) overstemmed and 145 (17.5%) understemmed. The words that are not conflated (with out any change) by the stemmer are 11. This constitutes about 1.2%. Examples of the sample words and the result after stemming are shown in Table 4.5. The accuracy of the stemmer is calculated to be 70.8%.

The evaluation is also made in terms of the degree of compression. The percentage of compression can be calculated using, $C = 100 \text{ X } (W - S)/W$, where W is the total word and S is a distinct stem after conflation. The value of C for the given sample text of *Afaan Oromoo* becomes $C = 100 \text{ X } (829 - 502)/829 = 39.44\ \%$.

It can be shown that, most of the incorrect results obtained from this test were due to the fact that some concatenated affixes were not compiled to the rules of the suffixes in this first trial. In the first version, the suffixes used were not more than 100. On the other hand, in the sample discussed, the word *barree* 'we knew/learn', for instance, was understemmed to *barr* since the algorithm operates on context-free base. If correctly stemmed *barree* becomes *bar* 'learn'. The stemming procedure in this regard require, a recording rule that removes one of the double *r* 's. Overstemming reported may be due to lack of recording of context – sensitive rules. The word *seenti* 'she entered', for instance, was understemmed to *see-*. Since there is no restriction on the removal procedure as far as the word length is three or more, the suffix *–n* was removed to give *–see*, according to the set of suffix rules.

Words, which have prefixes, are stemmed to an invalid term both linguistically and computationally. Examples of this sort include, *hinargitu* 'you will not find them' is conflated to *hinarg*; *hinbeektu* 'she do not know' to *hinbeekt* etc., are conflated to meaningless terms or it may result in understemming. The prefix *(hi)ni* was not removed in this version since the corresponding procedure was not included in the algorithm.

Terms, which are formed by reduplication of middle consonants and vowels, are also not stemmed accurately. The word *qaqqab* 'try to reach/catch', for instance, is returned with out any change by this algorithm. However, it should be conflated to 'qab' if stemmed correctly. Spelling errors are also causes for some of the non stemmed words. In order to correct such drawbacks, the algorithm was modified as discussed in the following section.

4.7 The Improved Stemmer

Having identified the drawbacks of the first version of the algorithm, a second version algorithm was developed to improve the stemming performance. The modifications made include:

- context-sensitive rules were introduced to reduce overstemming.
- prefix striping procedure was added to the algorithm
- a procedure was introduced to reduce some of the plurals formed by reduplication of the first syllable . The stemming of such words is not handled by the suffix removal procedures. As a result, separate procedure was developed. Examples of such terms include *kakkaas* 'pick them up' and *xixinoo* 'the small ones'. These words should be stemmed to *kaas-* 'pick ' and *xino* 'a small', respectively with appropriate algorithm.
- more number of suffixes used than the first version.

Some of the context sensetive rules included are:

1. remove one of the consonants if the conflated stem ends with doubled r, d,

   e.g., *bar-**r**ee* 'we knew/learn' → *bar-*

2. remove one of the consonants if the conflated stem ends with doubled l,q, and add "*at*"

   e.g., *jaa**ll**ach**u*** 'to love' → *jaal-* *l* →*jaal* → *jaal* + ***at*** → *jaalat*

3. change ***ch*** into ***at*** if occurs after short or long vowels

e.g., *huba**ch*** -is → *huba**at***

*nyaa**ch**is* 'feed him' → *nya**at***

5. change ***jett*** to ***jech***

   e.g., *jettaa* 'you say it'; *jette* 'you have said it'; *jetteen* 'she said it'

jettaa →*jett – tt* →*je + ch* →*jech*

6. if ***na, naa*** or ***nu*** (all indicates posseive conditions) appears at the beginning of a term

remove

e.g., ***nu****'argi* 'find/see for us' → *arg* 'find/see'

naa*kenn* 'give for me' → *kenn* 'give'

The Prefix Removal Algorithm

A procedure that removes the prefix *(hin)ni* was included to the modified algorithm. The corresponding procedure precedes the suffix stripping procedures such that prefix is removed prior to suffixes. In the word *hinilaalamne* 'it is not seen/observed', the stemmer first removes the prefix *hin* and then removes the suffix. The algorithm for the prefix removal is given in Fig. 4.2 Below.

FIG 4.2: PREFIX REMOVAL ALGORITHM

1. ***get input*** word
2. ***if*** (word ***startsWith*** *ni) &&*
   if (substringlength without *ni* >=3***) return substrings*** after *ni*, and **goto** 3.
   else if (word ***startsWith*** *hin) &&*,
   if (substringlength without *hin* >=3)
   return substring after *hin, and* ***goto*** *3*
   else
   return word
3. ***goto*** 1 if there are more words to be stemmed.

In this algorithm, first, the input term is entered. In the next step, the term is checked if it starts with the prefixes *ni* or *hin*. If satisfied, the algorithm checks for length of the word.

