DECLARATION

THIS THESIS IS MY ORGINAL WORK AND HAS NOT BEEN SUBMITTED FOR DEGREE IN ANY OTHER UNIVERSITY

ZELALEM SINTAYEHU

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

WERKSHET LAMENEW

MILLION MESHESHA

DEDICATION

This thesis is dedicated to my parents with love and appreciation, for continuously putting my life into perspective during the most difficult and busiest of times. Their gift to me is more than I can ever realize.

ACKNOWLEDGEMENT

First and foremost I thank God for he is always besides me in all my trouble times. I am also deeply grateful to my advisors Ato Werkshet Lamenew and Ato Million Meshesha, for their constructive suggestion and encouragement during this research work. I would like also to thank all SISA staffs, especially Ato Getachew Jemaneh, Dr. Taye Tadesse, Ato Tesfaye Biru, and W/o Tewabech for their good treatment during my stay in the school.

My special thanks goes to Office of the Federal Auditor General (OFAG) for giving me the opportunity to participate in this program. No less thanks is given to Ethiopian News Agency (ENA) for letting me use its data, and the friendly staffs particularly Ato Sileshi, Ato Teshager, Ato Simeneh and Ato Belsti for their valuable support throughout the course of the research.

Many people share in the credit of everything good that comes out of this work. Especially, Abreham Kassa, Melaku Kebede, all my friends in OFAG, W/t Robe Desta, W/o Asnakech Tefera and W/t Almaz Assefa deserve special thanks for their contribution and encouragement.

Last, but not least, I thank Dr. Nega Alemayehu who sow the seeds of IR into my mind and initiated the area of this thesis. Credit should also be given to Ato Gebreselassie of Custor Computing for the clear description of Visual Geez Amharic Software.

Zelalem Sintayehu Shibeshi

TABLE OF CONTENTS

DECLARATION ..... **I**

DEDICATION ..... **II**

ACKNOWLEDGEMENT ..... **III**

TABLE OF CONTENTS ..... **IV**

LIST OF TABLES AND FIGURES ..... **VIII**

FIGURES VIII

LIST OF ANNEXES ..... **IX**

ABSTRACT ..... **X**

CHAPTER ONE **1**

INTRODUCTION **1**

1.1 BACKGROUND 1

1.2 ORGANIZATIONAL STRUCTURE OF ENA 2

1.2.1 The Desks *2*

1.2.1.1. The regional desk 3

1.2.1.2. The city desk 3

1.2.1.3. The foreign language desk 4

1.2.1.4. Broadcast monitoring desk 4

1.2.2 The Current News Management System *5*

1.3 STATEMENT OF THE PROBLEM 7

1.4 JUSTIFICATION OF THE STUDY 10

1.5 OBJECTIVES OF THE STUDY 12

1.5.1 General Objective *12*

1.5.2 Specific Objectives *12*

1.6 METHODS 13

1.6.1 Data Collection *13*

1.6.1.1. Literature review 13

1.6.1.2. Document Analysis 13
1.6.1.3. Interview 13
1.6.2 Sampling Technique *14*
1.6.3 Development of Amharic News Classifier (ANC) *14*
1.6.4 Program Development Tool *15*
1.6.5 Testing Technique *15*
1.7 SCOPE AND LIMITATION OF THE STUDY 16
1.8 ORGANIZATION OF THE THESIS 16
CHAPTER TWO 18
AUTOMATIC CLASSIFICATION 18
2.1 INTRODUCTION 18
2.2 VIEWS REGARDING THE MEANING OF AUTOMATIC CLASSIFICATION 18
2.3 APPROACHES TO AUTOMATIC CLASSIFICATION 19
2.4 BASIC CONCEPTS IN AUTOMATIC CLASSIFICATION 20
2.5 STEPS IN AUTOMATIC CLASSIFICATION 22
2.5.1 Document Analysis *22*
2.5.1.1. Document representation. 23
2.5.2. Feature Selection *25*
2.5.3 Term Weighting *28*
2.5.4 Representing Classes *31*
2.5.5 Matching (Similarity) Technique *32*
2.5.5.1. Weighted cosine similarity measure. 33
2.5.5.2. Dice coefficient. 34
CHAPTER THREE 35
AMHARIC NEWS CLASSIFICATION 35
3.1 INTRODUCTION 35
3.2 NEWS CLASSIFICATION 35
3.2.1 General *35*
3.2.2 General Principle of News Classification *37*

3.3 NEWS CLASSIFICATION SCHEME AT ENA 38
3.3.1 General *38*
3.3.2 The Amharic News Classification Scheme *39*
3.4 AMHARIC NEWS STRUCTURE 41
3.4.1 Header *42*
3.4.2 Headline *42*
3.4.3 Lead *42*
3.4.4 Body *42*
3.5 THE AMHARIC WRITING SYSTEM 44
3.5.1 The Amharic Characters *44*
3.5.2 Punctuation Marks *45*
3.5.3 Problems In The Amharic Writing System *46*
3.5.3.1. Problems regarding consonants with different form 46
3.5.3.2. Problems related to certain interchangeably used orders 47
3.5.3.3. Problem regarding compound words 47
3.5.3.4. Problem regarding abbreviations 48
3.6 AMHARIC STEMMER 49
3.6.1. Suffixes Considered *49*
3.6.2 Prefixes Considered *50*
3.7 AMHARIC SOFTWARE 50

CHAPTER FOUR 53

DESIGN OF THE AUTOMATIC CLASSIFIER 53

4.1 INTRODUCTION 53
4.2 NEWS FOR EXPERIMENTATION 54
4.3 PREPROCESSING 55
4.4 WORD IDENTIFICATION 56
4.5 CHECKING THE VALIDITY OF A WORD 58
4.6 CHANGING CHARACTERS TO THEIR COMMON FORM 58
4.7 WORD STEMMING 59
4.8 STOP WORD REMOVAL 62

4.8.1 News Specific Stop Words *62*
4.8.2 Common Stop Words *63*
4.9 VECTOR TABLE GENERATION 63
4.10 AUTOMATIC CLASSIFICATION 67
4.11 THE PROTOTYPE 69
4.12 TESTING 70
4.13 DISCUSSION 71
CHAPTER FIVE 73
CONCLUSIONS AND RECOMMENDATIONS 73
5.1 CONCLUSIONS 73
5.2 RECOMMENDATIONS 75

LIST OF TABLES AND FIGURES

Tables

Table 3.1 Main Classes of Amharic News Classification Scheme 41

Table 3.2 Structure of Amharic News Items 43

Table 3.3 Amharic Alphabets with Different Forms 46

Table 3.4 Example of the Possible Combination of Suffixes from ‘ዎች’ and ‘ን’, ‘ም’, and ‘ና’. 50

Table 4.1 Number of News Items in Experimental and Validation Group 56

Table 4.2 Vector Representation of Document No. 38215. 64

Table 4.3 Vector Table of ANC 65

Table 4.4 The Top Ten Distinguishing Words of the Three Classes. 66

Table 4.5 Number of Keywords in Each Class 66

Table 4.6 Number of Common Terms Between Classes 67

Table 4.7 Accuracy in Each Class 700

Figures

Figure 4.1 Overview Diagram of ANC 54

Figure 4.3 Flowchart of ANC 68

Figure 4.2 Main Screen of ANC 69

LIST OF ANNEXES

Annex 1: Organizational Structure of ENA 84
Annex 2: Amharic News Classification Scheme 85
Annex 3: Amharic Fidel 94
Annex 4: List of Suffixes 95
Annex 5. ANC Program Code 96
Annex 6: Sample Vector Table 115

ABSTRACT

To organize its news stock efficiently and to facilitate the storage and retrieval of news items, Ethiopian News Agency (ENA) use a classification scheme developed in-house. With its large volume of news items produced each year, ENA is facing problems in classifying news items timely. This research has come up with Amharic News Classifier (ANC) that has the capability of classifying Amharic news items into the predefined classes automatically based on their content.

The development of automatic document classification system passes through different steps and there are different methods that can be used at each step. This research used statistical techniques of automatic classification in all the steps. The steps in automatic classification include document analysis, generation of document and class vectors based on document and class representatives, and matching document and class vectors to determine the class where a document belongs.

The process of document analysis requires some preprocessing activities such as stemming and stopword removal, which are language dependent. In this research, the key terms are stemmed using a simple depluralization and suffix and prefix removal program developed for this purpose. A database of stop word list, which contains most frequently occurring Amharic words, was also developed. In addition, problems related to Amharic language script were considered during text processing.

To identify document representatives, $tf \times idf$ weighting technique is used. Class vectors, also called centroid vectors, are generated by computing the average value of document vectors. After identifying class representatives from the learning data set,

cosine function is used as a matching technique to automatically classify the test data set that had no relation with the construction of the class vectors.

The overall result of this research has showed that statistical techniques can be used to analyze Amharic news items and classify them automatically into predefined classes. After training the classifier, 273 out of 321 news items were correctly classified by the system. The result is very promising, however, additional works are recommended in order to implement the system.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

The public needs to be updated on important public events such as actions of governments, social or economic trends, education, and international relationships, which are often referred to as hard news (ENA, 1993a).

Other news types include gossip items about celebrities, off beat incidents, sensational crime cases, sudden happenings, such as fire, a jury verdict, or a parliamentary decision etc., which present public interest (Ibid.). News is often the account of something rare and out of ordinary.

In general, news is a prompt, “bottom line” recounting of factual information about events, situations, and ideas (including opinions and interpretations) calculated to interest an audience and help people cope with themselves and their environment (ENA, 1993b).

For this purpose any government establishes an organization that facilitates the free expression of opinions and views among the people. As a result in Ethiopia, the Ethiopian News Agency (ENA) was established 58 years ago with the name “Agance Direcsione” under what is used to be known as the Ministry of Pen, which currently is called Ministry of Culture and Information (Ibid.).

In 1967, however, its name was changed to its current name, ENA. Finally, the Transitional Government of Ethiopia embarked on restructuring the news media in 1995. The agency then became an autonomous public agency having judicial personality under a board whose members are drawn from a cross-section of the society.

The responsibility of the agency, as proclaimed by the Transitional Government of Ethiopia, is to gather and distribute balanced and accurate news and news materials, concerning Ethiopia and the rest of the world, in Amharic and English languages (TGE, 1995).

1.2 ORGANIZATIONAL STRUCTURE OF ENA

Like any public organization in Ethiopia, the structure of ENA is divided according to the distinctive nature of the work at the agency. The staffs are identified as two groups: technical and administrative. The staffs of the information desks (journalists) and other technical staffs are administered under the Board, while the support staffs of the Agency are administered under the federal civil service laws.

In May 2000, ENA had some 100 journalists actively engaged in gathering, writing and editing of news and news materials. Of these, 32 junior reporters, 32 senior reporters, 19 assistant editors, 10 editors, 6 deputies editor-in-chiefs and 1 editor-in-chief. The organizational structure of the agency is attached as Annex 1.

1.2.1 The Desks

There are four desks within the Information Coordination Department classified according to the distinctive circumstances related to the source and destination as well

as the manner of gathering and disseminating news and news materials. These are (1) the Regional Desk, (2) the Addis Ababa Desk (also called the City Desk), (3) the Foreign Languages Desk and (4) Broadcast Monitoring Desk.

1.2.1.1. The regional desk

This desk is responsible for gathering news and news materials from across the nation except Addis Ababa and its surrounding localities.

ENA's correspondents at the 15 main branches and 22 sub-branches in the various national states are directly accountable to this desk. Most of the news on the various social, economic, political and cultural lives of the people in all corners of the country comes to the mass media, and through the efforts of the staff of this desk, to the minds of millions of the people.

1.2.1.2. The city desk

As it is currently called, Addis Ababa desk has been performing news gathering and dissemination activities for nearly four decades now, only that its focus is confined to the capital city. Being the seat of the Federal Government and several international organizations, the people at this desk are very busy.

The desk has standby reporters at Bole International Airport, and regularly covers events at the Parliament, the Federal High Court, the Palace and the Prime Minister's Office.

1.2.1.3. The foreign language desk

This desk is the main escritoire inside ENA that provides the foreign community with important and up-to-date news and information about current political, social and economic affairs of the country.

The desk acquaints the English speaking community with issues that have optimal importance. Significant stories sent by reporters of the regional desk are translated into English at this desk.

1.2.1.4. Broadcast monitoring desk

This desk is responsible to monitor more than 20 international broadcast stations around the world. It then supplies the other information desks with current news and news materials, thereby enriching the output of the agency.

The service is furnished with professional equipment that allows the multilanguage staff to render the desired service.

In general, every minute and hour the news agency receives a vast quantity of news from different places through the above desks. The agency receives overseas news from different news agencies, such as AFP, Xenuoa and Reuter. The correspondents who are in the different regions of the country send news through the newly established wide area network. The monitoring desk compiles the news that it gets from the different mass media and analyzed as important. As a result the news agency is flooded by news and views every day.

For example, with the four news desks ENA has produced a total of 18,868 news items last year (1992 E.C[1]) alone (ENA, 2000), of which the Amharic news accounts for 62% of the total news.

1.2.2 The Current News Management System

Recently, a computerization project was initiated by the agency. The project produced a networked software system called ENASoft for the purpose managing news items that are produced inside the agency.

The software is bi-lingual –Amharic and English – and it is used to create, edit, manage and archive news items. To enter and edit Amharic text, Visual Geez Amharic software is imbedded into the software. The City and Regional Desks prepare Amharic news items.

The software is a client-server application developed to store and edit news items in a networked environment. Currently, 15 branches are connected to the central server and journalists from these branches send news material through the network to the central server.

The software is a database application, whose front-end is made using Microsoft Visual Basic Programming Language. The database is developed using Microsoft Access. There are 17 tables to store different information. The database is used to store only the metadata about every news item, which includes headline, journalist name, editor name, classification code, sub-classification code, slug, keyword etc. Every news is assigned an ID through an auto number field.

[1] E.C. refers to Ethiopian Calendar.

The news ID is then used to create an “rtf” file that holds the actual news data without the above information (headline, slug, author etc). The ID along with the text "ENA" is assigned for the file name of the news. The files are stored in the folder of each news desk. That is, all news produced by City Desk, for example, are stored in the folder DocA (to mean documents of Addis Ababa).

The database stores metadata of both Amharic and English news items altogether in one table. However, data on Amharic news items is entered into the database in Amharic and data for English news items in English. The ‘NEWSID’ field, which contains the name of the file (without extension), has different formats for the two news types. In this field Amharic News items have value which starts with "**ኢዘአ**", while English news items start with “ENA”, and it is similar to their file name. For example, this field contains **ኢዘአ**41345 for a news item with an id 41345, if it is Amharic news, otherwise it will contain ENA41345.

The software has three major components that work as integrated system.

1. ENASoft Regional :- is used by regional sites to create, manage and transmit news to ENA headquarter.

2. Regional News Monitor: - is used to monitor incoming news from regional sites, which are then automatically routed to the ENASoft Central for processing,

3. ENASoft Central: - is the main system to process and manage news stories at the headquarter.

The software also provides a security feature that allows access to data only to eligible users according to their privilege.

1.3 STATEMENT OF THE PROBLEM

Every news is processed (shaped and sharpened) by editors and assistant editors before it is dispatched to reach the public. In other words, the raw data received is filtered, processed, assembled and then passed as finished product (Whatmore, 1978). An editor does the processing job, where he/she evaluates the news, cuts it to length, fill out background, checks facts, punctuates, indicates layout and type faces. Above all news has to be new, although there is a continuity of running stories. That is, the originality of the news should also be checked.

With a vast inflow of daily material to news agencies, each day's work must be completed before the next brings another bunch of news for processing. As stated above, checking whether the coming news as new or not and finding already existing news that can be used as background and checking facts are the daily routines of any agency.

As a result, developing a system of rapid reference to the news that have gone by, and to store facts for its own use is a pressing need of ENA. The retrieval of stored news for background purpose, for example, should be very fast, as the news to be produced has a very high obsolescence ratio (Ibid.).

Therefore, for reasons mentioned above the news stock have to be organized properly so as to facilitate storage and retrieval. Humans use classification techniques to organize things in various activities of their life. In fact, classification occurs in a wide

range of human activity, and as Kumar (1999) says, human progress would be impossible without classification.

Classification techniques are applied in an information storage and retrieval system in order to facilitate access to, and use of the system (Hunter, 1995). Especially in a system where there is large collection of documents, retrieval of a given document or sets of documents can be possible if the collection is organized systematically. Actually, as Cheng and Albert (1995) quoting Davies says, the greater our ability to store information, the more attention must be paid to the problem of organizing and retrieving it.

There are different classification schemes developed for different application areas. Advantages of using classification schemes include improved subject browsing facilities, potential multi-lingual access and improved interoperability with other services. Classification schemes vary in scope and methodology, but can be divided into universal, national, general, subject specific and home-grown schemes.

Libraries are the most beneficiaries of the classification process. They use different scheme, like Dewey Decimal Code, Library of Congress, Ranganatan's faceted classification scheme etc. These schemes can also be used for other knowledge organization purposes other than books. In fact, there is no one classification scheme which is suitable for all purposes and the choice, or design, of a classification scheme will be governed by factors such as type of information system, the objectives of the system, and user requirement (Hunter, 1995). Based on these principles, ENA is using an in house developed classification scheme.

The news agency has been using fewer classes (4 to be specific) to store its news items up until late 1999. Since then, however, the classification scheme has been revised to provide a more detail content-based retrieval facility. The revised classification scheme contains 17 main classes and a maximum of 300 subclasses.

As of May 2000, ENA has also implemented a networked computer system (ENASoft) to prepare, store and dispatch news produced by the agency. When entering a new news item, the system requires the classification, and subclasification code to be entered manually along with the news. The reporters and/or editors are required to classify the news and enter the code when they input the news using the software. In fact the software provides the list of classification codes in a list box.

However, the journalists do not have the necessary skill on how to give class codes for news items. Besides, there are no enough information professionals who can assist them in the classification process. As a result, they make mistakes when entering class codes for the news items[2]. For instance, since the software forces a class code to be filled in before the news is typed in, a hasty journalist can select a classification code that may not be related to the news content from the list.