If the term becomes three or more characters long after the removal of the prefix, the next sub step will be excuted, i.e., remove the prefix and stop. Then it will process stemming of other terms if there is any. If, on the other hand, the input word is less than three characters long, the word is returned as stem. Then, further processing of stemming will take place if there is more words.

Suffix Removal Algorithm

For the improved stemmer, the algorithm shown in Fig. 4.3 is developed for suffix stripping procedure.

FIG.4.3: ALGORTHM FOR SUFFIX REMOVAL

1. ***get input*** word
2. open suffix list file

// compare the word ending with the suffix from the suffix list

3. if (word ***endsWith*** suffix)

if ((wordlength – suffixlength) >= 3)

goto 4

else

return word

4. ***if (***suffix has specific conditions or recording rules)

apply the respective rules, ***goto*** 5

else

return word

5. ***return*** the remaining part of the word as stem
6. ***stop, if*** there are more words, ***goto*** 1.

Using the sample text tested for the first version of the stemmer, the results is given in Appendix V. The number of overstemmed and understemmed words were 38 (4.58%) and 20 (2.4%) respectively. An additional of 4 words was recorded as not stemmed. This roughly constitutes 0.5% of the total words. The accuracy rate of conflation of the stemmer is 92.52%. The modified stemmer, therefore, improves the performance by about 22%.

The compression, C for this stemmer is, C = (829 – 472)/829*100 = 43.06%. The compression is also improved.

To illustarte the result of compression from the stemmer, an example is given in Appendix VI. The paragraph in this example is taken from a reading comprehension in AOL. Out of the total 93 words, 26 of them are stop words. The resulting compression due to the removal of the stop words alone reduces the sample by about 30% (cf. With 30 – 50 % of Rijinsberg's argument in Section 2.2.6) of the original size. A quick glance at the results of the stemmer in Appendix VI reveals a substantial size reduction of *Afaan Oromoo* text due to stemming. Part (b) of the figuer indicates the result of the stemmer prior to eliminating stop words from the input text. A substantial size reduction is observed in part (c) is the result of the stemmer after elimination of stop words. Using the results in part (c), the sample is now reduced to (the unique word is 52),

C = (93 – 52) /93 * 100 = 43%, (Note: incorrectly conflated terms are also considered)

The result indicate that, *afaan Oromoo* texts can be significantly reduced in size due to stemming operation. The result is comparable to the values of similar tests in other

stem. First, the prefix *hin/ni* is removed, leaving *xixiinata*, *kakkastuu* and *deemna.* In the next step, the procedure that conflates words formed by duplication of first syllable is applied and removes *xi* and *kak* from the first and second terms respectively. The resulting conflated terms in this procedure are then checked against the suffixes in the list. Accordingly, *ata, tuu* and *na* are removed from the terms respectively. The last two terms are the required correct stems, while the first term still requires further conditional rules (not handled by this algorithm) to be conflated to a correct stem. The final outputs from this algorithm are *xiin*, *kas* and *deem.*

The errors occurred might be due to various reasons. The major ones are:

- matches not found in the suffix dictionary for some endings, thus the requirement of more number of endings;
- a number of contextual problems require individual treatments (e.g., the term *barbaacha* 'wanting' is conflated to *barb* instead of *barbaad*);
- although the modification results in improving the performance of the stemmer, the addition of some routines, such as the one for *na* and *nan* introduced other errors. The term *nadheen* 'woman' and *naquu* ' make in order', for instance are incorrectly conflated to *dheen* and *quu,* respectively, which are not stems. There are a number of similar cases.
- spelling errors of terms, and
- the size of the data used for the evaluation is not adequate.
- Terms with hudha 'glottal stop', " ' " are not conflated correctly. The term bu'a 'benefit', for example conflated to bu', instead of bu'a.

The stemming algorithm developed in this study follows the longest-match approach.

In using the longest match, we may require few numbers of context-sensitive and recording rules than iterative algorithm. In *beeknelle* 'even if we knew' and *beekaniruu* 'they have already knew', the suffixes are *nelle* and *niruu*, respectively. In addition to base suffixes these endings contain consonants like 'l' and 'r', which are included to it during the word formation process. If itrative algorithm is used to conflate such terms, they are understemmed, unless additional rule is formulated. There is no need to worry about such irregularities in using the longest match suffix removal algorithm as long as the ending is in the suffix list or dictionary. There are still more recordings and suffix rules to be formulated to increase the accuracy of the stemmer.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

To develop a stemming algorithm, the knowledge of the respective morphology is necessarily required. Accordingly, the morphology of *Afaan Oromoo* was reviewed to develop a stemmer. Words in *Afaan Oromoo* are formed largely by affixation process. Affixation involves either suffixing or prefixing. The only prefix used in the language is *(hi)ni*. Apart from affixation and compounding some words are formed by reduplication of middle consonants and vowels as in *kaas* 'pick' and *kakkaas* 'pick them up'.