There are also many problems observed with a human classifier that affect classification results, which include: perception, comprehension, and judgment (Cheng and Albert, 1995).

[2] Ato Teshager Shiferaw - Editor

In view of the foregoing discussions, it is felt that an automatic news classification system should be developed that assists journalists in addressing the problems encountered during manual classification of news items.

In the area of information storage and retrieval, extensive research has been done to test the possibility of automatic classification of documents. Some automatic classification procedures are concerned with clustering documents without priorly defined classes, which is usually referred to as document clustering (Larson, 1992; and Losee and Hass, 1995). On the other hand, there are also techniques for classifying documents automatically to predefined classes (Choi, et. al. 1996; and May, 1997).

This thesis work therefore is an attempt to explore methodologies for classifying news items automatically into predefined classes. In this research news item and documents are interchangeably used.

1.4 JUSTIFICATION OF THE STUDY

In a large collection of electronic documents, it is difficult to manage and classify documents manually; especially when timeliness is a very important factor as is the case of ENA. In fact, as classification is time consuming and expensive process it is obvious that investigation of the use of automated solutions are worthwhile (Takkinen, 1995).

Specific reasons that justify the consideration of automatic classification may be summarized as follows:

- The timeliness characteristic of news item is achieved by classifying news items every time (if possible every day).

- Currently journalists themselves who are not specialists in this area do the classification process. If this is to be continued, it will take long time for them to learn and use the classification scheme properly and correctly.

- It will decrease the burden of the journalist, where classifying news item is not their main responsibility and assumed as wasting of time.[3]

- Experience shows that, different reporters classify the same news in different classes.[4]

In addition to this, it is observed that, other news agencies are using a detailed classification scheme. Reuters, for example, has more than 200 main classes. And the reason for ENA's narrower classification scheme is lack of skilled manpower to implement a detailed classification code[5].

It will also be very cumbersome to reclassify old documents that were broadly classified. Therefore, to improve the discovery of old news stories, automatic method of news organization might be necessary.

The agency may also provide such facilities as a means of production of different bulletins, news analysis and feature stories if there is an efficient storage and retrieval system that properly classifies news items.

[3] Ato Teshager Shiferaw - Editor
[4] Ato Sileshi Tessema – Department Head
[5] Ato Simeneh Mekonen – Coordinator of the classification scheme

Researches showed that automatic classification can be helpful in the area of mechanized procedures for sorting letters on the basis of machine-read post codes, assigning individuals to credit status on the basis of financial and other personal information, in preliminary diagnosis of a patient's disease in order to select immediate treatment while awaiting definitive test results (Aas and Line, 1999). Other areas include determining the topic area of an essay; deciding to what folder an e-mail message should be directed, and deciding to which news group a news article belongs (Hsu and Sheau-Dong, 1999).

1.5 OBJECTIVES OF THE STUDY

1.5.1 General Objective

The general objective of this study is to investigate the characteristics of Amharic news items that flow into ENA and design a prototype using statistical techniques that automatically classify news items into their predefined classes based on their content.

1.5.2 Specific Objectives

- Review literature on the concept of classification and the available techniques of classifying documents automatically.
- Build stop word list
- Select classification techniques to build an automatic Amharic news classification system.
- Using the selected technique design a prototype that automatically classifies news items according to their content.

- Test the system developed to measure its performance
- Make recommendations on what should be done next.

1.6 METHODS

1.6.1 Data Collection

1.6.1.1. Literature review

To get an understanding of the various techniques of automatic classification, relevant published documents, materials on the Internet and journal articles are reviewed.

1.6.1.2. Document Analysis

To have further understanding of the manual Amharic news classification system at ENA, the following documents are analyzed.

- The current classification scheme
- Manually classified Amharic news items
- ENASoft software and documentation

1.6.1.3. Interview

To have a clear idea on the manual classification scheme and the problem area, interviews and discussions were conducted with appropriate staff of the agency, especially those involved in the development of the news classification scheme.

1.6.2 Sampling Technique

The population of this study is two year data. However, news items of the first year are stored inside the software with the old classification code. This means, the agency has only one-year data classified with the current (revised) classification code.

In this kind of research the data set required is of two types: experiment set and test set. Experiment (training) data is used to identify class representatives, and test data to evaluate the performance of the prototype developed.

For both training and testing purpose, the data having the current classification code are considered. As a result, eight month data is taken for training and the latest four month data is taken as test data.

1.6.3 Development of Amharic News Classifier (ANC)

Automatic document classification has different steps and there are different methods that can be used at each step in the system development. The methods are based on machine learning, statistics, or natural language processing (Choi et. al., 1996). However, in this research statistical techniques are used.

The reasons for using statistical techniques are varied. First, as described by Yang, (1999) statistical techniques are easier to work with. Second, because of the researcher's background, it is easier to use statistical concepts that do not require new ideas in the limited time given for the research. On the other hand, linguistic analysis, for example, has proved to be expensive to implement Cheng and Wu (1995). Third, the statistical approach has been examined and tried ever since the days of Luhn and produced good result (Ibid.).

The key terms are stemmed using a simple depluralization and suffix and prefix removal program. After identifying class representatives from the training data set, cosine function is used as a matching technique to automatically classify the test data. A database of stop word list, which contains most frequently occurring common Amharic words and news specific stop words was also developed.

1.6.4 Program Development Tool

The prototype was coded using Visual Basic programming language. The programming language was selected because:

- The language is an object-oriented programming language which can be easily integrated with other applications,
- The source code can be modified easily for further modifications,
- The language has easy to add menus and other features like user-friendliness (graphical user interface),
- The news management software (ENASoft) is also developed using this programming language.

1.6.5 Testing Technique

Whatever the purpose of the classification system, the 'goodness' of the system should finally be measured by its performance during storage and retrieval. The prototype developed is tested for effectiveness using the test data set. The result, which is automatically assigned code, is checked with the result of a code given by a human

(expert) classifier, and the percentage of correct assignments by the system was taken to decide the systems effectiveness.

1.7 SCOPE AND LIMITATION OF THE STUDY

The system developed did not entertain classification of English news items. Another similar research should be done to investigate the integration of automatic classification of English news items to the present system.

There is an Amharic stemmer developed by Nega (Abiyot, 2000). However, the researcher could not get the stemmer to create stemmed words (terms) for class and document representatives. The list of terms (words) would decrease a lot if a complete stemming program were used.

In addition, due to time constraint, the correctness of the existing classes (manual) was not checked using the principles they base. Nevertheless, the validation of codes that was given to documents in the sample in relation to the existing chosen classes were done by an expert from ENA.

1.8 ORGANIZATION OF THE THESIS

This thesis is divided into five chapters. The first chapter is introductory, in which the environment of the research is described. The chapter also presents statement of the problem, objective of the study and the methods followed. In chapter two the techniques available in the area of automatic classification are reviewed, and the techniques followed in this research are described in detail as well.

As the research is done on Amharic text documents, chapter three reviewed language aspects that should be considered in the development of text analysis. In addition, the classification scheme implemented at ENA is also reviewed in this chapter.

Chapter four discusses the development process, which is the main concern of the research, in detail. The approaches followed in each step are explained. The result achieved in the research is also presented in this chapter. The conclusions drawn from the study and the recommendations are stated in chapter five.

CHAPTER TWO

AUTOMATIC CLASSIFICATION

2.1 INTRODUCTION

Ever since the advent of computers, the idea of making computers to mimic human beings was the focus of many researches. Natural language processing, voice recognition and automatic classification are just to list few attempts in this line. This chapter tries to explore concepts behind automatic classification.

The chapter also discusses the basic steps of automatic classification. A review of the different approaches towards automatic classification is also made.

2.2 VIEWS REGARDING THE MEANING OF AUTOMATIC CLASSIFICATION

In the area of automatic classification, there are two different views towards the meaning of automatic classification.

The first view considers automatic classification as a technique of classifying documents automatically without having any prior knowledge of the categories where the documents would be classified. According to this view, the classification process is expected to create the classes (categories) based on the similarity that exist among the documents. In fact, this view is often referred as cluster analysis. Basically, cluster analysis is the identification of classes, and as described by Willet (1988), it is potentially misleading to refer cluster analysis as automatic classification.

The other view considers automatic classification as a process of classifying documents automatically into their predefined classes. Actually, this is the correct view of automatic classification (Rasmussen, 1992), for the meaning of classification itself is putting documents into defined categories. This concept is also referred as text categorization. Most of the documents reviewed in this thesis use this name to refer to automatic classification.

This thesis follows the second view of automatic classification, with the aim of developing a system that will classify documents (news items written in Amharic language) automatically into already predefined classes.

2.3 APPROACHES TO AUTOMATIC CLASSIFICATION

Automatic classification is concerned with the construction of a procedure that will be applied to continuous sequences of cases, in which each new case must be assigned to one of a set of predefined classes on the basis of observed attributes or features (Michie et al., 1994).

Solutions to the problem of automatic classification are done using different techniques. The techniques employed include machine learning, statistical, knowledge-based, Natural Language Processing (NLP) or a combination of them (Aas and Eikvil, 1990). May (1997) even tried to develop a solution using simple string matching technique.

As discussed in the first chapter of this thesis, since statistical techniques are chosen to develop an automatic classification system for Amharic news items, much emphasis is given in this chapter to these techniques. However, detail explanations on the other

techniques can be reviewed from the following references. Michie et al. (1994), Losee and Hass (1995) and Cohen and Singer (1998) discuss the use of natural language processing techniques to automatic classification; Lin et al. (n.d.) and Ruiz and Srinivasan (1998) discuss the use of machine-learning approach, specifically neural network method; Cahn and Herr (1977) and Pullock (1988) describe knowledge-based techniques to implement automatic classification and Blosseville (1992) explains how to do automatic classification by merging NLP, statistical and expert system techniques.

2.4 BASIC CONCEPTS IN AUTOMATIC CLASSIFICATION

Automatic classification attempts to select the correct predefined class based on the characteristics of a document to be classified and the characteristics of the documents previously assigned to each class. In other words, the process involves training systems to recognize characteristics of documents belonging to a particular classification group.

It is obvious that the goal of any classification process is to group similar documents together. This implies, the procedure of classifying documents into groups require a quantitative measure of the "likeness" of the document for a given class, and the separation of unlike ones. In other words, it involves measuring similarity of a document with the different classes.

Whenever we talk of similarity of items, we are talking of their similarity in some respect (usually with their attributes). The attributes can be anything that characterizes the bases of the classes for the purpose of the classification process. For example, size

could be one attribute to classify various dwellings as villas, bungalows etc. On the other hand, capacity could be used as an attribute to classify different sized cans. Therefore, the first thing in the process of automatic classification is to identify the attributes of documents and classes so that the matching of the two becomes simple or at least possible.

When it comes to documents, for a document to be classified under a given class, it must be ascertained that its subject matter relates to the area of discourse. This actually seems simple for a human being. The question is whether it is possible to program the computer to determine the content of a document and categories, and classify it accordingly. The process of understanding content of documents is often termed as document analysis.

In fact, representing documents for the purpose of classification is the first and important step in the process of automatic classification. In other words, documents must be represented through features[6] and the features will be used to determine the similarity of documents with the different classes. Similarly, the classes should also be represented in order to be able to evaluate their similarity with documents. Once the classes and documents are represented, then a matching technique can be applied to determine the similarity between documents and classes to identify the most similar class to a given document.

In general, automatic classification is the process of matching document representatives with class representatives to automatically assign classification codes

[6] Features are words and/or phrases that represent documents.

to documents based on the similarity that exist between documents and classes. The codes assigned to a document should be the code of a class that has the maximum similarity value with the given document. The following section discusses the steps followed in automatic classification.

2.5 STEPS IN AUTOMATIC CLASSIFICATION

2.5.1 Document Analysis

Document (text) analysis is the process of analyzing the text of a document to find meaning out of it. As discussed by Cheng and Wu (1995) classification requires text analysis, which is heavily dependent on the representation of the document. In fact, document analysis is very important especially when there are a huge number of electronic documents. The reason is, manipulating this huge collection for whatever purpose will be very difficult in terms of storage space and processing time.

Salton (1989) has classified three methods of text analysis, namely statistical, linguistic, and statistical linguistic. However, most researches have shown that linguistic analysis is very expensive to implement and the results obtained are also generally unsatisfactory (Cheng and Wu, 1995). Likewise, Artificial Intelligence (AI) based NLP techniques are also computationally intensive (Lin, n.d.). On the other hand, statistical analysis including statistical-linguistic analysis, has been examined and tried ever since the days of Luhn, and found to be successful (Cheng and Wu, 1995).

Generally speaking, text analysis is the analysis of documents to find efficient document representatives for the purpose of storage and retrieval, which is also called

indexing. The purpose of indexing is to obtain a number of descriptors, which act as surrogates for the document. This means, given a written text in natural language, it is essential to represent the information contained in the text by one or more entries, variously known as indexes, keywords, or key terms. In fact, the classic models in IR consider that each document is described by a set of representative index terms. These descriptors, or keywords, can be obtained manually or automatically by computer analysis of the document file, abstract or text. The problem is to choose "good" terms, which collectively reflect the information content as accurately as possible.

The representation can be by analyzing the whole document or only part of the document. For instance, Losee and Haas (1995) have tried to use the titles of the documents for representation purpose while Kwok (1975) used title and cited titles to represent documents. On the other hand, Enser (1985) used back of book indexes to classify books. In fact, as discussed by Enser (1985) documents can be suitably classified by examining only limited portions of the document, which can save considerable time and money. Therefore, the first step in document analysis is deciding on the representation of documents.

2.5.1.1. Document representation.

From the outset, documents can be considered as a stream of characters. However, for the problem of automatic classification these streams should be transformed into representatives, which are suitable for the process of classification.

As suggested by researchers in the IR community, words seem to be good features for many classification tasks. This means, the representation of documents can be

simplified from stream of characters to sequences of words. Inaddition, it is also mentioned that, the order of the words does not matter. This means documents can be seen as a bag of words.

On the other hand, documents can be represented using word-based and/or phrase-based indexing. But Lewis (1992b) showed that lower effectiveness level is achieved for a syntactic phrase indexing than for word-based indexing on text categorization tasks. In addition a phrase-based indexing requires some linguistic knowledge. Because of this, word-based indexing is considered in this research.

The basic idea behind word-based representation (also called bag of word representation) is that a document that describes a certain concept is more likely to have words from that domain. In other words, a document about “crime” will have the word “crime” or its synonyms in it. And a document that contains the words "boys", "girls", "teachers" "schools", "reading" etc. probably deals with education.

This technique is the main, if not the only technique used to represent documents in most experiments and applications of automatic classification (Scott and Marwin, 1998).

The problem, however is that all of the words in the documents cannot be considered as features of documents. Rather only the good descriptors should be taken to minimize the problem of storage and computation time mentioned in the previous section. In fact, in the field of text categirization, it has been seen that maximum performance is often not achieved by using all available features, rather by using only

a “good” subset of those (Joachim, 1996). The problem of finding these “good” subset features is called feature selection.

Having too few features can make it impossible to develop a good system, but having features that do not help to discriminate between classes also adds noise. The following section reviews the different techniques followed by researchers to identify document features for the purpose of automatic classification.

2.5.2. Feature Selection

Any classification technique should first process a document or part of it, or its surrogates to identify “good” representatives of documents. The first step in identifying the features (which can be words or phrases) is parsing the document to convert it to list of words or phrases. The selection of features can be taken freely from the text, or just from part of the text.

There are different techniques of feature selection. But, statistical techniques to feature selection are widely used in the area of automatic classification. Statistical techniques to text analysis and feature selection are based on term frequency[7]. The pioneering work of Luhn showed statistical analysis of the words in a document will provide some clues as to its content. The idea behind this principle is, a term that is frequently present in a document is useful to represent the document. The problem, however, is how frequent should a term be found in a document to be accepted as an index term.

On the other hand, the selection of words can also be made from words making the text or by consulting a dictionary or thesaurus file. Though it seems simple to

[7] Term frequency is the number of times a term occurs in a given document.

impliment, no noun dictionary is developed for Amharic text, and as a result this method cannot be used in this study.

One technique proposed by Luhn says both high-frequency words and rare-frequency words are unlikely to be able to represent documents and should not be considered (Cheng and Wu, 1995). The former are discarded because they occur too often to indicate the subject matter and the later because they are too rare. Therefore, the remaining intermediate frequency terms are assumed important.

It has also been proved that, grammatical function words such as "the", "and", "of" and "to" exhibit approximately equal frequencies of occurrence in all the document collection and should not be considered in the selection process. The indexing technique, therefore, should consider frequency of the content bearing words only. The frequency of occurrence of non function words[8] may actually be used to indicate term importance for content representation.

There are also other techniques discussed for the selection of index terms. Pao (1989), for instance, described three methods of selecting index terms. The first method she suggested is to take the top 5% of the words frequently occurring in the word list of the document. The second is to take words appearing at least twice within the same paragraph after removing stop words. The third method, which involves the knowledge of normal word frequency usage of each word in a given language, is to take a word if its relative frequency exceeds that found in the normal usage of the term in the given language.

[8] Non function words are terms that relate to the subject of a given document.

Salton (1989), on the other hand, suggested an index extraction method that can be done using the following three steps:-

1. Eliminate common function words from the document by consulting a special dictionary, called negative dictionary or stop word list, which contains a list of high frequency words.

2. Compute the term frequency tf_{ij} for all remaining terms t_j in each document d_i, specifying the number of occurrences of t_j in d_i.

3. Choose a threshold frequency T, and assign to each document d_i all terms t_j for which tf_{ij} >T.

The main task in the above procedure is to find the threshold frequency value that actually affects significantly the result of the classification. A slightly low threshold will lead to close classification, while a high threshold will lead to a broad classification[9] (Cheng and Wu, 1995).

In fact, a high frequency term is acceptable for indexing purpose only if its occurrence frequency is not equally high in all the documents of a collection. That is, it must have a discrimination power between documents (classes, in our case). This implies, some terms are more important than others or have high discrimination power than others.