The analysis of word ratios of total words to distinct words calculated from sample texts shows that, *Afaan Oromoo* is morphological complex language than English and Arabic, but less complex than Amharic. Likewise, the possiblity of forming significant number of words from a single root in *Afaan Oromoo* is another indication for the complexity of the language's morphology (see Appendix II).

Semi-automatic procedure was used to compile suffix list. About seventy of the suffixes are base suffixes. Function and the most frequently occuring words from the sample

texts are compiled to stoplist. In Afaan Oromoo, most of these function words may appear in different forms as they can be inflected or derived (e.g., *isa* 'he', *isaaf* 'for him', *isaan* 'they', *isaaniif* 'for them', etc.,). As a result, a high proportion of words in a document is excluded from further processing of stemming. Hence, the compression value is high (about 40%). This result shows the possibilty of employing a stemming algorithm for conflating *Afaan Oromoo* words.

The stemming algorithm developed follows longest-match truncation of suffixes from roots. The degree of accuracy of the stemmer in stemming words is evaluated. A sample of 1061 words was used to test the algorithm. The *Afaan Oromoo* stemmer developed for this study is accurate to a rate of 92.52%. Error due to overstemming and understemming were about 4.58 % and 2.4% respectively, after the stemmer was modified. Like any other stemmers for other languages, this stemmer can be tested and used for IR purpose. The outcome is thus a promising for further research and applications. Morphological complexity is the major causes for the errors.

5.2 RECOMMENDATIONS

As various relevant studies shows, the performance of retrieval is enhanced with stemming than with non-stemming operations, specially for morphologically complex languages, such as Slovene (Popovic and Willett, 1992). Therefore, the stemming algorithm such as the one developed from this study would be used in IR system for retrieval of text in *Afaan Oromoo*. Considering the amount of work done and its limitation the following recommendations

are made for further research:

- The algorithm developed does not conflate words with irregular patterns of formation/affixation in general, as it needs more detailed study of the morphology to develop more effective algorithm. Incorporating procedures that conflate such words could make the stemmer much more efficient.
- Since the data utilized for this study is not and can not be exhaustive, further study will be require to proof the result using diversified and more larger size data.
- Further study can be done to use the result from this research for IR or other applications.
- Further study can be conducted on the word distribution of the language and thus the respective quantitative analysis to give more detailed elaboration of the language's behavior.
- A number of irregularities (e.g., the prefix *(hi)ni* is assimilated to a root or written as a separate word or found mixed even in some texts) have been observed on the written system of *Afaan Oromoo* as reviwed the existing literatures. This is also another unexplored and important area of research. Therefore, standard has to be set.
- Futhermore, this work can be used to develop application tools, such parsers, spell checkers, and dictionaries for *Afaan Oromoo*.

This stemmer is probably the first of its kind for the language. It is thus believed to be a starting material for those interested to extend the stemming operations as a tool to the IR environment for *Afaan Oromoo*.

BIBILIOGRAPHY

Ahmad F., Yosuf M. and Sembok T. M. T. (1996). Experiments with Stemming Algorithm for Malay Words. *Journal of the American Society for Information System*, 47(12): 909 – 918.

Alemayehu, N. (1999). *Development of a Stemming Algorithm for Amharic Language Text Retrieval.* P.h.D Thesis, University of Sheffield (Unpublished).

Al-Kharashi, I. A. & Evens, M. W. (1994). Terms in an Arabic Information Retrieval System. *Journal of the American Society for Information System*, 45(8): 548 – 560.

Bender, M. L., Bowen, J. D., Cooper, R. L. and Ferguson, C. L. (1976). Two Cushitic languages. In M.L Bender et al. (Eds), *Language in Ethiopia*, (pp. 135- 148), London, Oxford University Press.

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

Croft, B. (1999). Stemming at *URL: http://ciir.cs.umass.edu/cmpsci646/ir4/tsld017.htm*

Dawson J. L., (1974). Suffix Removal and Word Conflation. *Bulletin of the Association for Literacy and Linguistic Computing,* 2, 33-46.

Dietel M. H. and Dietel J. P. (1997). *Java: How to Program*. Upper Saddle River: Prentice Hall.

Ekmekcioglu C. F., Lynch M. F. and Willett P. (1996). Stemming and N-gram Matching for Term Conflation in Turkish Texts at URL: *http://panizzi.shef.ac.uk/tom/information_research/paper13.html*. Visited on 2/17/2000.

Flynn R. R. (1987). *An Introduction to Information Science*. New York: Marcel Dekker Inc.

Frakes, W. B. (1992). Stemming Algorithms. In W.B. Frakes & R. Baeza Yates (Eds), *Information Retrieval, Data Structures & Algorithms* (pp. 231-160), Englewood Cliff. NJ: Printice-Hall.

Frants V. I., Shapiro J. and Voiskunskii V. G. (1997). *Automated Information Retrieval: Theory and Methods*. San Diego: Academic Press.

Grag G. (1976). Oromo of Wellegga. In M. L. Beder (Eds.), *The Non-Semitic Languages of Ethiopia,* (pp. 166- 195). Michigan: African Studies Center, Michigan State University & Illinois University.

Hammid M. M. (1995). *Haamid Muudee's Oromo Dictionary Volume I, English - Oromo*: Atlanta: Sagalee Oromoo Pub. Co., Inc.

Harman D. (1991). *Journal of the American Society for Information System*, 42(1): 7 -15.

Heine B. (1981). *The Waata Dialect of Oromo: Grammatical Sketch and Vocabulary.* Berline: Dietrich Reimer Verlag.

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.

Hull D. A. (1996). Stemming Algorithms: A Case Study for Detailed Evaluation. *Journal of the American Society for Information System*, 4(1): 70 – 84.

Kebede H. (1996). *A Comparative Study Oromo Dialects: Aspects of Assimilation. Academy of Ethiopian Languages.* Ministry of Information & Culture: Addis Ababa. (Unpublished).

Flynn R. R. (1987). *An Introduction to Information Science*. New York: Marcel Dekker Inc.

Frakes, W. B. (1992). Stemming Algorithms. In W.B. Frakes & R. Baeza Yates (Eds), *Information Retrieval, Data Structures & Algorithms* (pp. 231-160), Englewood Cliff. NJ: Printice-Hall.

Frants V. I., Shapiro J. and Voiskunskii V. G. (1997*). Automated Information Retrieval: Theory and Methods*. San Diego: Academic Press.

Grag G. (1976). Oromo of Wellegga. In M. L. Beder (Eds.), *The Non-Semitic Languages of Ethiopia,* (pp. 166- 195). Michigan: African Studies Center, Michigan State University & Illinois University.

Hammid M. M. (1995). *Haamid Muudee's Oromo Dictionary Volume I, English - Oromo*: Atlanta: Sagalee Oromoo Pub. Co., Inc.

Harman D. (1991). *Journal of the American Society for Information System*, 42(1): 7 -15.

Heine B. (1981). *The Waata Dialect of Oromo: Grammatical Sketch and Vocabulary.* Berline: Dietrich Reimer Verlag.

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.

Hull D. A. (1996). Stemming Algorithms: A Case Study for Detailed Evaluation. *Journal of the American Society for Information System*, 4(1): 70 – 84.

Kebede H. (1996). *A Comparative Study Oromo Dialects: Aspects of Assimilation. Academy of Ethiopian Languages.* Ministry of Information & Culture: Addis Ababa. (Unpublished).

Lewis D. D. and Sparck Jones K. (1996). Natural Language Processing for Information Retrieval. *Communication of the ACM*, 39 (1): 92 – 100.

Lewis J. and Loftus W. (1998). *Java Software Solutions: Foundations of Program Design.* Reading: Addison-Wesely.

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

Melba, G. (1988). *The Oromoo People and Oromia.* Khartoum, Sudan.

MOE (1994). *Education and Training Policy.* April 1994, Addis Ababa.

Naughton P. and Schildt (1999). *Java 2: The Complete Reference,* Third Edition. New Delhi: Tata McGraw-Hill.

Paice D. C. (1996). Methods of Evaluation of Stemming Algorithms Based on Error Counting. *Journal of the American Society for Information System*, 47(8): 632 – 649.

Payne, T. (1999). The Structure of Languages, at URL: *http://darkwing.uoregon.edu/~tpayne/Phonotics.htm.*

Perirotti T.(1997). How Search Engine Works. Microsoft Corporation at URL: *http://www.tgpconsulting.com/articles/mund.html* (visited on 10 April, 2000).

Popovic M. and Willett P. (1992). The Effectiveness of Stemming for Natural-Language Access to Slovene Textual Data. *Journal of the American Society for Information System*, 43(5): 384 – 390.

Porter M. F. (1997). *An Algorithm for Suffix Stripping.* In Spark Jones K. and Willet P. (Eds.), Readings in Information Retrieval, pp. 312 – 316. San Francisco: Morgan Kaufmann Pubs., Inc.

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

Salton G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Massachusetts: Addison-Wesley Pub. Company.

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

Savoy, J. (1999). A Stemming Procedure and Stopword List for General French Corpora, *Journal of the American Society for Information Science* 50(10): 944 - 952, 1999.

Schiffman, H. F. (1999). Content in Language, at URL: *http://ccat.sas.upenn.edu/~haroldfs/popcult/handouts/morphology/nodel.*

Schinke R., Greengrass M., Robertson A. M. and Willett P. (1996). A Stemming Algorithm for Latin Text Databases. *Journal of Documentation,* 52 (2), pp. 172 –187.