[9] Close classification is classifying each subject as completely or as fully as possible, and Broad classification is classifying the material only in main divisions and subdivisions without using the minute breakdown of individual categories.

Salton (1989) suggests a formula to evaluate a term discriminating power by an inverse function. That is, in a collection of N documents, if a term T_j occurs in df_i documents, then the discrimination value of the term is N/df_i.

In other words, even after we identified the index terms, not all the terms are equally important for classification. Some are more important than others or have high discrimination value. The technique of assigning importance value for terms is term weighting. In IR weighting is done to rank results for a given query.

Term weighting assigns indications of importance to terms. Weighting also helps to avoid false matches (Ardo and Trougott, 1997). As a basic step in automatic classification, the following section reviews the term weighting techniques used by the research community.

2.5.3 Term Weighting

As mentioned earlier term weighting assigns values to terms that indicate their level of importance. Different techniques can be used to assign weight to terms. Some used frequency of each term as weighting technique (Hoch, 1994; Borko and Bernick, 1963).

Other techniques include, giving weight to terms according to the occurrence of the term in a particular part of the document. In this regard, Choi et al. (1996) used a weighting technique that assigns highest weight for terms from the document titles. Jones (1986), on the other hand, proposes the following formula as an appropriate mechanism for computing the weight of a term in a document.

$$\text{Term Weighting} = \frac{\text{Frequency in the document}}{\text{Frequency in all documents}}$$

In general, Asa and Eikvil (1990) review the above and other weighting techniques used by the research community. The *tf*×*idf* is one among the weighting techniques reviewed. Since it is acclaimed for its good results, it is worth considering this technique here. In fact, Salton (1989) is the one who proposes this weighting technique. In *tf*×*idf* weighting, *tf* is used to refer the within document frequency and *idf* is the inverse document frequency.

This weighting technique assigns a weight to word *i* in document *k* in proportion to the number of occurrence of the word in the document, and in inverse proportion to the number of documents in the collection for which the word occurs at least once. In other words, the idea behind *idf* is to assign higher weight of importance to terms occurring in only a few documents.

Accordingly, the formula to compute the weight of word *i* in document *k* is given by:

$$a_{ik} = f_{ik} * \log(\frac{N}{n_i}), \quad \text{.....(2.1)}$$

where f_{ik} is the frequency of word *i* in document *k*,

N is number of documents in the collection, and

n_i is total number of times word *i* occurs in the collection (i.e. the number of documents in which term *i* occurs at least once).

In general, a higher weight assigned for a feature implies that the feature is more important for the classification process.

Once terms are given weight in the above manner, then the representation of documents will be not with a simple list of terms that was proposed in the previous sections, rather it will be with a weighted list. As a result, a document D can be represented with index terms and their weight as $D_i = (t_{i1} w_{i1}, t_{i2} w_{i2}, \ldots, t_{in} w_{in})$

Where D_i - document *i*,

t_{ij} - descriptor *j* in the document D_i, and

w_{ij} - weight of descriptor t_{ij}.

The above model, suggested by Salton (1989), is known as vector space model. The model, however, has one major problem, which is the high dimensionality of the feature space. In this model, there exists one dimension for each unique word found in the collection of documents, and processing will be extremely costly. For this purpose, many researches designed a mechanism for reducing this feature dimension. Details on these techniques can be obtained in (Han et al., 1999; Kar and White, 1978; and Borko and Bernick, 1963).

Actually, feature reduction should be done intelligently. In other words, if we become too aggressive in reducing the number of words, then we might loose critical information for the classification tasks (Han et al., 1999).

In vector space model we have a matrix developed from the collection of documents. The rows represent the documents, while the columns represent the terms. The entries of the matrix represent the weight of a word in a given document. Therefore, a document *A* is represented as:

$$A = (a_{ik}) \quad \text{.....(2.2)}$$

where a_{ik} is the weight of word *i* in document *k*.

Once documents are represented using the above method, classes can also be represented easily and the process of automatic classification will become simple.

2.5.4 Representing Classes

As discussed before, for the machine to be able to classify documents automatically, it should understand the characteristics of the different predefined classes. Therefore, in a similar manner to document representation, classes should be represented using terms so that matching of a new document with existing classes is possible.

This is usually done by having a labeled training document set for each class. In fact, experimental learning systems often gain knowledge from one set of data (usually referred as training set) and then use this knowledge to process another set of data (referred as test set). They may also learn from an entire set and then classify the set again (Losee and Haas, 1995).

After each document is represented using the technique mentioned in the previous section, the terms identified from the training documents are used to create the class representatives. The most common way of representing classes is using a class vector,

also called *centroid vector*. It is obtained by summing or averaging the weights of the various vectors of the training documents in each class. That is, in a set *S* of documents and their corresponding vector representations, we define the centroid vector $\vec{C}$, which is nothing more than the vector obtained by averaging the weights of the various terms present in the documents of *S*. In general, $\vec{C}$ appears as $\vec{C}$ = (C_1, C_2, ... , C_n) , where the ith element of the centroid, C_i, can simply be assigned the average value of the weights of the ith term in all the training set *S*, as:

$$\vec{C}_i .. = \frac{1}{N} \sum_{k=1}^{N} \vec{d}_{ki} \text{ , ...(2.3)}$$

Where N is the number of documents in the set *S*, and $\vec{d}_{ki}$ is the weight of term *i* in document *k*.

Once the centroid vectors are identified then it can be possible to classify an incoming new document, using similarity measures between the document and the classes.

2.5.5 Matching (Similarity) Technique

The matching step is the final step in the automatic classification process. Once all the classes are represented through centroid vectors, the system can classify new documents by matching the document vector with the centroid vector of each class. The document is then routed to the most similar class. In other words, the similarity between the document vector and the class vectors is computed to determine the class where the document belongs.

In the literature different similarity techniques are suggested, however, the most common is *cosine measure* (Han et al., 1999).

In another research, Cheng and Wu (1995) present the different similarity techniques in the area of automatic classification, including cosine coefficient. Because of their popularity and effectiveness, *cosine* and *dot product* are described below.

2.5.5.1. Weighted cosine similarity measure.

This similarity technique is based on the mathematical cosine rule, which measure the angle between two vectors to identify the proximity of the vectors in vector space. The formula defined for weighted cosine measure between document X and Y with their weight vector W and a set of terms T is the following (Han et al., 1999):

$$Cos(X,Y,W) = \frac{\sum_{t \varepsilon T}(X_t \times W_t) \times (Y_t \times W_t)}{\sqrt{\sum_{t \varepsilon T}(X_t \times W_t)^2} \times \sqrt{\sum_{t \varepsilon T}(Y_t \times W_t)^2}} \quad \text{..... (2.4)}$$

Where X_t and Y_t are term frequencies of word *t* for X and Y respectively, and W_t is the weight of word *t*.

The smaller the angle, the larger the cosine. So, classifiers based on this technique assign a document to the class which has the smallest angle between its vector and a given centroid vector. Therefore, the formula to assign class code for a document whose vector is represented, say by vector $\vec{x}$ is given by:

$$\arg\max_{j=1,2,\ldots,k} \left(\cos(\vec{x}, \vec{C}_j)\right) \quad \text{..... (2.5)}$$

where, *k* represents the number of classes and C_j is the centroid vector of class *j*.

2.5.5.2. Dice coefficient.

In this technique, the formula to measure the similarity between document D_i, and centroid vector C_j is given by

$$\frac{2\sum_k \left(D_{ik} \times C_{jk}\right)}{\sum_k D_{ik}^2 + \sum_k C_{jk}^2}$$

If binary terms weights are used, the Dice coefficient reduces to its popular form, which is,

$$\frac{2C}{A+B}$$

Where C is the number of terms that D_i and C_j have in common, and A and B are the number of terms in D_i and C_j respectively.

CHAPTER THREE

AMHARIC NEWS CLASSIFICATION

3.1 INTRODUCTION

In a system where there is large collection of documents, retrieval of a given document or sets of documents can be possible if the collection is organized systematically. The most common document organization method used in most information systems is classification. This chapter, therefore, discusses the principles of news classification in general, and Amharic news classification scheme in particular.

As Amharic is the official language of Ethiopia, ENA has implemented the news classification scheme in Amharic. In fact, it also developed English classification code for news items produced in English. The thesis, however, looks into the Amharic classification scheme.

The Amharic language faces some problems regarding its script that must be considered in the development of automatic text processing systems, automatic classification being one of them. Therefore, this is also another issue that is discussed in this chapter.

3.2 NEWS CLASSIFICATION

3.2.1 General

Because of the nature of current affairs, the media do not have to seek news; rather it arrives unsolicited in huge quantities. Official bodies and government, different

organizations and committees, political parties and pressure groups, all compete for attention, which then becomes good news.

However, as Whatmore (1978) says, the medias are not content merely to publish the news that flows in unasked. They are expected to uncover the news, to ask questions, to find out facts, not to accept easily, and finally to write what they find in the format accepted by their own medium.

Because of this, one can consider news agencies as news factories, where the raw material is received, processed and passed as a finished product. The processing includes selecting, assembling, adjusting, adding background, checking facts, and correcting.

To accomplish most of the above-mentioned tasks, journalists have to investigate previous news items. In fact, the retrieval of stored news in this regard is very frequent and urgent. As a result, journalists must be assisted by news libraries. The job of these libraries is to develop an efficient information system for the purpose of storage and retrieval. This shows the demand for classification, which is the most common technique in the organization of information for efficient retrieval. Actually, news libraries use classification as a main technique to organize the huge quantity of their news stock.

As Whatmore (1978) says, this organization or correct grouping of subject allocation is central to all activities of news libraries. Continuing his explanation, he added, "if it fails, the service fails because, unlike libraries of books, news libraries' stock does not arrive in prepackaged format".

3.2.2 General Principle of News Classification

As news falls out-of-date faster than any other kind of information, a classification scheme with response time ranging from 'at once' to a few minutes is expected. As discussed above, the main requirement of the classification scheme is to provide sufficient background information on any topic. However, the classification of news material is so difficult because the librarian tries to organize current information while it is still in a fluid state before its structure and underlying trends become apparent (Whatmore, 1973).

On the other hand, the theoretical classifications and hierarchies for subjects in their academic aspects are of little use when they appear as current affairs (Ibid.). Current affairs, by its nature, is a developing group of topics, subject to no structure, uncertain where it is going, leading in several directions simultaneously and consequently, of no shape, possessing no natural hierarchy, with no agreed terminology. In other words, news classification is not the science of absolutes but an art (Ibid.). As a result, every newspaper agency produces its own classification schemes.

Actually, Whatmore (1973 and 1978) has suggested the things that should be considered when developing a news classification scheme; some of them are discussed below.

As new events and ideas are covered through current affairs every time, the classification scheme must be hospitable to new aspects and angles. The classification must also reflect the way news is presented in the newspapers. In other word, it should have the form which inquiries are most likely to take.

Like other classification schemes, the arrangement must be easy to remember, minimizing the need to look-up before locating. In fact, "headings of classes should employ the same word as appear in the news" (Whatmore, 1973:211). They should also be precise and say exactly what they mean. On the other hand, the classification system should have the effect of excluding irrelevant matters.

In relation to specificity and exhaustivity, his suggestion is that the smaller the library, the more specific and direct the headings should be. As the collection grows, however, and element of classification will become essential, no other ground than convenience of access should be taken. In general, Whatmore has listed the guidelines, which he referred as “canons of classification” (Whatmore, 1978:52).

In reality, good classification is the application of judgment, foresight, knowledge and experience, all within the framework. In other words, the classifier should use imagination to anticipate demand and try to cater for the unexpected and ideally will have a file ready on any topic asked for. Having this in mind, ENA has produced its own classification scheme.

3.3 NEWS CLASSIFICATION SCHEME AT ENA

3.3.1 General

As discussed in the first chapter, ENA is the first news agency, which has been covering local stories in the various parts of the country for half a century. In its long history, ENA has been using technologies of the time to produce and store news items.

The Agency was using teleprinter and stencil to prepare and store news items for long time. During this time, ENA was organizing the news items (stencils, for example) by region. The aim was only to find out how many news items produced in each region. No classification technique, which considers the subject of news items, was utilized.

However, recently the need for a better classification scheme was felt and a new classification scheme is implemented, in parallel with the computerization project.

3.3.2 The Amharic News Classification Scheme

The agency produced the new classification scheme based on Reuter and AFP's classification scheme. However, as the interviewees[10] say, the two known news agencies utilized a scheme that is prepared for international purpose, which is a detailed one. As a result, the agency prepared a somewhat summarized scheme, which is expected to suit the national demand. The classification scheme was revised two times. The first proposed scheme had only four main classes, and it has been in use only for about a year (in the year 1991 E.C.).

The revised classification scheme has hierarchical nature, having two levels. There are seventeen main (parent) classes and all in all three hundred subclasses. Though it is not explicitly defined, some of the subclasses have some hierarchical nature, and could be further sub classified.

For example, under the main class **ጤና** ('hea' in English to mean health), there are 27 subclasses, which can be put under three categories (**ሕክአ** – 'ts' or treatment services, **ጤጥሚ** – 'moh' or Ministry of Health, and **በሽተ** – 'dis' or diseases).

[10] Ato Simeneh, and Ato Sileshi, both committee members.

Similarly, under the main class **ዓመበ** (‘ph’ – public holiday), there are 17 subclasses, which can be put under 2 subheadings, one containing 3 and the other 14 subheadings respectively. In fact, there are also classes, which do not have any trend to further subdivide. Examples of such class include, **ዓዓግ** – (‘ir’ – international relations) and **ማኅደ** – (‘sose’ – social security).

The number of subclasses in each main class is not uniform. There are main classes with three subclasses, and also with a longer list, 79 subclasses. The following table summarizes the classification scheme according to the number of subclasses. A detailed list of the classification scheme is attached in Annex 2.

No	Main Class		No. of subclasses
	Code	Description	
1	**ጤናጥ** (hea)	**ጤና ጥበቃ** (health)	27
2	**ዓመበ** (ph)	**ዓመታዊ በዓላት** (public holidays)	17
3	**ዓአግ** (ir)	**ዓለምአቀፍ ግንኙነቶች** (international relations)	10
4	**ማኅደ** (sose)	**ማኅበራዊ ደኅንነት** (social security)	5
5	**ባህለ** (cul)	**ባህል** (culture)	19
6	**ብፖለ** (na)	**ብሔራዊ ፖለቲካ** (national politics)	21
7	**ፍፍት** (saj)	**ፍርድና ፍትህ** (sentence and justice)	12
8	**ገበዜ** (mn)	**የገበያ ዜና** (market news)	23
9	**መከጸ** (das)	**መከላከያና ጸጥታ**(defense and security)	12
10	**ሣቴክ** (sat)	**ሣይንስና ቴክኖሎጂ** (science and technology)	5
11	**ጠቅል** (od)	**ጠቅላላ ልማት** (overall development)	7

No	Main Class		No. of subclasses
	Code	Description	
12	ስፖር (spo)	ስፖርት (sport)	23
13	ትምህ (edu)	ትምህርት (education)	20
14	ኢኮኖ (eco)	ኢኮኖሚ (economy)	79
15	አደም (acc)	አደጋዎች (accidents)	11
16	አየጸ (wc)	የአየር ጸባይ (weather condition)	3

Table 3.1 Main Classes of Amharic News Classification Scheme

As the classification is a first trial, it has some problems. For example, the main class **አአገ** does not show subject; rather it shows nature of news items. That is, the headings of subclasses under this class are not based on subject analysis. For example, there is a heading with code **እግድ** (blocked news) that can apply for any type of news (political or economic), not for a specific news type. However, there is also one subclass under this class that shows some concept, which is **ዜዕ** – 'news about deceased personalities'. Because of this, this class is not included in table 3.1 above.

Since the structure of documents help to decide which part to take in the process of automatic classification, the following section discusses the structure of Amharic news items.

3.4 AMHARIC NEWS STRUCTURE

The structure of ENA's news item is assumed to hold four parts, namely, a header, headline, lead paragraph, and body (ENA, 1993a).

3.4.1 Header

Every news item should have a header, which consists of:

- Classification code.
- Slug: - which is a general identification of subject in the form of a generic master slug, which may be followed by a slug specific to that story, e.g. Somalia – summit – Menawi. A slug is no substitute for a headline.
- Author's name.
- Dateline: - gives date and place of story's origin and agency's acronym.

3.4.2 Headline

Headline should give the content of a story in a few crisp words to catch the reader's interest. It does not exceed one line.

3.4.3 Lead

The lead is the opening paragraph. A lead would contain no more than 30 words. The lead captures the essence of a situation event clearly, and if possible, dramatically.

3.4.4 Body

Body also called catchall paragraphs elaborate on the lead and provides any necessary details. There can be at least six paragraphs in the body.

In fact, every part is there to elaborate the information contained in the news item and as such they may contain similar words (phrases).

It would be easier to process the headline, lead, or a combination of both in order to classify news items automatically. However, most of the news items at ENA (particularly Amharic news items) do not follow the above structure. For example, 214 news items do not have headline, 315 do not have slug, and 443 do not have keyword. The following table summarizes the structure of Amharic news items produced in the last two years.

Code	No. of news	Number of news with		
		Headline	Keyword	Slug
ጤናጥ	788	762	770	761
ዓመበ	142	142	141	140
ዓአግ	299	288	280	271
ማኅደ	1546	1421	1278	1400
ባህለ	145	140	135	135
ብፓለ	674	665	638	652
ፍፍት	488	487	482	481
ገበዜ	157	157	156	156
መከጸ	70	70	70	70
ሣቴክ	90	88	84	84
ጠቅል	332	330	324	323
ስፖር	340	332	327	328
ትምህ	454	452	435	441
ኢኮኖ	1084	1063	1050	1054
አደዎ	271	270	268	270
አየጸ	24	23	23	23

Table 3.2 Structure of Amharic News Items

On the other hand, from the 11,484 Amharic news items, 2,868 news items are stored along with the classification codes that the agency was using before refining the

classification scheme and 1,712 news items do not have classification code. The news items in the above table 3.2 are those news items with the new classification code.

After having a detailed review of the news items, it was found out that some language processing aspects should be made to assist the statistical techniques in order to come up with a "good" classifier. In fact, as the classification scheme is also implemented in Amharic, it is necessary to discuss issues related to Amharic Language and its scripts. The following section discusses these issues.

3.5 THE AMHARIC WRITING SYSTEM

3.5.1 The Amharic Characters

As mentioned previously, Amharic is the official language of Ethiopia. The present writing system of Amharic is taken from Ge'ez alphabet, which was the language of literature in Ethiopia in the early time. The Amharic writing system consists of a core of 33 characters each of which occurs in a basic form and in six other forms known as orders. Each graphic symbol represents a consonant together with its vowel. The vocalic symbol cannot be detached from the consonant element. That is, Amharic does not use independent symbols for vowels. In other words, as Leslau (1965) discusses, the Amharic script is a syllabic rather than an alphabet.

The seven orders represent the different forms of a consonant. Each form is made in accordance with the sound that goes with the symbol. The non-basic forms are derived from the basic forms by more-or-less regular modifications. For example, using the consonants ሀ (hä), ለ (lä) and መ (mä) the orders are;

1st	2nd	3rd	4th	5th	6th	7th
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
hä	hu	hi	ha	he	h	ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
lä	lu	li	la	le	l	lo
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
mä	mu	mi	ma	me	m	mo

In addition to these, there are four so – called labio–vellars, which have five orders (e.g. ቈ ቍ ቊ ቋ ቌ), and eighteen additional labialized consonants (e.g. ኳ ሟ ሯ ሷ ...). The language has also its own numeral symbols (without a symbol for 0), but not mostly used except for writing dates. A listing of the whole Amharic character set, also called fidel (**ፊደል**) is attached in Annex 3.

The Amharic alphabet, all in all has around 290 letters. The alphabet does not have any distinction between capital and lower case letters.

3.5.2 Punctuation Marks

The individual words in a sentence are separated by two dots '፡', although this practice is not exercised especially in type-written texts. The end of a sentence is marked by a square-formed four dots '።'. The symbol '፣ ' represents a comma, while '፤' corresponded to a semicolon. Apart from this, the language has borrowed some punctuation marks from foreign languages. For example, '!' and '?' are used in the language, which are borrowed from foreign languages.

Though not used by the general public, there are about 17 punctuation marks (Beletu, 1982). In fact, existing Amharic software do not also implement some of them.

3.5.3 Problems In The Amharic Writing System

There are many problems observed regarding the writing system of Amharic language. Some of them are summarized below.

3.5.3.1. Problems regarding consonants with different form

As discussed above, Amharic borrowed most of its scripts from Geez. However, it did not select from Geez alphabet those symbols that are only necessary for its consonants. As a result, there are certain phonemes with different symbols, where they have meaning in Geez, but their meaning is not known in Amharic. The following table shows the list of consonants that have different forms.

Consonant	Other forms of the Consonant
ሀ (hä)	ሐ and ኀ
ሠ (sä)	ሰ
ኧ (ä)	ዐ
ጸ (tsä)	ፀ

Table 3.3 Amharic Alphabets with Different Forms

The pronunciation of these paired characters is the same and each of them has their own orders.

The distinction between these symbols in Geez is when spelling certain words, however there is no rule as for their usage in Amharic language, and as Getachew (1967) says, the proper use of these symbols is not studied exhaustively and also there is no standard dictionary to refer to. Therefore, it is not clear whether one should write

“häyl” (power) as **ሀይል**, **ሐይል** or **ኀይል**, also “tsähäy” (sun) as **ፀሐይ** or **ጸሐይ**. As a result there arises some confusion and inconsistencies in Amharic spelling, and these redundant consonants are assumed surplus.

3.5.3.2. Problems related to certain interchangeably used orders

In a similar research, Beletu (1982) mentions the confusion regarding the first order and the fourth order of some consonants. For instance, it is not clear which one to choose **ሀ** **(**as **ሀይሉ)** or **ሃ** **(**as **ሃይሉ)** to spell ‘Häylu’ – name of a person. As a result, one can find the same word “Häylu”, spelled differently in six forms, which are **ሀይሉ**, **ሃይሉ**, **ሐይሉ**, **ሓይሉ**, **ኀይሉ**, and **ኃይሉ**. Similarly, “eye” can have three forms, which are **ዐይን**, **ዓይን**, and **አይን**, with pronunciation “äyn”.

It has also been found that the second order of the consonant **ወ**, which is **ዉ**, and its sixth order, which is **ው** are interchangeably used and there is no consistency. Because of this, one can find the word ‘dog’ spelled as **ዉሻ** and **ውሻ** (“wshä” and “wushä” respectively).

3.5.3.3. Problem regarding compound words

In another research, Bender and Ferguson (1964) has mentioned another problem regarding the division of compound words. For example, it is not clear which one, **ወጥቤት** “wät’bet” or **ወጥ ቤት** “wät’ bet” is the correct spelling for ‘kitchen’. They also mentioned another problem of the writing system, which has something to do with regularizing spellings and regularizing punctuation. For example, the word “sämtoäl” (‘he has heard’) may be spelled as **ሰምቶአል**, **ሰምቷል** or **ሰምትዋል**. This problem

exists in different languages that have words of different forms of writing. For example, the words ‘recognize’ and ‘recognise’ in English language are two variants of the same word.

Translation from foreign words into Amharic is also another problem mentioned by Getachew (1967). As a result, one can find the word ‘television’ translated into different forms, which include: **ቴሌቭዢን**, **ቴሌቭዥን**, **ቴሌቪዥን**.

Therefore, any automatic Amharic text processing should consider the aforementioned problems.

3.5.3.4. Problem regarding abbreviations

As seen from documents reviewed, there is also no consistency when spelling abbreviations. For example, when abbreviating the phrase **ዓመተ ምህረት** (in the year AD), one can find **ዓ.ም**., **ዓ.ም** or **ዓም** as possible abbreviations. So these kinds of words should come into a common word. Similarly, the use of hyphen is also not consistent.

The problems mentioned under 3.5.3.2 and 3.5.3.3 were not considered in this research because of time constraint. However, the researcher assumed the statistical techniques would not miss the variant words created due to these problems. The reason is that, a journalist who used a specific variant of a word to spell the word continues to do so, and the classifier will identify the word because of its frequency.

3.6 AMHARIC STEMMER

In the literature, there is a mention of an algorithm for Amharic stemmer developed by Nega (Abiyot, 2000). However, despite the effort made, the algorithm could not be found. As a result, a simple depluralization and suffix and prefix removal technique is used.

It is obvious that nouns are most important for indexing. Therefore, review of the suffixes and prefixes for nouns was made and the following section discusses them. Some of these prefixes and suffixes can be also applied to other language parts. In fact, verbs are equally important index terms in the case of news items and because of this they are also included in the selection of index terms.

3.6.1. Suffixes Considered

The plural form of a noun in Amharic is expressed by attaching the suffix **ዎች** (woch) to singular nouns. There are also different suffixes that can be attached to nouns. The most common are **ን**, **ም**, and **ና** (Abiyot, 2000). Their attachment is made to show possession (**ን**), emphasis (**ም**), and object marker (**ና**).

A noun can take one or more of these suffixes in a number of ways. The table 3.4 below presents an example of the possible combination of suffixes to nouns created from **ዎች** and the above three suffixes. In general, all in all 70 suffixes are used in this research. They are presented in Annex 4.

Suffixes Combined	Suffix used
ዎች + ን	ዎችን
ዎች + ን + ና	ዎችንና
ዎች + ን + ም	ዎችንም
ዎች + ና	ዎችና
ዎች + ና + ም	ዎችናም
ዎች + ና + ን	ዎችናን
ዎች + ም	ዎችም
ዎች + ም + ና	ዎችምና

Table 3.4 Example of the Possible Combination of Suffixes from '**ዎች**' and '**ን**', '**ም**', and '**ና**'.

3.6.2 Prefixes Considered

Nouns take four prefixes, which are also taken by verbs (Abiyot, 2000). These are **በ**, **ለ**, **ከ**, **የ**. In addition there are two prefixes identified after reviewing the news documents, which are attached to verbs. They are **ስለ** and **እንደ**.

3.7 AMHARIC SOFTWARE

Since Amharic is not known in the ASCII code table, various software experts have tried to develop their own keyboard driver program, which converts the English keyboard into Amharic keyboard. Technically speaking, the software convert the default code table where each key is associated with English symbols by Amharic code table, so that users can use the same keyboard to write and edit Amharic letters. In other words, these programs associate the keyboard buttons with Amharic symbols.

In fact, this is done at the screen level. That is, the symbols stored inside files are the associated ASCII symbols of the default code table, not the Amharic symbols. The software converts these symbols into associated Amharic symbols when they are read

from files into memory. As a result users see Amharic symbols on their screen and when printing.

Ever since 1987, there have been different software developed to assist users to write and edit Amharic text inside the computer. Most of the software are written to work only with Microsoft Word. However, there are few which can work in other programs, one of them being Visual Geez, which currently becomes very popular. The software is developed by Custor Computing Pvt. Co. The software has two versions, VG2 and VG2000 developed for the various versions of Microsoft Office products. ENA is using the first version, which is VG2.

All Amharic software have succeeded in helping users to enter and edit Amharic text inside the computer (especially for word processing purpose). However, all face some problems especially when they are used for database purpose.

The problem with all Amharic software is when sorting data; the result will be different from what one anticipates. The problem is due to the large number of Amharic symbols to represent inside the computer. That is, the number of the Amharic characters is beyond the number of ASCII codes and the available buttons on the keyboard. As a result, all Amharic software implement the assignment of Amharic symbols (especially the non-basic forms) as two characters. For example, the fidel "ሁ", for example, is stored as two characters "ሀ" and the diactric mark " ◌ሁ " inside the computer. The codes assigned to the characters did not also consider their precedence in the language. Because of this, for example, when we sort the data (ሀገር, ሂሣብ, ህዝብ), we will find ሂሣብ (not ሀገር) first. Therefore, any text processing should

consider this as well. That is, consider these characters as two symbols, not as is seen outside (one).

On the other hand, when the software implement the representation of symbols borrowed from other languages, most of them have changed the codes of these symbols. For example, the ASCII code for '?' (question mark) is 63 in the default code page. However, Visual Geez version 2.0 has implemented it by giving code 41.

Therefore, any string searching and matching procedure should consider the codes, as implemented in each software, not the symbol when searching for these symbols. In other words, the search should not be done, for example by using "?" to search question mark symbol as it is done in any string searching program.

As mentioned above, when searching for Amharic symbols we can use their equivalent in the default code page. For example, the code for '**ፉ**' in Visual Geez is the code for '/' in the default code table. This means, '**ፉ**' is stored inside files as '/'. Therefore, to search for '**ፉ**' one can use either its ASCII code or use "/" in the search statement.

CHAPTER FOUR

DESIGN OF THE AUTOMATIC CLASSIFIER

4.1 INTRODUCTION

In this chapter the techniques used to develop ANC (Amharic News Classifier) that classifies Amharic news items into predefined classes are discussed. The implementation and testing of the techniques is also presented. The overall work presented in this chapter is based on the techniques reviewed in chapter two. The important aspects of Amharic language, which are discussed in chapter three and that should be considered in any text processing system, are also considered in the development process.

The proposed system is designed using vector space model, which as discussed in chapter two has been reported to perform better in various researches.

The design consideration of the system is split into two phases. The first phase is aimed to produce the vector table, which is a collection of the class vectors. The experimental group of news items was processed to derive these class vectors.

In the second phase, the validation group of news items, which had no relation to the construction of the class vectors, are classified and the percentage of correct assignments is recorded.

In both phases document analysis was the key aspect. That is to say, after preprocessing, documents are analyzed and document vectors generated. The procedure followed in designing ANC is shown in figure 4.1 below.

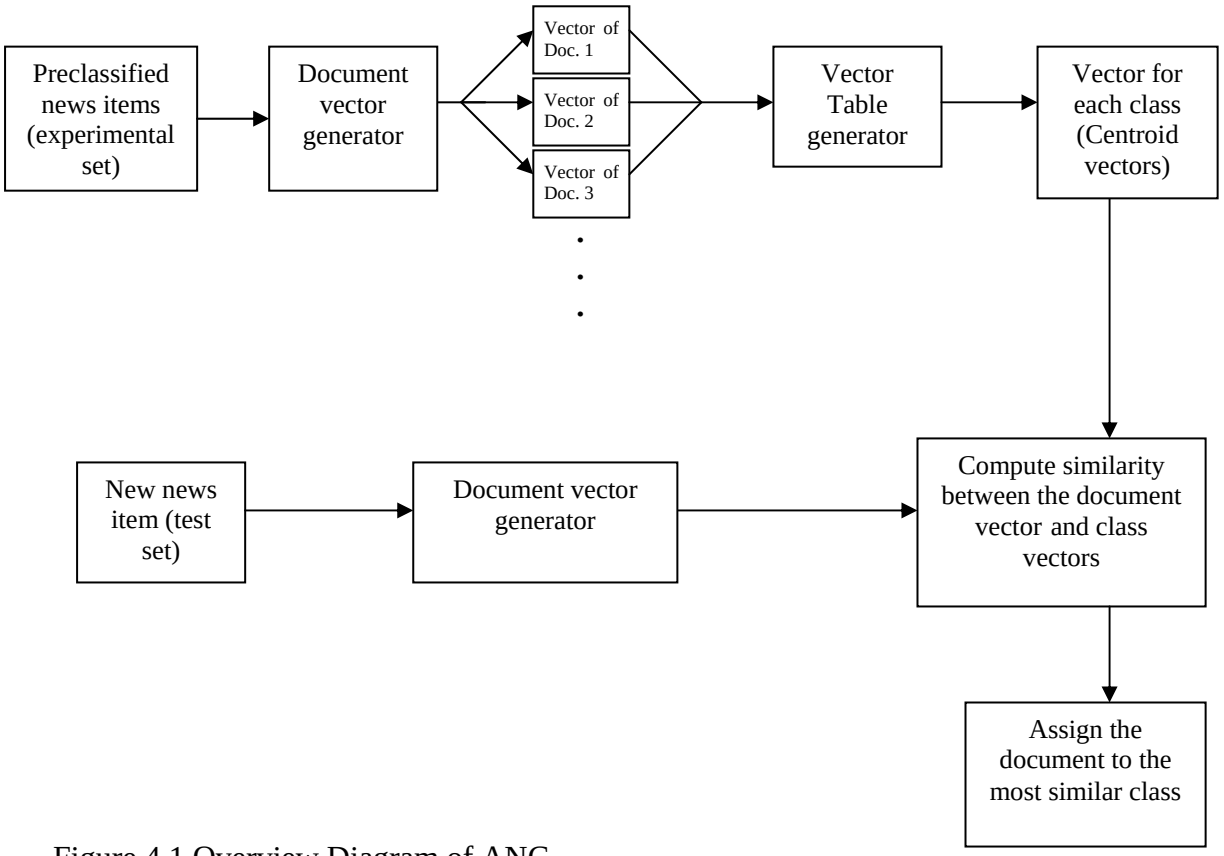


Figure 4.1 Overview Diagram of ANC

4.2 NEWS FOR EXPERIMENTATION

The news collection under this research comprises of one-year Amharic news items of three classes, namely 'acc', 'culture', and 'eco'[11]. All in all the news items used in the experiment constitute 1,481 news items. As mentioned previously, this data set is divided into two groups: training set and test set.

The training set contains the news items produced during the time Jan-Aug/2000. The test set includes all news items in the 3 classes produced during Sep-Dec/2000.

[11] In this chapter, English codes are used for the classes that are presented in table 3.1 in parenthesis.

Fifty news items were randomly taken from each class to verify the correctness of the manually assigned codes by journalists. An expert from the agency checked the class codes for correctness and only 15 incorrect assignments were identified.

4.3 PREPROCESSING

As described in section 1.2.2, the data set (all Amharic news items) are stored in different folders by the news management software. News items produced by City Desk are found in DocA folder, and that produced by Region Desk are put in DocR folder, both under ENASoft folder.

As the news management software that the agency is using is new, there are files created erroneously and also for testing purpose stored inside the folders mentioned above. On the other hand, as mentioned in section 3.4 any news item should have on average three to five paragraphs.

Consequently, to discover files that are erroneously created, 50 files with five or more paragraphs long were identified and the average size was recorded. The average size of these files was found to be 500 bytes excluding formatting characters and file header information[12].

In order to identify key terms of a particular class, we must have the news items already belonging to that class, and look for the frequency of occurrences of their terms. Therefore, to bring the news items that belong to a given class together, a program was written that puts them into their respective folders that are created for this purpose. Accordingly, all news items classified with 'acc' classification code were put

[12] the file size we get when we use directory listing includes file header information and formatting characters.

in a folder named ‘acc’, those with ‘culture’ classification code into a folder named ‘culture’, and with ‘eco’ classification code into a folder named ‘eco’.

Finally, each file in these directories was read and the length was checked against this figure (500 bytes) and all files below this size are removed from the two folders (DocA and DocR).

The data in each class is then divided into two exclusive groups: the experimental and test group. The number of news items in experimental and validation group of the three classes are presented in the following table.

Class	Number of news items in the experimental group	Number of news items in the validation group
Accident	210	59
Culture	100	41
Economy	850	221

Table 4.1 Number of News Items in Experimental and Validation Group

4.4 WORD IDENTIFICATION

Since every news item is stored in a file with rich text format (rtf), many formatting characters are stored along with the text of the news inside the files. As a result, simple file processing technique would lead to construction of noisy data. For this purpose, before processing begun the data in the files is loaded into an rtf control, which can consume the formatting characters, and each word can be obtained without these characters.

Unlike English text, Amharic text uses '#' to separate Amharic words. However, as mentioned previously this practice is not used much currently, especially for computer-based texts and instead space is used. In reality, a word is separated not only with space or '#', but also with other delimiters identified during the development process.