Silzer, P. (1999). Morphology at URL: *http://www.people.biola.edu/faculty/peter/lingustics/morphology.htm.*

Sparck Jones K. (1981). *Information Retrieval Experiment*. London: Butterworths.

Strazalkowski T. (199). Natural Language Information Retrieval. *Information Processing and Management,* 31(3), p. 397-417.

Stroomer H. (1987). *A Comparative Study of Three Southern Oromo Dialects in Kenya: Phonology, Morphology and Vocabulary*. Hamburg: Buske.

Temesgen N. (1995). Word Formation in Oromo. *Ethiopian Journal Languages and Litrature,* No. 5, Institute of Language Studies, Addis Ababa.

Tilahun G. (1994). Afaan Oromoo, Journal of Oromoo Studies. URL: *http://www.sas.upenn.edu/African.studies/Hornet/Afaan_Oromoo_19777.html*.

Tizana P. (2000). Under the Covers: How Search Engine Work at URL: *http://www.tpgconsulting.com/articles/mind.html*. Visited on 4/8/00.

Wardhaugh, R. (1977). *Introduction to Linguistics.* New York: McGraw-Hill Book company.

APPENDCES

APPENDIX I*
A) PRONUNCIATION OF AFAAN OROMOO VOWELS

Afaan Oromoo Vowels	*Approximately corresponds to English*	*Examples*
a	a	as in but
aa	a	as in father
e	e	as in bet
ee		as in late
i	i	as in sit
ii	i	as in seat
o	o	as in no
oo	o	as pope
u	u	as in put
uu	u	as in pool

* Taken from Hammid Mudee's English to Oromoo Dictionary (Hammid, 1995).

... Appendix I

(B) PRONUNCIATION OF AFAAN OROMOO CONSONANTS

Afaan Oromoo consonants	*approximately corresponds to english*	*examples*
b	b	as in book
c	no approx. equivalent	
ch	ch	as in church
d	d	as in door
dh	no approx. equivalent	
f	f	as in fun
g	g	as in good
h	h	as in hat
j	j	as in jump
k	k	as in king
l	l	as in lamp
m	m	as in man
n	n	as in north
ny	no approx. equivalent	
p	p	as in public
ph	no approx. equivalent	
q	no approx. equivalent	
r	r	as in root
s	s	as in sand
sh	sh	as in sharp
t	t	as in table
v	v	as in victory
w	w	as in west
x	No approx. equivalen	
y	y	as in yonder
Z	z	as in zebra

APPENDIX II
EXAMPLE OF A STEM AND ITS DIFFERENT VARIANTS

Note that the words in this list are all formed from the stem *BEEK* ' to know'. And all of the words have meanings relevant to 'know'.

bebeekan
bebeeke
beeka
beekaa
beekaammoodha
beekaani
beekadha
beekama
beekamaa
beekaman
beekame
beekamee
beekadhee
beekamin
beekamoo
beekamte
beekamto
beekamtu
beekamu
beekamut
beekamuu
beekan
beekanii
beekaniruu
beekaniruuf
beekee
beekiin
beekkadh
beekkama
beekkame
beekkami
beekkamo
beekkamt
beekkamu
beekkan
beekkums
beekkuum
beekna
beekne
beekneen
beeknelle
beeknerr
beeko
beekoo
beeksiis
beeksisa
beeksise
beeksisu
beekta
beektaa
beektelle
beekti
beektoot
beektota
beektu
beektuu
beektuuf
beeku
beekumsa
beekumsi
beekus
beekuu
beekuudh
beekuuf
beekuufi
beekuuka
beekuule
beekuun
hinbeekaa
hinbeekaani
hinbeekama
hinbeekamaan
hinbeekame
hinbeekamee
hinbeekamin
hinbeekamtu
hinbeekamu
hinbeekamuu
hinbeekamuu
hinbeekanii
hinbeekee
hinbeekkama
hinbeekkamu
hinbeeksisaan
hinbeeksise
hinbeekta
hinbeektu
hinbeektuu
hinbeeku
nibeekkame
nibeekne