In general, after analyzing the documents it was found that an Amharic word can be identified using the following delimiters: #, ።, ፣, ፤, ", /, space, tab, carriage return and line feed characters. However, as discussed in section 3.7, most Amharic software that developed their code table, shifted the ASCII codes of most symbols borrowed from English, like "/", "(", ")", "!" etc. to a different code.

Therefore, the ASCII codes (as implemented in the software) are used for these symbols in the process of word identification. After analysing Amharic news items, the following word identification algorithm is developed and used for Amharic news items[13].

1. Initialize the variable to hold the word

2. Read a character from the sentence (document)

3. Check if the character is Amharic word delimiter

4. If not, concatenate the character to the variable,

[13] Actually, it can be used for any Amharic text.

5. Else if the character length is above one character report the word[14]

6. If there is more data to process, go to step 1

The implementation of this algorithm is presented in Annex 5 (Function GenerateWord). In fact, this function also identifies unique words and produces a frequency count of each word from the document supplied.

4.5 CHECKING THE VALIDITY OF A WORD

After the word is identified, a simple checking was done to verify whether the word identifed is a valid Amharic word or not. In this reseach a word is considered invalid if it has any numeric character in it. In other word, the checking is done to verify if no numeric character exists inside the word. For example, the word '**የ**1992**ቱ**' is not a valid Amharic word. If the word is not a valid word, it is discarded.

4.6 CHANGING CHARACTERS TO THEIR COMMON FORM

As mentioned in section 3.5.3.1, some characters have different forms. There are also characters that have some orders that are used interchangeably. Therefore, after a word is identified these characters were searched and replaced to bring to common forms of words created with different spelling. For example, in the example given in section 3.5.3.2, the different forms of the word 'Hailu', which are **ሀይሉ**, **ሃይሉ**, **ሐይሉ**, and **ኀይሉ** are all converted to the first form **ሀይሉ** by changing the first character of the last three words by its first form.

[14] There are few words with one character length. Examples include **ና** or 'come' (used for male), and **ኑ** or 'come' (used for many individuals). However, they are not important to be taken as index terms.

In most cases, we get the non-basic orders of consonants by attaching a diactric mark to their first order. As a result, Amharic software mostly implement the non-basic orders as two characters. In fact, the diactric ‘ُ’, which is applied to **ሀ** to make it **ሁ**, is also applied to **ለ** to make it **ሉ**. Similarly, the diactric ‘ِ’, which is applied to **ሐ** to make it **ሒ**, is also applied to **ቨ** to make it **ቪ**. Therefore, when changing characters to common form, it is only enough to change the base character. For example, to change **ሐኩ** to **ሁ**, we only have to change **ሐ** to **ሀ** and it will become **ሁ**.

Therefore, to accomplish this task, the different forms as implemented in the software are analyzed and only those to be changed were identified and changed accordingly. To manage the task easily, one function for each character is written. Additionally, a change was made to some non-basic forms that are used interchangeably, which are discussed in section 3.5.3.2. The functions of the prototype are presented in Annex 5.

Finally, as mentioned in section 3.5.3.4, the use of dot (‘.’) and hyphen (‘-‘) is not uniform. Therefore, these symbols were removed from the identified words to make them common.

4.7 WORD STEMMING

The use of stemming is to reduce redundancy. For example, stemming will bring the different forms of the word **ውሻ** or ‘dog‘ (**ውሻው**, **ውሻም**, **ውሻን**, **ውሻና**, **ውሻዎች**, **ውሻዎችን** etc.) into their root word **ውሻ**.

Stemming a word is by itself one big task. And, as cited by Abiyot (2000), Nega has developed a stemming algorithm for Amharic language. However, despite the effort

made, the research could not be found. Therefore, a simple depluralization, and suffix and prefix stripping technique is applied to the words identified in the above process. The list of suffixes applied is presented in Annex 4.

As can be seen from the discussion made in section 3.6.1 and Annex 3, the possible number of suffix size (in terms of character length) is not more than five characteres long[15]. Therefore, the possible suffixes were stripped from each word and put in different variables, and the existence of the suffixes in the suffix database was checked starting from the longest suffix down to the shortest suffix. The procedure used to strip suffix from a word is as follows:

- Check if the maximum length suffix exists in the word by stripping the right most five characters. That is, if the length of the word is greater than six[16], then strip the right most five characters and check if it exist in the suffix database. If the suffix exists then report the word without the suffix.

- Else, if the length of the word is greater than five, then strip the right most four characters and check if it exists in the suffix database. If the suffix exists then report the word without the suffix.

- Else, if the length of the word is greater than four, then strip the right most four characters and check if it exists in the suffix database. If the suffix exists then report the word without the suffix.

[15] In other words, the possible suffix length is between 1 and 5 characters long
[16] Length of the word and/or remaining string is checked in two ways. That is, if there is a diactric in the remaining string, the length required will be increased by one from that if a diactric is not exist. Therefore, the length required would be seven for this case.

- Else, if the length of the word is greater than four, then strip the right most four characters and check if it exists in the suffix database. If the suffix exists then report the word without the suffix.

- Else, if the length of the word is greater than three, then strip the right most four characters and check if it exists in the suffix database. If the suffix exists then report the word without the suffix.

- Else, if the length of the word is greater than two, then strip the right most four characters and check if it exists in the suffix database. If the suffix exists then report the word without the suffix.

As mentioned in section 3.6.2, the prefixes considered in this research are: **ከ**, **በ**, **የ**, **ለ**, **ስለ** and **እንደ**. Prefix removal is done in two steps. One character length prefixes are checked first and removed; and then the two other prefixes are checked. The algorithm developed to remove a prefix from a word is as follows:

- If length of the word is greater than two and if the first character is **ከ**, **በ**, **የ**, and **ለ**, then check if the second character is not a diactric to these characters[17].

- If it is not, then check also if there is a diactric in the remaining string (which will affect the length to be considered). That is, if there is a diactric in the remaining string the length should be greater than two; else the length should be greater than one.

- If the length satisfies the requirement then strip the prefix and report the word.

[17] The first character is **ከ**, **በ**, **የ**, **ለ** does not necessarily mean it is a prefix. The character may be **ኩ**, **ቡ**, **ሉ**, **ዩ** etc. In other words, the 2nd character determines whether the first character is a prefix or not.

The same technique is applied for prefixes ‘ስለ‘ and ‘እንደ‘.

The stemming process was successful in reducing the list of words identified. It reduced the word list of the training set by 18% percent.

4.8 STOP WORD REMOVAL

After the word is stemmed using the procedures discussed in section 4.7, the existence of the word in the stop word list is checked. In fact, the stop words are also stored in stemmed form. In this thesis two kinds of stop words were identified: news specific stop words and common stop words.

As discussed in chapter three, stop words are words that occur frequently in almost all documents. Using this concept, all high frequency words that exist in all documents were retrieved and reviewed to identify the stop words of both types.

4.8.1 News Specific Stop Words

Reporters and journalists most of the time report an incident to the public. As a result, they use vocabularies peculiar for this purpose. An example of such words, which they use very frequently, is 'notify' (**አስታወቀ**, **አስታወቁ**, **አስታውቀዋል**). Basically, they use this word when reporting about an official or organizational press release. In fact, these words are pure verbs and are usually found at the end of a sentence[18]. Therefore, the identification of these words was done in two steps:

- List of high frequency terms from all classes were selected.

[18] The structure of Amharic language differs from English in such a way that while the structure of English is Subject-Verb-Object (SVO) that of Amharic is Subject-Object-Verb (SOV). This means, usually verbs come at the end of a sentence in Amharic sentences.

- Then, common terms in all the classes were selected, and the result was reviewed manually.

The assessment was made with an expert from the Agency.

4.8.2 Common Stop Words

Like other languages, some words in Amharic are used very frequently in the normal usage of the language. A list of 100 documents (audit reports) was processed using the above technique to identify these function words.

In general, the two techniques mentioned above produced a total of 750 stop words, and the implementation of this step reduced the feature size on average by 20%.

4.9 VECTOR TABLE GENERATION

As described in chapter three, centroid vectors are generated by averaging the document vector belonging to a certain class. Therefore, to identify the centroid vectors of the three classes, all documents identified as training set were processed and their vectors stored in a table. That is, the first step in vector table generation is document vector identification. This was done in three steps.

- Identify high frequency words of each document in the training set.
- Calculate document frequency for each of the high frequency words.
- Calculate weight only for terms with document frequency more than one[19]. The weight is calculated using equation 2.1 presented in section 2.5.3.

[19] For document frequency, a threshold of 2 was taken in this experiment.

After calculating the weight of each term, then documents are represented by a vector using a table as discussed in equation 2.2 of section 2.5.3. The following table, for example, presents the vector of a document with Id 38215 from economy class.

Word	Weight
ጨረታ	22.82367
ባንኩ	16.64913
ብር	15.29998
ዶላር	21.27727
መመንዘሩ	19.5193
ሳንቲ	23.44123
አሜሪካ	21.53831

Table 4.2 Vector Representation of Document No. 38215.

In the above table, the other terms in the vector space have zero weight value. As a result, if we assume the above terms are consecutive in the vector space and come after the third term, we can have the following vector form for the document under discussion.

Document 38215= (0,0,0,22.82,16.65,15.30,21.28,19.52,23.44,21.54,0,0,...)

Once the document vectors are identified using the above step, then to identify the centroid vector of each class a table was designed with fields: Keywords, WeightInClassA, WeightInClassC, and WeightInClassE. Only unique words from the total words identified during feature selection were then taken and inserted into the first column of the vector table.

The weight of each word is then calculated using the formula (equation 2.3) presented in section 2.5.4. That is, for each word, the weights in the different documents of a given class are summed and divided by the number of documents in that class and the result entered in the appropriate column (WeightInClassA, WeightInClassC, WeightInClassE).

The following table shows the vector table developed with its first 10 entries. A sample of the vector table is attached in Annex 6[20].

Keyword	WeightInClassA	WeightInClassC	WeightInClassE
ጤንዚዛ	0.4444896	0	0.5778365
ጤንዝዛ	0	0	0.2014329
ጤስ	0	0	0.5984692
ጤና	0.2823364	0.2823364	4.235047
ጤናማ	0.3111427	0	0
ጤፍ	0	0	2.313637
ጤንነቱ	0	0	0.0968913
ጸድቀ	0	0	9.20721562
ጸሀፊ	0	0.1245869	8.305793502

Table 4.3 Vector Table of ANC

In general, the lists of keywords produced are assumed the most distinguishing terms for each class. The following table displays the top ten discriminating words of the three classes.

[20] Because the number of terms is very large, a sample of the vector table is assumed enough.

No	Class Accident		Class Culture		Class Economy	
	Word	Weight	Word	Weight	Word	Weight
1	አደጋ	12.40586	ጦር	2.435237	ስራ	32.33935
2	ፖሊስ	6.806531	ቅርሶች	1.947207	ብር	23.2146
3	መኪ	3.573478	ስራ	1.620441	ዉሀ	18.40815
4	እርዳታ	3.265789	ምኒልክ	1.601014	ግብር	18.3733
5	ጉዳት	2.725931	ስራ	1.599758	ልማት	17.05214
6	እሳት	2.619697	ባህል	1.469573	አርሶ	13.77669
7	ትራፊክ	2.551131	ጣሊያ	1.399788	ገበያ	10.75025
8	ቃጠሎ	2.525168	መካከል	1.270231	ምርት	10.53446
9	እህል	2.499818	ኢጣሊያ	1.215289	አገልግሎት	10.14114
10	መከላከል	2.368053	ስነ	1.138181	ቡና	9.382524

Table 4.4 The Top Ten Distinguishing Words of the Three Classes.

The total number of keywords identified reached 3,326. The following table shows the number of keywords in each class.

Class	Number of Keywords
Accident	914
Culture	943
Economy	2585

Table 4.5 Number of Keywords in Each Class

In fact, each class has some common keywords with the other classes. The following table shows the number of common terms between the classes.

	Culture	Economy
Accident	291	561
Culture	—	511

Table 4.6 Number of Common Terms Between Classes

4.10 AUTOMATIC CLASSIFICATION

To classify new document, first the document vector is produced using the same procedure mentioned in section 4.9 above. Here again, only words with document frequency greater than one are considered when producing the vector of the document to be classified.

As the formula to calculate weight of terms needs a count of total documents in the collection and document frequency for the term, this information is required to be stored permanently. Therefore, to easily calculate *df*, each term is stored along with this information in a table. On the other hand, to hold a running count of total documents in the collection, a table (Number of New Documents table) is used and each time a new document is processed, the value in this table is incremented.

Once the document vector is identified, then the matching of the vector with class vectors is done. The technique used to determine the class to which the document belongs is as follows:

- A variable is declared for each class to hold the similarity value between the document and that class.
- The similarity between the document and each class is then calculated using formula 2.4 discussed in section 2.6.5.1 and stored in the variables declared above.

- When the similarity computation is complete the maximum similarity value from the three is assigned to the document using formula 2.5 of section 2.5.5.1.

The implementation of this procedure is presented in Annex 4 (Function Classify). The following flowchart shows the steps followed by ANC to classify new document.

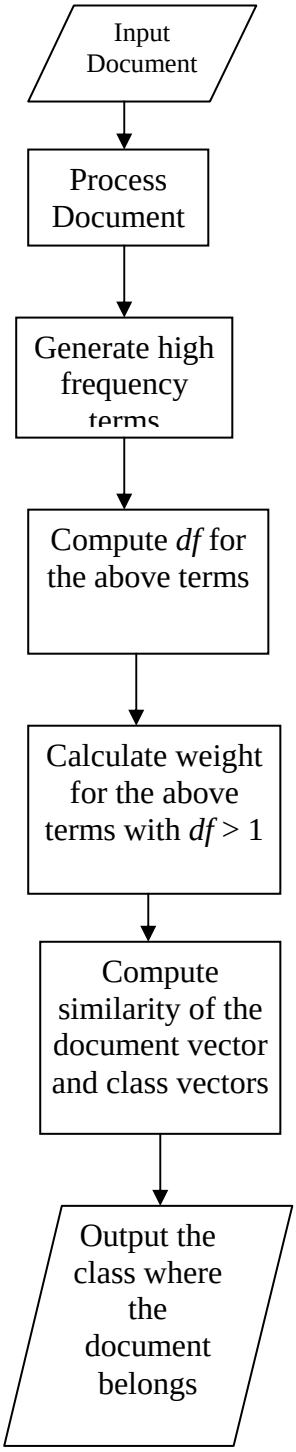


Figure 4.3 Flowchart of ANC

4.11 THE PROTOTYPE

The news item to be classified can be given to the classifier either by typing the text directly on the space provided (the Rich Text Box) or by selecting the file which contains the news item. To select the file that contains the news item, the following procedure needs to be followed.

- If the drive is different it should be changed by using the drive list box.
- If the drive is correct and the directory where the file is stored is different, then the directory should be changed.
- Finally, the file can be selected from the file list. The file control is set to display only files with “rtf” and “doc” extension.

The following figure shows the main screen of ANC.

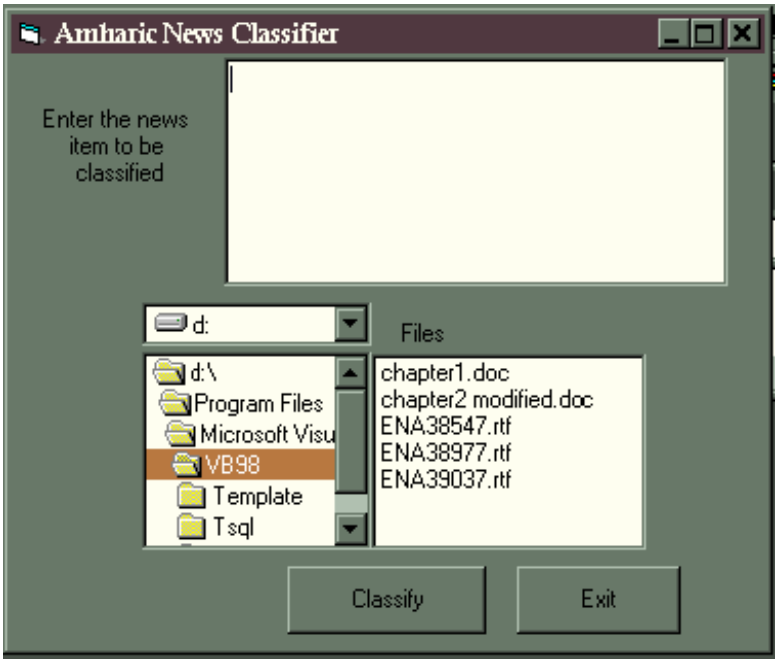


Figure 4.2 Main Screen of ANC

When the classify button is pressed, checking is done whether the news item is typed in or a file is chosen. If both options are not selected, a message is displayed and the program returns to the main screen. Otherwise, depending on the option chosen, the classification is done and the result, i.e. the class where the news item belongs, will be displayed. The program waits for another news item to classify, untill exit is pressed.

4.12 TESTING

Testing is done using the documents in the test set. That is, each document in the test set was classified using the above procedure and the result was recorded. Then, the result was compared with the class code assigned manually. Using the result, percentage was computed to see the effectiveness of the classifier.

The following table shows the test result and percentage accuracy for each class.

Class	Number of documents		Accuracy (%)
	Tested	Correctly Classified	
Accident	59	50	84.7
Culture	41	38	92.68
Economy	221	208	94.12

Table 4.7 Accuracy in Each Class

From the above table, it was found out that the classifier gave correct code on the average for 90.5% of the test documents. That is, from the 321 documents 296 of them are assigned ‘correct’ code by ANC. Since the test set is assumed reasonably representative, the system can be said successful. However, to improve the system, it

should be supported by other preprocessing techniques such as thesaurus and stemming.

The results are further analyzed and the following observations are made that needs additional considerations to enhance the developed Amharic news classification system.

4.13 DISCUSSION

All the 25 news items that are not classified correctly by the system were reviewed. The review showed that 6 items were recorded as incorrectly classified because of the wrong code given to them by journalists. That is, the code given by ANC for these items was correct. This would increase the percentage accuracy of ANC.

In addition, from table 4.7, it can be seen that larger number of news items that belong to the class economy are classified correctly. This shows the class has a clear concept that distinguishes it from other classses. In fact, the weight of the first 10 terms of the three classes show that, economy has many discriminating terms, while culture does not have much (see table 4.4. for detail). On the other hand, the news items in the culture class are not clear in regard to the concept they present, but the result was good. The reason for the high percentage found in this class might be the size of the test data taken from this class. In other words, it implies larger test data is required to confirm the performance of automatic classifiers. In general, this may be an indication that some of the classes might not be formulated clearly.

Examining the list of keywords, many interesting entries were found. The lack of a complete stemming algorithm manifests itself, for instance, with the occurrence of

both **ጥንታዊ** and **ጥንታዊት** words (two forms of the word ‘historical’) in the class ‘culture’, or because of the words **ፓኬጅ** and **ፓኬጆች** (two forms of the word ‘package’) in the class ‘economy’. It was also observed that the problems mentioned in section 3.5.3.3 exist in Amharic news items. For example, a city at North (Debremarkos) was spelled in two different forms, which are **ደብረማርቆስ** and **ደብረ ማርቆስ**. Also, the word ‘president’ was translated as **ፕሬዚዳንት** and **ፕሬዝዳንት**.

Because the number of keywords was few for some of the documents, the threshold considered made the feature size bigger. Therefore, the threshold for term frequency, which is the minimum frequency a term should have to qualify as document representative, should be checked for different values and the best threshold that will decrease the feature size should be taken. The minimum value of document frequency (*df*) that a term should have in order to be included in the document vector should also be reconsidered.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

To facilitate the work of news agencies, where the use of information storage and retrieval is very frequent and urgent, possession of an automated system that will assist journalists to produce timely and accurate news seems very critical.

As a major work in information storage and retrieval, research studies in the area of automating classification have started long ago. The results of these research studies have showed that classification can be automated and good results could be obtained. The result of this research also showed that the use of automatic classification techniques for Amharic texts is possible and very promising. The research has tried to look into the techniques of automatic classification and tested some of them that seem good for the problem at hand. The development of Amharic News Classifier (ANC), which would act as an intelligent assistant to journalists, could also potentially help other users to cope with large volumes of information handled under their control with some modifications. The overall result seems to support the following conclusions.

In regard to the language aspect, it has become clear that Amharic text processing is highly dependent on the kind of Amharic software used to produce the text. Currently, there are many Amharic software in the market, but none of them seems to understand each other. In other word, every software uses its own ASCII assignment for Amharic

symbols, and as a result any text processing activity done for one software will be difficult to implement for another software.

In addition to this, different people spell some Amharic words differently. This shows the need for the development of a standard dictionary for the language. As Getachew (1967) says, we are never taught how to spell Amharic words. Therefore, it can be concluded that no emphasis is given even now to train students on how to spell at least confusing words in school.

In relation to stemming, the result showed that stemming would enhance the result by bringing variants of a given word into a common word thereby reducing the feature size. It was also observed that if documents were represented using words and phrases, not simply by words alone, the feature size would decrease and the system would be effective and efficient.

Regarding the techniques of automatic classification, the statistical method is used for Amharic text analysis and a good result was obtained. However, the statistical technique needs to be supported with preprocessing techniques, like stop word removal and stemming, which have a great contribution in document analysis.

On the other hand, the weighting technique used, *tfXidf*, seems a good choice than other weighting techniques, like Boolean weighting. This is true, especially for the current system where full stemming is not used and exact term matching would be difficult. In fact, it was found out that all the three classes share many terms in common (including the top 10 terms of the three classes that are presented in table 4.1) and it would be difficult, for example, if Boolean weighting was used.

From the experiment it was clear that variants of some words existed in the vector. This increases the feature size and also decreases the discrimination power of words.

It was also learnt that the development of stop word list highly depends on the subject matter. That is, in addition to common stop words that will be used in any domain, there are high frequency words used in the specific subject area that exist evenly throughout the entire collection. These words should also be taken as stop words. Otherwise, they would deter the result.

In any case, the feature size in automatic classification systems will be large enough, and the processing of the vector table and document will take time. This shows the need for a high capacity computer with at least 64MB of memory to be in place, if the system developed is to function efficiently.

5.2 RECOMMENDATIONS

The results found in this research showed that text analysis and automatic classification can be done automatically for Amharic texts. However, it is also learnt that more research and developmental effort need to be conducted so as to enable the full exploitation of this technology.

In particular, the following areas are identified as deserving further research work:

1. The system developed is a prototype system. This means, it has to be tested and validated for the other classes not considered in the development process. In addition, the manual classification system, as stated in chapter three, has hierarchical nature. However, because of time constraint this aspect is not taken

into consideration in the development of ANC. Therefore, the system should be enhanced to consider the hierarchical nature of the classification scheme.

2. Many preliminary works in Amharic language have to be done as prerequisites for any research on Amharic text processing. Otherwise, it would be very difficult for researchers to concentrate only on specific problems, like automatic indexing and automatic classification. In this regard, at least the following systems should be in place:

    a. Amharic spell checker. As can be seen in many documents, Amharic documents suffer from spelling errors. In reality spelling errors will deteriorate the performance of text processing systems.

    b. Stop word list. The stop lists used in this research are mainly terms of journalists. As a result, they may not be helpful in other areas or domains. Therefore, an exhaustive stop list, which can be used in any area, should be developed.

    c. Thesaurus file. As there are many variants of words in Amharic, a thesaurus file would help in reducing the features identified by bringing variants of the same word into a one word. This will increase the discrimination power of terms, as they will have high frequency when they come into a common form.

3. As discussed in chapter 3, ENA has implemented the classification scheme both in Amharic and English. However, this research work attempted to automate only the

Amharic classification scheme. Therefore, another research should be done to classify English news items automatically.

4. Because of the different types of Amharic software, there is a problem to apply Amharic text processing results for different texts. This shows the need for standardizing Amharic software.

5. The system developed does not update the class vectors to incorporate new terms. Therefore, it has to be enhanced to have a capability of adding new terms automatically in the course of classification.

6. Finally, the result shed some light on the need for revising the manual classification scheme. This implies that a research should be done to verify whether the classification scheme meets the information requirement of journalists.

BIBLIOGRAPHY

Aas, Kjersti and Line Eikvil. 1999. *Text Categorization: A Survey*. Available from http://www.iit.demokritos.gr/~sigletos/references.html

Abiyot Bayou. 2000. *Design and Development of Word Parser for Amharic Language.* Master Thesis at The School of Information Studies For Africa. Adddis Ababa University. Addis Ababa.

Adamson, George W. and Judith A. Bush. 1973. A Method for the Automatic Classification of Chemical Structures. *Information Storage and Retrieval*. 9:561-568.

Ardo, Andres and Trougott Kock. 1997. *Automatic Classification Applied to the Full-Text Internet Documents in a Robot-Generated Subject Indexes. DESIRE II D3.6a, automatic classification, working Paper 2*. Available from http://www.lub.lu.se/desire/DESIRE36a-WP2.html

Beletu Reda. 1982. *A Graphemic Analysis of the Writing System of Amharic*. Paper for the Requirement of the Degree of Bachelor of Art in Linguistics. Addis Ababa University.

Beghtol, Clare. 1986. Bibliographic Classification Theory and Text Linguistics: Aboutness Analysis, Intertextuality and the Cognitive Act of Classifying Documents. *Journal of Documentation*. 42(2):84-113.

Bender, M., and C. Ferguson. eds. 1976. The Ethiopian Writing System. In *Languages in Ethiopia*. London: Oxford University Press.

Bichteler, Julie, and Ronald G. Parsons. 1974. Document Retrieval by Means of an Automatic Classification Algorithm for Citation. *Information Storage and Retrieval*. 10:267-278.

Blosseville, M. J., G. Hebrail, M. G. Monteil, and N. Penot. 1992. Automatic Document Classification: NLP, Statistical Analysis, and ES Techniques Used Together. *In proceedings of the fifteenth annual international ACM SIGIR conference on Research and Development in IR*. Edited by Nicholas Belkin, Peter Ingwersen, and Annelise Mark Pijtersen. pp.51-58.

Borko, Horold and Myrna Bernick. 1963. Automatic Document Classification. *Journal of ACM*. 10: 151-163.

Burden, Peter. n.d. *The Automatic Classification Engine*. Available from http://scitsc.wlv.ac.uk/~jphb/research/old.ace.html.

Cahn, D. F., and J. J. Herr. 1977. *Automatic Document Classification Based on Expert human decision*.

Chen, H. et al. 1994. Automatic Concept Classification of Text from Electronic Meetings. *Communication of the ACM*. 37(10): 56-73.

Cheng, Patrick T. K., and Albert K. W. Wu. 1995. ACS: An Automatic Classification System. *Journal of the American Society for Information Science*. 21(4):289-299.

Chenkuri, Chandra and Michael Ho. Goldwasser. n.d. *Web Search Using Automatic Classification*. Available from http://theory.stanford.edu/people/wass/publications/ Web_Search /Web_Search.html.

Choi, DongSee, Kyung Taek Chong, and SeYoung Park. 1996. Design and Implementation of Automatic Document Classification Using Correlation Between Categories and Keywords. *In Proceedings of the workshop on Information Retrieval with Oriental Languages*. Edited by Sung Hyun Myaeng. Yoosung-Ku: Korea Research and Development Information Center.

Cohen, William W. 1995. Text Categorization and Relational Learning. *In Proceedings of the Twelfth International Conference*. Available from http://citeseer.nj.nec.com /cohen95text.html.

Cohen, William W. 1996. Learning Rules that Classify E-Mail. *In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning and Information Access*.

Cohen, William W. and Yoran Singer. 1996. Context Sensitive Learning Methods for Text Categorization. *In SIGIR '92: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in IR*. pp. 307-315.

ENA. 1993a. *Handbook for Editorial Staff*. Addis Ababa.

ENA. 1993b. *#1 Link to Ethiopia*. Unpublished Material.

ENA. 2000. *1992 Budget Year Performance Evaluation Report*. Unpublished Report.

Enser, P. G. B. 1985. Automatic Classification of Book Material Represented by Back-of the Book Index. *Journal of Documentation*. 41(3):135-155.

Frakes, W. B. and Baeza-Yates, R. eds. 1992. *Data Structures and Algorithm*. New Jersey: Printice Hall PTR.

Frants, Valery I., and Nick I. Kamenoff. 1992. One Approach to Classification of Users and Automatic Clustering of Documents. *Information Processing And Management*. 29(2): 187-195.

Garland, Kathleen. 1983. An Experiment in Automatic Hierarchical Document Classification. *Information Processing And Management*. 19(3):121-129.

Getachew Haile. 1967. *The Problems of Amharic Writing System*. Unpublished.

Gorniak, Peter. 1998. *Sorting Email Messages by Topic*. Available from http://www.cs.ubc.ca /~pgorniak/um/bucfe.html

Griffiths, Alan, Lesley A. Robinson and Peter Willett. 1984. Hierarchic Agglomerative Clustering Methods for Automatic Document Classification. *Journal of Documentation*. 40(3):175-205.

Han, Eui-Hong, George Karypis, and Vipin Kumar. 1999. *Text Categorization Using Weight Adjusted K-Nearest Neighbor Classification*. Available from http://www.cs.umn.edu/~han.

He, Ji, Ah-Hwee Tan, and Chew-Lim Tan. 2000. A Comparative Study on Chinese Text Categorization Methods. *In PRICAI 2000 Workshop on Text and Web Mining*. Edited by A. H. Tan and P. Yu. pp. 24-35.

Hoch, Rainer. 1994. Using IR Techniques for Text Classification in Document Analysis. *In Proceedings of the Seventh Annual International ACM SIGIR*. pp. 31-40.

Hsu, Wen Lin and Sheau-Dong Lang. 1999. *Classification Algorithms for Netnews Articles*. Available from citeseer.nj.nec.com/331418.html

Hunter, E. J. 1995. *Classification Made Simple*. Aldershot: Gower Publishing House.

Jacobs, Paul S. 1992. Joining Statistics with NLP for Text Categorization. *In Proceedings of the Third Conference on Applied Natural Language Processing*. pp:178-185. Chaired by M. Bates and O. Stocks. pp. 178-185.

Jenkins, Charlotte et al. 1997. *Automatic Classification of Web Resources Using Java and DDC*. Available from http://www.scit.wlv.ac.uk/~ex1253/classifier

Joachims, Thorsten. 1996. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. *In Proceedings of the 14th International Conference on Machine Learning ICML97*. pp. 143-151.

Kar, Gawtam, and Lee J. White. 1978. A Distance Measure for Automatic Document Classification by Sequential Analysis. *Information Processing And Management*. 14: 57-69.

Koch, T., and Ardo A. 2000. *Automatic Classification: DESIRE II D3.6a, Overview of results*. Available from http://www.lub.lu.se/desire/DESIRE36a-overview.html

Koch, T., and Vezine-Goetz D. 1998. *Automatic Classification and Content Navigation Support for Web Services DESIRE II*. Available from

http://www.oclc.org/oclc/research/publications/review98/koch-vizine-goetz/automatic.htm.

Kumar, Krishan. 1989. *Theory of Classification*. 4th edition. New Delhi: Vikas Publishing House Pvt. Ltd.

Kwok, J. L. 1975. The Use of Title and Cited Titles as Document Representation for Automatic Classification. *Information Processing And Management*. 2:201-206.

Larkey, Leah S. and W. Bruce Croft. 1996. Combining Classifiers in Text Categorization. *In SIGIR '96: Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 289-297.

Larson, Ray R. 1992. Experiments in Automatic Library of Congress Classification. *Journal of the American Society for Information Science*. 43(2):130-148.

Leslau, Wolf (1965) *An Amharic Text Book of Everyday Usage*. University of California, Los Angles.

Lewis, David D. 1992a. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. *In Proceedings of the fifteenth annual international ACM SIGIR Conference on Research and Development in IR*. Edited by Nicholas Belkin, Peter Ingwersen, and Annelise Mark Pijtersen.

Lewis, David D. 1992b. Feature Selection and Feature Extraction for Text Categorization. *In Proceedings of the Speech and Natural Language workshop*. pp. 212-217.

Lin, Chung-hsin, and Hsinchum Chen. n.d. *An automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents*. Available from http://ai.bpa.arizona.edu/papers/chinese93/chinese93.html

Litman, Diane J. 1996. Cue Phrase Classification Using Machine Learning. *Journal of Artificial Intelligence Research*. 5:53-94.

Losee, R. M. and Hass, S. W. 1995. Sublanguage Terms: Dictionaries, Usage, and Automatic Classification. *Journal of the American Society for Information Science*. 46(7):519-529.

Maron, M. E. 1961. Automatic Indexing: An Experimental Inquiry. *Journal of ACM*. 8: 404-417.

May, Andrew D. 1997. Automatic Classification of E-Mail Messages by Message Type. *Journal of the American Society for Information Science*. 48(1):32-39.

Michie, D., D. J. Spiegelhalter, and C. C. Taylor. eds. 1994. *Machine learning, Neural and Statistical Classification*. New York: Ellis Horwood.

Pao, Miranda Lee. 1989. *Concepts of Information Retrieval*. Colorado: Libraries Unlimited inc.

Pollock, Stephen. 1988. A Rule-Based Message Filtering System. *ACM Transaction on Office Information Systems*. 6(3):232-254.

Qiyu, Zhang, Liu Xiangsheng, and Wang Dongbo. 1996. *Contemporary Classification Systems and Therausrus in China*. Available from www.ifla.org/IV/ifla62/62-qiyz.htm

Rasmussen, E. 1992. Clustering Algorithms. *In Data Structures and Algorithm*. eds. William B. Frakes, and Ricardo Baeza-Yates. New Jersey: Printice Hall PTR.

Ruiz, Miguel E., and Padmini Srinivasan. 1998. Automatic Text Categorization Using Neural Networks. *In Advances in Classification Research: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop*. Edited by. Effhimis Efthimiadis. Information Today, Medford: New Jersey. 8:59-72.

Salton, Gerard. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading: Addison-Wesley Publishing Company.

Scott, Sam, and Stan Marwin. 1998. *Using Lexical Knowledge in Text Classification*. Available from www.citeseer.nj.nec.com/34358.html

Shankar, Shrikanth, and George Karypis. 2000. *A Feature Weight Adjustment Algorithm for Document Categorization*. Available http://www.cs.umn.edu/~karypis.

Sharif, Carolyn A Y. 1988. *Developing an Expert System for Classification of Books Using Micro-Based Experiment System Shells*. British Library Research Department. London.

Smith, Peter D. 1990. *An Introduction to Text Processing*. Cambridge: The MIT Press.

Tague-Sutcliffe, Jean. 1992. Measuring the Informativeness of a Retrieval Process. *In Proceedings of the fifteenth annual international ACM SIGIR conference on Research and Development in IR*. Edited by Nicholas Belkin, Peter Ingwersen, and Annelise Mark Pijtersen.

Takkinen, Juha. 1995. *An adaptive Approach to Text Categorization and Understanding*: A Preliminary study. Available from http://www.ida.liu.se/~juhta/publications.html.

Transitional Government of Ethiopia (TGE). 1995. *A Proclamation for the Establishment of Ethiopia News Agency*. Proclamation no. 115/1995. Negarit Gazeta. 54(13): 125-38.

Whatmore, Geoffrey. 1973. Classification for News Libraries. *Aslib Proceedings*. 25(6):207-219.

Whatmore, Geoffrey. 1978. *The Modern News Library: Documentation of Current Affairs in Newspaper and Broadcasting Libraries*. London: The Library Association.

Willett, Peter. 1981. A Fast Procedure for the Calculation of Similarity Coefficients in Automatic Classification. *Information Processing And Management*. 17:53-60.