APPENDIX III
STOP WORDS COMPILED FROM AFAAN OROMOO SAMPLE TEXTS

aadaa
abbaasaa
abbootii
abdii
abdiin
abdiinis
ada'aa
adabbii
adda
addaa
addaan
afaan
afoola
afurtama
akka
akkan
akkas
akkasuma
akkuma
amma
ammayyaa
ammoo
ana
anaaf
anaafi
ani
argaman
armaan
arraabaa
as
asheeta
asirratt
ati
baatii
bakka
bakkan
bakkee
fi
fuula
fuuldura
gabaabdu
gad
gadaa
gadi
gahe
galaana
gama
ganda
gara
garaa
garaagar
garee
garuu
gatii
gidduu
gidiraa
gilgaala
guddaa
gumaacha
haa
haala
haalli
hanga
har'a
harka
harmee
harmeen
hawaasic
hawaasum
hayyuule
heddu
hedduu
hedduun
hidhuu
kaarrota
kafana
kam
kamirray
kan
kana
kanaa
kanaaf
kanaafuu
kanaan
kanan
kanarrat
kee
keenya
keenyaa
keessa
keessaa
keessatt
kiilolee
kiyya
kiyyaa
kiyyaan
kkf
koo
kootiin
kootu
kudhan
kun
kunneen
kurnan
kutaa
lafa
lakkoofs
lama
lamaan
leenci
loosuu
roobaa
roobni
sababa
sadarkaa
saddeet
saddeeti
safuu
san
sana
sanatti
sani
sanii
shan
shanan
shani
shawaa
si
sigabaa
sigabaar
sirboonn
sirbu
siree
sirna
sirni
sirnoonn
siyaasaa
soddoma
sun
suufii
taa'ee
ta'an
ta'e
ta'ee
ta'eefis
tartiiba
ta'uu
ta'uusaa

APPENDIX IV
LIST OF AFAAN OROMOO WORD ENDINGS (SUFFIXES)

aawaan
achiis
amanii
eettii
aachi
aachu
aaiif
aannu
aatan
aatte
aatti
aatto
aawaa
achaa
achii
achis
achuf
adhaa
adhee
aadhe
adhuu
aattu
amaan
amarr
ameen
amett
amsis
amtan
amuuf
amutt
aniif
aniin
aniir
anitt
annaa
annoo
annuu
ataan
isis
isne
isnu
isuu
itee
itti
naan
neen
nerr
ofte
olee
olii
oole
ooma
oofi
oota
ootn
siis
sisa
sisu
teem
teet
tiif
tota
toon
toot
ttii
tuuf
umaa
umma
umsa
umsi
umtu
unis
urra
utti
uuda
uudh
aatu
amaa
amee
amin
ataa
atto
acha
achi
achu
adha
adhe
adhu
amaa
aman
amee
ameh
amne
amni
amoo
amta
amte
amti
amto
amtu
amus
amut
amuu
anii
anin
anir
anis
anne
anni
annu
anuu
asin
ataa
atam
iin
iis
iit
ihu
ile
ina
inu
isa
isi
isu
ite
iti
itt
itu
iun
i'uu
lle
mma
mmo
msa
msi
nne
nni
ole
oon
oot
ota
sii
taa
tan
tee
tii
tte
tti
ttu
tun
tus
tuu
inee
inis
inni
inus
irra
isan
isee
iseh
isii
a
e
f
i
n
o
s
t
u

... List of *Afaan Oromoo* word endings (suffixes)

aann	ess	iitt	tu
aata	ete	iiww	uf
aate	etu	ilee	un
aati	fii	immo	us
aatt	iif	inaa	uu

APPENDIX V
EXAMPLES OF WORDS AND THE RESULTING STEM BY THE STEMMING ALGORITHM.

unstemmed term	Expected stem	result	error type
dhiiraa	dhiir	dhiir	
guntuta	guntut	guntu	overstemmed
dubraa	dubr	dubr	
tuttuquu	tuq	tuq	
jaallatan	jaallat	jaall	
fayisaa	fay	fayis	understemmed
kootu	koot	koo	overstemmed
hime	him	him	
fayisaan	fay	fay	
hiriyaadha	hiriy	hiriy	
gandi	gandi	gandi	
keenyas	keeny	keeny	
walitti	wal	wal	
aana	aan	aan	
jabbii	jabb	jabb	
tiksinutti	tiks	tiks	
arginee	arg	arg	
walitti	wal	wal	
dhihaanne	dhih	dhih	
barree	bar	barr	
jaallanne	jaalat	jaalat	
ijoollee	ijooll	ijooll	
tiksituun	tiks	tiksit	understemmed
biratti	bir	bir	
hunda	hund	hund	
nu'argiti	arg	arg	
fayisaafi	fay	fay	
jedhanii	jedh	jedh	
qabu	qab	qab	
ukoo	ukoo	ukoo	
dhoksinee	dhoks	dhoks	
fidna	fid	fid	
fayisaan	fay	fay	
baasee	baas	baas	

… Examples of words and the resulting stem by the stemming algorithm

naakenna	kenn	kenn	
ukoo	ukoo	ukoo	
seenee	seen	seen	
fudhata	fudh	fudh	
guntuta	guntut	guntu	overstemmed
tuttuqaa	tuq	tuq	
mi'aa	mi'a	mi'	understemmed
akaayii	akaay	ayii	overstemmed
hima	him	him	
harka	hark	hark	
duwwaa	duww	duww	
arginus	arg	arg	
qubni	qub	qub	
ukoo	ukoo	ukoo	
seenee	seen	seen	
sakatta'a	sakatt	sakatta'	understemmed
barbaacha	barbaach	barbaad	
akaayii	akaay	ayii	overstemmed
fakkeessee	fakk	fakk	
nahooksa	hook	hook	
rakadhen	rakkoo	rakadhen	not stemmed
waadaa	waad	waad	
seena	seen	see	overstemmed
gooftaa	gooft	gooft	
borin	bor	bor	
fida	fid	fid	
jedheen	jedh	jedh	
kakadha	kadh	kadh	
guntuta	guntut	guntu	overstemmed
dabree	dabr	dabr	
naseena	seen	seen	
harka	hark	hark	
qabee	qab	qab	
naqirqida	qirqid	qirqid	
kofalchiiseen	kofl	kofalchi	understemmed
aboochisu	aboochis	aboochis	
malee	mal	mal	

... Examples of words and the resulting stem by the stemming algorithm

gadi	gad	gad	
nadhiisu	dhiis	dhiis	
oolee	ool	ool	
bulee	bul	bul	
qubni	qub	qub	
fayisaa	fay	fayis	understemmed
araada	araad	araad	
utuun	utu	utu	
hinargin	arg	arg	
aduun	adu	adu	
dhiite	dhii	dhii	
nanyaachisa	nyaat	nyaat	
uumeen	uum	uum	
quba	qub	qub	
barbaacha	barbaad	barb	overstemmed
jabbii	jabb	jabb	
ilaaleen	ilaal	ilaal	
deebi'a	deeb	deeb	
horiin	hor	hor	
midhaan	midh	midh	
seente	seen	seen	
jedhu	jedh	jedh	
eeleen	eel	eel	
barbaada	barbaad	barbaad	
jedheen	jedh	jedh	
dhugaatti	dhug	dhug	
siqu	siq	siq	
harmeetti	harm	harm	
himeen	him	him	
fiiga	fiig	fiig	
fayisaa	fay	fayis	understemmed
gahee	gah	gah	
hooksamuuf	hook	hooks	understemmed
fayisaan	fay	fay	
xaaxeessaa		xaaxeess	

... Examples of words and the resulting stem by the stemming algorithm

fiiga	fiig	fiig
niaraada	araad	araad
sagalee	sagaal	sagal
qabee	qab	qab
sirba	sirb	sirb
waddeessa	wadd	wadd
maasii	maas	maas
nabarbaada	Barbaad	barbaad

APPENDIX VI
COMPARISON OF UNSTEMMED AND STEMMED TEXTS

a) Unstemmed

Bosonniifi bineensonni qabeenya uumamaa ti. Garuu, namoonni baay'een bu'aa bosonaafi bineensota sirriitti hin hubatan; akka qabeenyaattiis hin ilaalani. Fakkeenyaaf, akka aadaa fi muuxannoo hawaasa baadiyyaatti, bal'ina lafa qonnaa guddina ooyruufi baay'ina lafa horataman qofa. Bosonni akka tuuta mukkeen baay'inaan margee qofatti ilaalama. Warra baadiyyaa biratti, bu'aa bineensonni qaban caalaa balleessaa isaaniitu hubatama midhaan balleessuu fi beeyladaa miidhuu isaanii. Bineensa ajjeesuunis akka jagnummaa fi gootummaatti ilaalama. Kanaafuu, qabeenya bosonaa fi bineensotaatti akka malee fayyadamuun bal'inaan mul'ata. Haa tahuu malee, qabeenyaa fi faaya uumamaa tahuun gamatti, qabeenya biyyaa tahuun isaanii fi faayidaaleen isaanii hubatamuu qaba.

b) After stemming (including stop words)

Bosonn bineenson qab uum ti Gar nam baay' bu' boson bineenso sirri hin hubatan; akk qabeenyaatti hin ilaal Fakkeeny akk aad fi muux hawa baadiyy bal'I laf qonn guddi ooyr baay'I laf hor qof Boson akk tuu mukk baay'I marg qof ilaal Warr baadiyy bir bu' bineenson qab caal balleess isaanii hub midh balleess fi beeylad miidh is Bineen ajjeesuun akk jagn fi gootumm ilaal Kanaaf qab boson fi bineensot akk mal fayyad bal'I mul' Haa tah mal qabeeny fi faay uum tah gam qab biyy tah is fi faayidaal is hub qab

c) Stemmed (stop words excluded)

Bosonn Bineenso qab uum nam baay' bu'a boson bineens sirri qabeenyaatt ilaal aad muux hawaa baadiyy bal' qonn gudd ooyr baay' horat qof boso tuu mukk baay'i marg qof ilaal warr baadiyy bir bu' bineenso qab caal balleess isaani hubat midh balleess beeylad miidh bineen ajjeesu jagn gootumm ilaal kanaaf qab boson bineensot mal fayyadam bal'I mul'a mal qabeeny faay uum gam qab biyy faayidaal hubat qab.