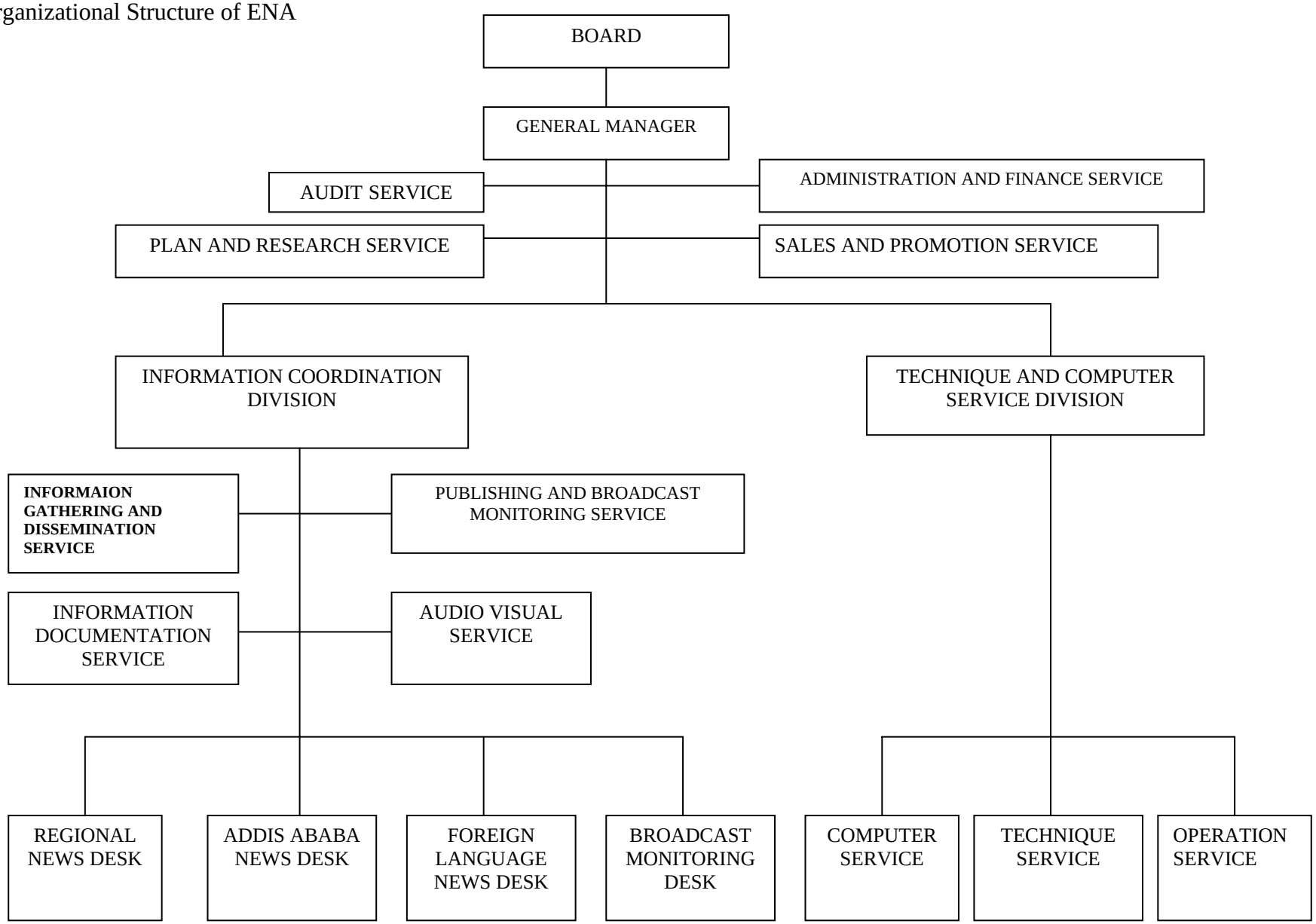
Willett, Peter. 1985. An Algorithm for the Calculation of Exact Term Discrimination Values. *Information Processing And Management*. 21(3): 225-232.

Willett, Peter. 1988. Recent Trends in Hierarchic Document Clustering: A Critical Review. *Information Processing And Management*. 24(5): 577-597.

Yang, Yiming. 1995. Noise Reduction in a Statistical Approach to Text Categorization. *In Proceedings of SIGIR-95*. pp. 562-263.

Yang, Yiming. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval*. 1(1/2): 67-88.

Annex 1: Organizational Structure of ENA



Annex 2: Amharic News Classification Scheme

Parent	Code	Description
ጤናዋ	ሕክአ-ሕአሕ	የሕክምና አገልግሎት -የሕፃናትና የእናቶች ሕክምና
ጤናዋ	ሕክአ-ጤጣክ	የሕክምና አገልግሎት -የጤና ጣቢያዎች፣
ጤናዋ	ሕክአ-መሆስ	የሕክምና አገልግሎት -መደበኛ ሆስፒሎች
ጤናዋ	ሕክአ-ረሆስ	የሕክምና አገልግሎት -ረፈራል ሆስፒሎች
ጤናዋ	ጤዋሚ-ጤፖሊ	ጤና ጥበ፟ሚኒስቴር-የጤና ፖሊሲ
ጤናዋ	ጤዋሚ-ጤስም	ጤና ጥበ፟ሚኒስቴር-የጤና ስምምነቶች
ጤናዋ	ጤዋሚ-ጤአመ	ጤና ጥበ፟ሚኒስቴር-የጤና አስተዳደራዊ መዋቅር
ጤናዋ	ጤዋሚ-ውዜና	ጤና ጥበ፟ሚኒስቴር-የውጭ ዜናዎች
ጤናዋ	ጤዋሚ-አደመ	ጤና ጥበ፟ሚኒስቴር-አዋጆች፣ደንቦችና መግለጫዎች
ጤናዋ	በሽተ-ዓይማ	በሽታዎች- የዓይን ማዝ
ጤናዋ	በሽተ-ማጅገ	በሽታዎች-ማጅራት ገትር
ጤናዋ	በሽተ-ሆውበ	በሽታዎች-የሆድ ውስጥ በሽዎች
ጤናዋ	በሽተ-ደምማ	በሽታዎች-የደም ማነስ
ጤናዋ	በሽተ-ደምግ	በሽታዎች-የደም ግፊት
ጤናዋ	በሽተ-ጉሮቀ	በሽታዎች-የጉሮሮ ቁስል
ጤናዋ	በሽተ-ግርበ	በሽታዎች-ግርሻ በሽ
ጤናዋ	በሽተ-ኩሌረ	በሽታዎች-ኩሌራ
ጤናዋ	በሽተ-ሌበሽ	በሽታዎች-ሌሎች በሽዎች
ጤናዋ	በሽተ-ልድካ	በሽዎች-የልብ ድካም
ጤናዋ	በሽተ-ሣንነ	በሽዎች-የሣንባ ነቀርሣ
ጤናዋ	በሽተ-ተስበ	በሽታዎች-ተስቦ
ጤናዋ	በሽተ-ወባበ	በሽታዎች-ወባ በሽታ
ጤናዋ	በሽተ-ውዜና	በሽዎች-የውጭ ዜናዎች
ጤናዋ	በሽተ-ኤድስ	በሽዎች-ኤድስ
ጤናዋ	በሽተ-አባዘ	በሽታዎች-የአበለ ዘርዕ
ጤናዋ	በሽተ-አስበ	በሽታዎች-የአስም በሽ
ጤናዋ	በሽተ-አእበ	በሽታዎች-የአእምሮ በሽ
ዓመበ	ዓአበ-ሜይዲ	ዓለምአቀፍ በዓላት-ሜይ ዲይ
ዓመበ	ዓአበ-ማር	ዓለምአቀፍ በዓላት-ማርች 8
ዓመበ	ዓአበ-ሌዓበ	ዓለምአቀፍ በዓላት-ሌሎች ዓአበ
ዓመበ	ብሔበ-ዐድድ	ብሔራዊ በዓላት-የዐድዋ ድል
ዓመበ	ብሔበ-ጥምቀ	ብሔራዊ በዓላት-የጥምቀት በዓል
ዓመበ	ብሔበ-ግንሃ	ብሔራዊ በዓላት-ግንቦት 20
ዓመበ	ብሔበ-ፋሲበ	ብሔራዊ በዓላት-የፋሲካ በዓል
ዓመበ	ብሔበ-ሌበዓ	ብሔራዊ በዓላት-ሌሎች በዓላት
ዓመበ	ብሔበ-ልደበ	ብሔራዊ በዓላት-የልደት በዓል

Parent	Code	Description
ዓመበ	ብሔበ-መስቀ	ብሔራዊ በዓላት-የመስቀል በዓል
ዓመበ	ብሔበ-መውሊ	ብሔራዊ በዓላት-መውሊድ
ዓመበ	ብሔበ-ረመዳ	ብሔራዊ በዓላት-ኢድ አልፈጥር
ዓመበ	ብሔበ-ስቅለ	ብሔራዊ በዓላት-ስቅለት
ዓመበ	ብሔበ-ውዜና	ብሔራዊ በዓላት-የውጭ ዜናዎች
ዓመበ	ብሔበ-ኢአአ	ብሔራዊ በዓላት-ኢድ አል አድሐ
ዓመበ	ብሔበ-አድበ	ብሔራዊ በዓላት-የአርበኞች ድል በዓል
ዓመበ	ብሔበ-ዘመመ	ብሔራዊ በዓላት-የዘመን መለወጫ
ዓአግ	ዓአድ	ዓለምአቀፋዊ ድርጅቶች
ዓአግ	ላአጉ	የላቲን አሜሪካ ጉዳይ
ዓአግ	ጉብጀ	ጉብጀቶች
ዓአግ	ሰአጉ	የሰሜን አሜሪካ ጉዳይ
ዓአግ	ውጉሚ	ውጭ ጉዳይ ሚኒስቴር
ዓአግ	ውዜና	የውጭ ዜናዎች
ዓአግ	ኤህጉ	የኤሽያ ሀገሮች ጉዳይ
ዓአግ	ኤምባ	ኤምባሲዎች
ዓአግ	አጎድ	አጎጉራዊ ድርጅቶች
ዓአግ	አሮ	የአውሮ ጉዳይ
ማጎደ	ሕፃማ	ሕፃናት ማሳደጊያ
ማጎደ	ጎዳተ	ጐዳና ተዳዳሪ
ማጎደ	ማጎዋ	ማጎበራዊ ዋስትና
ማጎደ	ውዜና	የውጭ ዜናዎች
ማጎደ	እጓማ	እጓለ ማውታ
ባህለ	ታሪከ-ጥታሪ	ታሪክ-ጥንተ ታሪክ
ባህለ	ታሪክ-ጥታሪ	ታሪክ-ጥንተ ታሪክ
ባህለ	ታሪከ-ቅታሪ	ታሪክ-ቅድመ ታሪክ
ባህለ	ታሪከ-ውዜና	ታሪክ-የውጭ ዜና
ባህለ	ታሪከ-ዘታሪ	ታሪክ-ዘመናዊ ታሪክ
ባህለ	ቋንቀ-ውዜና	ቋንቋዎች-የውጭ ዜና
ባህለ	ቋንቀ-ኢቋጥ	ቋንቋዎችጠ-የኢትዮጵያ ቋንቋዎች ጥናት
ባህለ	መንባ-ፈልጥ	መንፈሳዊ ባህልጠ-የፊልም ጥበብ
ባህለ	መንባ-ግጥቅ	መንፈሳዊ ባህል-ግጥምና ቅኔ
ባህለ	መንባ-መዘቀ	መንፈሳዊ ባህል-ሙዚ
ባህለ	መንባ-ቅቅጽ	መንፈሳዊ ባህል-ቅርጻቅርጽ
ባህለ	መንባ-ተውነ	መንፈሳዊ ባህልጠ-ተውኔት
ባህለ	መንባ-አብመ	መንፈሳዊ ባህል-አብያተ መጻሕፍት
ባህለ	መንባ-አመዛ	መንፈሳዊ ባህል-አብያተ መዛግብት

Parent	Code	Description
ባህለ	መንባ-አመዘ	መንፈሳዊ ባህል-አብያተ መዘክሮች
ባህለ	ቁሣባ-ቁሣቅ	ቁሣዊ ባህል-ቁሣዊ ቅርሶች
ባህለ	ቁሣባ-ቅጥም	ቁሣዊ ባህል-የቅርስ ጥናትና ምርምር
ባህለ	ቁሣባ-ውዜና	ቁሣዊ ባህል-የውጭ ዜናዎች
ባህለ	ትውባ-ብብት	ትውፊዊ ባህል-የብሔር/ብሔረሰቦች ትውፊት
ብለ	ሕፖተ	ሕዝባዊ የፖለቲካ ተሳትፎ
ብለ	ሕእጠ	የሕዝብ እንባ ጠባቂ
ብለ	ጠሚጽ	ጠ/ሚ/ጽ/ቤት
ብለ	ሚምቤ	ሚኒስትሮች ም/ቤት
ብለ	ሚመቤ	ሚኒስትር መ/ቤቶች
ብለ	ፖር	የፖለቲካ ርቲዎች
ብለ	ብምጫ	ብሔራዊ ምርጫ
ብለ	ክምቤ	ክልል ምክር ቤት
ብለ	ክመስ	ክልል መስተዳድር
ብለ	ኰሚሺ	ኰሚሺኖች
ብለ	ፌምቤ	ፌዴሬሽን ምክር ቤት
ብለ	ፕጽቤ	ፕሬዚዳንት ጽ/ቤት
ብለ	ሰመኰ	የሰብዓዊ መብት ኰሚሽን
ብለ	ተምቤ	ተወካዮች ምክር ቤት
ብለ	ውዜና	የውጭ ዜናዎች
ብለ	ውዜና	የውጭ ዜናዎች
ብለ	አደመ	አዋጆች፣ደንቦችና መግለጫዎች
ብለ	አደመ	አዋጆች፣ደንቦችና መግለጫዎች
ብለ	አደመ	አዋጆች፣ደንቦችና መግለጫዎች
ብለ	አደመ	አዋጆች፣ደንቦችና መግለጫዎች
ብለ	አደመ	አዋጆች፣ደንቦችና መግለጫዎች
ፍፍት	ሕአአ-ዐሕ	የሕግ አስፈፃሚ አካላት-ዓቢሕግ
ፍፍት	ሕአአ-ዐሕ	የሕግ አስፈፃሚ አካላት-ዓቢሕግ
ፍፍት	ሕአአ-ፖሊሠ	የሕግ አስፈፃሚ አካላት-ፍርድ ቤቶች
ፍፍት	ሕአአ-ፍቤት	የሕግ አስፈፃሚ አካላት-ፍርድ ቤቶች
ፍፍት	ሕአአ-ውዜና	የሕግ አስፈፃሚ አካላት-የውጭ ዜናዎች
ፍፍት	ሕአአ-አደመ	የሕግ አስፈፃሚ አካላት-አዋጆች፣ደንቦችና መግለጫዎች
ፍፍት	ፖሊች-ፍብክ	ፖሊስና ችሎቶች-የፍትሐብሔር ክሶች
ፍፍት	ፖሊች-ሙስነ	ፖሊስና ችሎቶች-ሙስና
ፍፍት	ፖሊች-ወንጀ	ፖሊስና ችሎቶች-የወንጀል ክሶች
ፍፍት	ፖሊች-ውዜና	ፖሊስና ችሎቶች-የውጭ ዜናዎ
ፍፍት	ፖሊች-አደመ	ፖሊስና ችሎቶች-አዋጆች፣ደንቦችና መግለጫዎች

Parent	Code	Description
ፍፍት	ፖሊች-ዘማክ	ፖሊስና ችሎቶች-የዘር ማጥፋት ክሶች
ገበዜ	ጥጥዋ-ሽ	የጥራጥሬ ዋጋ-ሽንብራ
ገበዜ	ጥጥዋ-ባ	የጥራጥሬ ዋጋ-ባቄላ
ገበዜ	ጥጥዋ-አ	የጥራጥሬ ዋጋ-አተር
ገበዜ	ሸሸዋ	የሸቀጣሸቀጥ ዋጋ
ገበዜ	ፋምዋ-ቤቁ	የፋብካ ምርቶች-የቤት ቁሳቁሶች
ገበዜ	ፋምዋ-አል	የፋብካ ምርቶች ዋጋ-አልባሳት
ገበዜ	መጠዋ-ለመ	የመጠጥ ዋጋ-ለስላሳ መጠጦች
ገበዜ	መጠዋ-አመ	የመጠጥ ዋጋ-የአልኮል መጠጦች
ገበዜ	ቅቅዋ-ጨ	የቅመማ ቅመም ዋጋ-ጨው
ገበዜ	ቅቅዋ-በር	የቅመማ ቅመም ዋጋ-በርበሬ
ገበዜ	ቅቅዋ-ስ	የቅመማ ቅመም ዋጋ-ስኳር
ገበዜ	ቅቅዋ-ዘ	የቅመማ ቅመም ዋጋ-ዘይት
ገበዜ	ውጭም	የውጭ ምንዛ
ገበዜ	አፍዋ-ሽን	የአትክልትና ፍራፍሬ ዋጋ-ሽንኩርት
ገበዜ	አፍዋ-ብ	የአትክልትና ፍራፍሬ ዋጋ-ብርቱካን
ገበዜ	አፍዋ-ድ	የአትክልትና ፍራፍሬ ዋጋ-ድንች
ገበዜ	አፍዋ-ሌ	የአትክልትና ፍራፍሬ ዋጋ-ሌሎች
ገበዜ	አፍዋ-ሙ	የአትክልትና ፍራፍሬ ዋጋ-ሙዝ
ገበዜ	እህዋ-ጤ	የእህል ዋጋ-ጤፍ
ገበዜ	እህዋ-ማ	የእህል ዋጋ-ማሽላ
ገበዜ	እህዋ-በ	የእህል ዋጋ-በቆሎ
ገበዜ	እህዋ-ገ	የእህል ዋጋ-ገብስ
ገበዜ	እህዋ-ስ	የእህል ዋጋ-ስንዴ
መከጸ	ሀውጸ-ሀውደ	የሀገር ውስጥ ጸጥ-የሀገር ውስጥ ደኅንነት
መከጸ	ሀውጸ-ውዜና	የሀገር ውስጥ ጸጥ-የውጭ ዜናዎች
መከጸ	ሀውጸ-ኢስደ	የሀገር ውስጥ ጸጥ-ኢሚግሬሽንና ስደተኞች
መከጸ	ሀውጸ-አደመ	የሀገር ውስጥ ጸጥ-አዋጆች፣ ደንቦችና መግለጫዎች
መከጸ	መከሚ-ሚሊሠ	መከላከያ ሚኒስቴር-ሚሊሺያ ሠራዊት
መከጸ	መከሚ-ምድጦ	መከላከያ ሚኒስቴርጠ-የምድር ጦር
መከጸ	መከሚ-ጦኃጠ	መከላከያ ሚኒስቴርጠ-የጦር ኃይሎች ጠቅላይ መምያ
መከጸ	መከሚ-ጦርዘ	መከላከያ ሚኒስቴር-የጦርነት ዘገባ
መከጸ	መከሚ-ወታስ	መከላከያ ሚኒስቴር-ወደራዊ ስምምነቶች
መከጸ	መከሚ-ውዜና	መከላከያ ሚኒስቴር-የውጭ ዜናዎች
መከጸ	መከሚ-አደመ	መከላከያ ሚኒስቴር-አዋጆች፣ደንቦችና መግለጫዎች
መከጸ	መከሚ-አየኃ	መከላከያ ሚኒስቴር-የአየር ኃይል
ሣቴክ	ባህተ	የባህላዊ ቴክኖሎጂ