APPENDIX VII

FREQUENCY OF WORDS AND ZIPF'S CONSTANT FOR SELECTED RANKS

a) AOL

Rank	Word	*f*	*f*r*
1	fi	597	597
5	tokko	184	920
10	isaa	151	1510
20	irraa	113	2260
30	maal	92	2760
40	isa	74	2960
50	ture	58	2900
60	fayyadam	53	3180
70	wajjin	47	3290
80	aadaa	42	3360
90	Kana	40	3600
100	Jechoota	37	3700
200	BOQONNAA	22	4400
300	rakkoo	16	4800
400	duwwaa	12	4800
500	Aseennaa	10	5000
600	lameenii	9	5400
700	kan	7	4900
1000	dhalate	5	5000
2000	mul'achu	3	6000
3000	maxxansi	2	6000
4000	bal'isuu	1	4000
5000	ergamu	1	5000
8153	Zo'ooloo	1	8153

b) SOS

Rank	Word	f	f*r
1	fi	537	537
5	mata	207	1035
10	dha	172	1720
20	kaartaa	108	2160
30	isaanii	69	2070
40	baruu	58	2320
50	Kanaafuu	53	2650
60	Wayitii	30	1800
70	dhiheess	15	1050
80	barnooti	10	800
90	ol	8	720
100	kennamu	6	600
200	qo'atani	5	1000
300	KUTAA	4	1200
400	ga'ee	3	1200
500	ofirraa	3	1500
1000	Baratoon	2	2000
2000	jjiran	1	2000
2477	Yookiin	1	2477

c) FIC

Rank	Word	f	f*r
1	koo	73	73
5	akka	31	155
10	isa	21	210
20	har'a	15	300
30	Abdiin	12	360
40	ta'ee	11	440
50	akkan	9	450
60	sun	9	540
70	jiru	8	560
80	waa'ee	7	560
90	jechuun	6	540
100	bira	5	500
200	daa'imum	3	600
300	akkasii	2	600
400	fudheen	2	800
500	mara	2	1000
600	Akkam	1	600
1000	dhufu	1	1000
2000	siigale	1	2000
2222	Yuuniver	1	2222

c) FIC

Rank	Word	f	f*r
1	koo	73	73
5	akka	31	155
10	isa	21	210
20	har'a	15	300
30	Abdiin	12	360
40	ta'ee	11	440
50	akkan	9	450
60	sun	9	540
70	jiru	8	560
80	waa'ee	7	560
90	jechuun	6	540
100	bira	5	500
200	daa'imum	3	600
300	akkasii	2	600
400	fudheen	2	800
500	mara	2	1000
600	Akkam	1	600
1000	dhufu	1	1000
2000	siigale	1	2000
2222	Yuuniver	1	2222

d) HR

Rank	Word	f	f*r
1	fi	496	496
5	ni	186	930
10	ykn	128	1280
20	fo'aa	81	1620
30	qajjisoo	67	2010
40	mi'a	57	2280
50	bakkee	44	2200
60	keessatt	37	2220
70	darbattu	30	2100
80	bifa	28	2240
90	Akka	25	2250
100	Darbattu	22	2200
200	mul'atu	13	2600
300	maruu	9	2700
400	sirrii	7	2800
500	cuuphamu	5	2500
600	baasii	4	2400
1000	Tubbaan	3	3000
2000	corroded	1	2000
3000	marrii	1	3000
3754	Zuurich	1	3754

e) SPOR

Rank	Word	f	f*r
1	akka	1127	1127
5	isaanii	323	1615
10	barattoo	240	2400
20	kana	166	3320
30	kanaan	120	3600
40	yeroo	96	3840
50	qaama	87	4350
60	shaakals	75	4500
70	keessa	68	4760
80	akkaataa	63	5040
90	Haala	59	5310
100	fakkeess	53	5300
200	k.k.f.	31	6200
300	koo	22	6600
400	haarawaa	16	6400
500	ta'een	13	6500
600	taphatto	11	6600
1000	kaasuun	6	6000
6551	Zaaphoo	1	6551

f) AON

Rank	Word	f	f*r
1	Gadaa	36	36
5	sirna	18	90
10	hin	14	140
20	ittiin	7	140
30	yaada	6	180
40	jiran	5	200
50	Tuulamaa	5	250
60	Oromoota	4	240
70	aadaa	3	210
80	Buttaa	3	240
90	haalli	3	270
100	lakkoofs	3	300
200	Kan	2	400
300	argamsii	1	300
400	dhufaati	1	400
500	guddachu	1	500
600	kanarra	1	600
855	Yoo	1	855

DECLARATION

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

Wakshum Mekonnen
19 May 2000

The thesis has been submitted for examination with my approval as university advisor.

Nega Alemayehu (Ph.D.)

19 May 2000