Parent	Code	Description
ሣቴክ	ሣምጥ	የሣይንስ ምርምርና ጥናት
ሣቴክ	ሣቴስ	የሣይንስና ቴክኖሎጂ ስምምነቶች
ሣቴክ	ቴምጥ	የቴክኖሎጂ ምርምርና ጥናት
ሣቴክ	ውዜና	የውጭ ዜናዎች
ጠቅል	ሕዝማ	ሕዝባዊ ማኅበራት
ጠቅል	ማተፈ	ማኅበራዊ ተሐድሶ ፈንድ
ጠቅል	ከልቤ	ከተማ ልማትና ቤት
ጠቅል	ሙያማ	የሙያ ማኅበራት
ጠቅል	ሲቪሰ	ሲቪል ሰርቪስ
ጠቅል	ስታቲ	ስቲስቲክስ
ጠቅል	ቫስታ	ቫይታል ስቲስቲክስ
ስፖር	ሜቴ	የሜዳ ቴኒስ
ስፖር	ማአ	ማርሻል አርት
ስፖር	ባድ	ባድሚንተን
ስፖር	ባስ	ባህላዊ ስፖርቶች
ስፖር	ኦሊኬ	ኦሊምፒክ ኮሚቴ
ስፖር	ሞው	ሞተር ውድድር
ስፖር	ቢሲ	ቢሲክሌት
ስፖር	ቸዘ	ቼዝ
ስፖር	ፈረ	የፈረስ ስፖርት
ስፖር	ቦክ	ቦክስ
ስፖር	ቦው	ቦውሊንግ
ስፖር	ክማ	ክብደት ማንሳት
ስፖር	መካ	መረብ ኳስ
ስፖር	ቅካ	ቅርጫት ኳስ
ስፖር	ስፖኬ	ስፖርት ኮሚሽን
ስፖር	ስፖፌ	የስፖርት ፌደሬሽኖች
ስፖር	ውዋ	ውኃ ዋና
ስፖር	ውዜና	የውጭ ዜናዎች
ስፖር	እካ	የእጅ ኳስ
ስፖር	አጉ	የአካል ጉዳተኞች ስፖርት
ስፖር	እግካ	እግር ኳስ
ስፖር	አቅ	የአካል ቅርጽ
ስፖር	አትሌ	አትሌቲክስ
ትምህ	ሌትተ-ሕትተ	ሌሎች የትምህርት ተቋማት-የሕዝብ ትምህርት ቤቶች
ትምህ	ሌትተ-ሃትቤ	ሌሎች የትምህርት ተቋማት-የሃይማኖት ትምህርት ቤቶች
ትምህ	ሌትተ-ሚትቤ	ሌሎች የትምህርት ተቋማት-የሚሲዮን ትምህርት ቤቶች

Parent	Code	Description
ትምህ	ሌትተ-ድግተ	ሌሎች የትምህርት ተቋማት-የድርጅቶች ማሠልጠኛ ተቋማት
ትምህ	ሌትተ-ግትተ	ሌሎች የትምህርት ተቋማት-የግል የትምህርት ተቋማት
ትምህ	መምተ-መምማ	መምህራንና ተማዎች -የመምህራን ማኅበር
ትምህ	መምተ-ተማማ	መምህራንና ተማሪዎች -የተማዎች ማኅበራት
ትምህ	መምተ-ተማጉ	መምህራንና ተማዎች -የተማዎች ጉዳይ
ትምህ	ትድስ-ሀትጉ	የትምህርት ድጋፍ ሰጪዎች -ሀገርአቀፍ የትምህርት ጉባኤ
ትምህ	ትድስ-ትምፋ	የትምህርት ድጋፍ ሰጪዎች -የትምህርት ፋሲሊቲ
ትምህ	ትድስ-ትምመ	የትምህርት ድጋፍ ሰጪዎች -የትምህርት መሣያዎች
ትምህ	ትድስ-ትምቴ	የትምህርት ድጋፍ ሰጪዎች -የትምህርት ቴክኖሎጂ
ትምህ	ትድስ-ትአአ	የትምህርት ድጋፍ ሰጪዎች -የትምህርት አደረጃጀትና አመራር
ትምህ	ትድስ-ትአአ	የትምህርት ድጋፍ ሰጪዎች -ሀገርአቀፍ የትምህርት
ትምህ	ትምፈ-ብሔፈ	የትምህርት ምዘናና ፈተና-ብሔራዊ ፈተናዎች
ትምህ	ትምፈ-ተሠም	የትምህርት ምዘናና ፈተና -የተማዎችና ሠልጣኞች ምረ
ትምህ	ትምእ-ኅልት	የትምሕርት እርከን-የኅልማሶች ትምሕርት
ትምህ	ትምእ-ሀደት	የትምሕርት እርከን-የሀለተኛ ደረጃ ትምሕርት
ትምህ	ትምእ-ከደት	የትምሕርት እርከን-የከፍተኛ ደረጃ ትምሕርት
ትምህ	ትምእ-መደት	የትምሕርት እርከን-የመጀመሪያ ደረጃ ትምሕርት
ኢኮኖ	ማህል-ማዕሀ	የማእድን ሀብት ልማት-የማዕድናት ሀብት
ኢኮኖ	ማህል-ማልስ	የማእድን ሀብት ልማት-የማዕድን ልማት ስምምነቶች
ኢኮኖ	ማህል-ጂኦጥ	የማእድን ሀብት ልማት-የጂኦሎጂ ጥናት
ኢኮኖ	ማህል-ጂተጥ	የማእድን ሀብት ልማት-የጂኦተርማል ጥናት
ኢኮኖ	ማህል-ውዜና	የማእድን ሀብት ልማት-የውጭ ዜናዎች
ኢኮኖ	ማህል-አማፍ	የማእድን ሀብት ልማት-የአግሮ ማዕድናት ፍለጋና ጥናት
ኢኮኖ	ማህል-አደመ	የማእድን ሀብት ልማት-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ግብና-ምራመ	ግብርና-በምግብ ራስን መቻል
ኢኮኖ	ግብና-ምግዋ	ግብርና-የምግብ ዋስትና
ኢኮኖ	ግብና-ተሀል	ግብርና-የተፈጥሮ ሀብት ልማት
ኢኮኖ	ግብና-ውዜና	ግብርና-የውጭ ዜናዎች
ኢኮኖ	ግብና-አደመ	ግብርና-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ግብና-እሀል	ግብርና-የእንስሳት ሀብት ልማት
ኢኮኖ	ግብና-እልማ	ግብርና-የእርሻ ልማት
ኢኮኖ	ኮንስ -ሕንኮ	ኮንስትራክሽን-የሕንፃ ኮንስትራክሽን
ኢኮኖ	ኮንስ -መንኮ	ኮንስትራክሽን-የመንገድ ኮንስትራክሽን
ኢኮኖ	ኮንስ-ባቡኮ	ኮንስትራክሽን-የባቡር ሐዲዶች ኮንስትራክሽን
ኢኮኖ	ኮንስ-ኮንስ	ኮንስትራክሽን- የኮንስትራክሽን ስምምነቶች
ኢኮኖ	ኮንስ-ውዜና	ኮንስትራክሽን-የውጭ ዜናዎች

Parent	Code	Description
ኢኮኖ	ኮንስ-አማኮ	የአይሮፕላን ማረፊያ ኮንስትራክሽን
ኢኮኖ	ኮንስ-አደመ	ኮንስትራክሽን-አዋጆች፣ ደንቦችና መግለጫዎች
ኢኮኖ	ልማማ-ኦልማ	የልማት ማኅበራት-የኦሮሚያ ልማት ማኅበር
ኢኮኖ	ልማማ-ደልማ	የልማት ማኅበራት-የደቡብ ሕዝቦች ልማት ማኅበር
ኢኮኖ	ልማማ-ሌልማ	የልማት ማኅበራት-ሌሎች የልማት ማኅበራት
ኢኮኖ	ልማማ-ልማስ	የልማት ማኅበራት-የልማት ማኅበራት ስምምነቶች
ኢኮኖ	ልማማ-ትልማ	የልማት ማኅበራት-የትግራይ ልማት ማኅበር
ኢኮኖ	ልማማ-ውዜና	የልማት ማኅበራት-የውጭ ዜናዎች
ኢኮኖ	ልማማ-አልማ	የልማት ማኅበራት-የአማራ ልማት ማኅበር
ኢኮኖ	ቱሪዝ -ቱሪን	ቱሪዝም-የቱሪዝም ማስፋፊያ
ኢኮኖ	ቱሪዝ -ውሀኅ	ቱሪዝም-የውጭ ሀገር ኅብኚዎች
ኢኮኖ	ቱሪዝ-ሀውኅ	ቱሪዝም-የሀገር ውስጥ ኅብኚዎች
ኢኮኖ	ቱሪዝ-ቱሪሀ	ቱዝም-የቱዝም ሀብት
ኢኮኖ	ቱሪዝ-ውዜና	ቱሪዝም-የውጭ ዜናዎች
ኢኮኖ	ቱሪዝ-አደመ	ተሪዝም-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ቱሪዝ-አስድ	ቱሪዝ-አስኅብኚ ድርጅቶች
ኢኮኖ	ትራን -የብት	ትራንስፖርት-የየብስ ትራንስፖርት
ኢኮኖ	ትራን-ባሕት	ትራንስፖርት-የባሕር ትራንስፖርት
ኢኮኖ	ትራን-ትራስ	ትራንስፖርት-የትራንስፖርት ስምምነቶች
ኢኮኖ	ትራን-ውዜና	ትራንስፖርት-የውጭ ዜናዎች
ኢኮኖ	ትራን-አደመ	ትራንስፖርት-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ትራን-አየት	ትራንስፖርት-የአየር ትራንስፖርት
ኢኮኖ	ውኃሀ	ውኃ ሀብት
ኢኮኖ	ውኃሀ-ሐይጉ	ውኃ ሀብት-የሐይቆች ጉዳይ
ኢኮኖ	ውኃሀ-ሽልጥ	ውኃ ሀብት-የሸለቆዎች ልማት ጥናት
ኢኮኖ	ውኃሀ-መስል	ውኃ ሀብት-የመስኖ ልማት
ኢኮኖ	ውኃሀ-መውል	ውኃ ሀብት-የመጠጥ ውኃ ልማት
ኢኮኖ	ውኃሀ-ወንጉ	ውኃ ሀብት-የወንዞች ጉዳይ
ኢኮኖ	ውኃሀ-ውዜና	ውኃ ሀብት-የውጭ ዜናዎች
ኢኮኖ	ውኃሀ-አደመ	ውኃ ሀብት-አዋጆች፣ ደንቦችና መግለጫዎች
ኢኮኖ	ኢንዱ	ኢንዱስትሪ
ኢኮኖ	ኢንዱ-ጨጨኢ	ኢንደስትሪ-የጨርጨርቅ ኢንደስትሪ
ኢኮኖ	ኢንዱ-ብብኢ	ኢንደስትሪ-የብረታብረት ኢንደስትሪ
ኢኮኖ	ኢንዱ-ኬምኢ	ኢንደስትሪ-የኬሚካል ኢንደስትሪ
ኢኮኖ	ኢንዱ-ከኢን	ኢንደስትሪ-ከባድ ኢንደስትሪ
ኢኮኖ	ኢንዱ-ምግኢ	ኢንደስትሪ-የምግብ ኢንደስትሪ
ኢኮኖ	ኢንዱ-መጠፋ	ኢንደስትሪ-የመጠጥ ፋብሪካዎች

Parent	Code	Description
ኢኮኖ	ኢንዱ-ቆኢን	ኢንደስትሪ-የቆዳ ኢንደስትሪ
ኢኮኖ	ኢንዱ-ትምፋ	ኢንደስትሪ-የትምባሆ ፋብሪካዎች
ኢኮኖ	ኢንዱ-ወማድ	ኢንደስትሪ-የወረቀት ማተሚያ ድርጅቶች
ኢኮኖ	ኢንዱ-ውዜና	ኢንደስትሪ-የውጭ ዜናዎች
ኢኮኖ	ኢንዱ-አደመ	ኢንደስትሪ-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ኢንዱ-አግኢ	ኢንደስትሪ-አግሮ ኢንደስትሪ
ኢኮኖ	ኢንዱ-አነኢ	ኢንደስትሪ-አነስተኛ ኢንደስትሪ
ኢኮኖ	ኢነር -ኢልስ	ኢነርጂ-የኢነርጂ ልማት ስምምነቶች
ኢኮኖ	ኢነር -አደመ	ኢነርጂ-አዋጆች፣ደንቦችና መግለጫዎች
ኢኮኖ	ኢነር -አኢም	ኢነርጂ-አማራጭ የኢነርጂ ምንጮች
ኢኮኖ	ኢነር-ኃኤሌ	ኢነርጂ-የኃይድሮ ኤሌክትሪክሲቲ
ኢኮኖ	ኢነር-ኃኤሌ	ኢነርጂ-የኃይድሮ ኤሌክትክሲቲ
ኢኮኖ	ኢነር-ተጋል	ኢነርጂ-የተፈጥሮ ጋዝ ልማት
ኢኮኖ	ኢነር-ተዘፍ	ኢነርጂ-የተፈጥሮ ዘይት ፍለጋ
ኢኮኖ	ኢነር-ውዜና	ኢነርጂ-የውጭ ዜናዎች
ኢኮኖ	ኢነር-ኢሣተ	ኢነርጂ-የኢነርጂ ሣይንስና ቴክኖሎጂ
ኢኮኖ	ኢነር-ኢአዋ	ኢነርጂ-የኢነርጂ አቅርቦትና ዋጋ
ኢኮኖ	ኢንቨ -ውጭኢ	ኢንቬስትመንት-የውጭ ኢንቬስትመንት
ኢኮኖ	ኢንቨ -ኢንማ	ኢንቬስትመንት-የኢንቬስመንት ማስተዋወቂያ
ኢኮኖ	ኢንቨ -ኢንስ	ኢንቬስትመንት-የኢንቬስትመንት ስምምነቶች
ኢኮኖ	ኢንቨ-ሀውኢ	ኢንቬስትመንት-የሀገር ውስጥ ኢንቬስትመንት
ኢኮኖ	ኢንቨ-ውዜና	ኢንቬስትመንት-የውጭ ዜናዎች
ኢኮኖ	ኢንቨ-አደመ	ኢንቬስትመንት-አዋጆች፣ደንቦችና መግለጫዎች
አደዎ	ሰሠአ-መአ	ሠራሽ አደጋዎች-የመርከብ አደጋ
አደዎ	ሰሠአ-ትአ	ሰው ሠራሽ አደጋዎች-የትራፊክ አደጋ
አደዎ	ሰሠአ-ውዜና	ሰው ሠራሽ አደጋዎች-የውጭ ዜናዎች
አደዎ	ሰሠአ-እአ	ሰው ሠራሽ አደጋዎች-የአይሮፕላን አደጋ
አደዎ	ተፈአ-ድር	የተፈጥሮ አደጋዎች-ድርቅ
አደዎ	ተፈአ-መመ	የተፈጥሮ አደጋዎች -የመሬት መንቀጥቀጥ
አደዎ	ተፈአ-ወሙ	የተፈጥሮ አደጋዎች-የወንዝ ሙላት
አደዎ	ተፈአ-ውዜና	የተፈጥሮ አደጋዎች-የውጭ ዜናዎች
አደዎ	ተፈአ-እገ	የተፈጥሮ አደጋዎች -እሣተ ገሞራ
አደዎ	አመዝ-አኂእ	የአደጋ መከላከልና ዝግጁነት-የአደጋ ጊዜ አርዳዎች
አደዎ	አመዝ-አቅማ	የአደጋ መከላከልና ዝግጁነትጠ-የአደጋ ቅድሚያ ማስጠንቀቂያ
አአገ	ፊዘ	ፊቸር ዘገባ
አአገ	ድዜ	ድንገተኛ- ዜና
አአገ	ቦዜ	ቦግታ- ዜና

Parent	Code	Description
አአገ	ክሮ	ክሮኖሎጂ
አአገ	ለአርኢ ትኩረት	ለአርኢ ትኩረት
አአገ	መደዜ	መደበኛ ዜና
አአገ	ቀዳማይ-ዜና	ቀዳማይ-ዜና
አአገ	ተነዜ	የተነሳሽነት ዜና
አአገ	ፆመ	ልመጠይቅ
አአገ	እግድ	እግድ
አአገ	እቅዜ	የእቅድ ዜና
አአገ	እርማት	እርማት
አአገ	አስዜ	አስቸኳይ - ዜና
አአገ	የሚዜ	የሚጠበቁ - ዜናዎች
አአገ	ዜዕ	ዜና ዕረፍት
አአገ	ዜና-ዋዜማ	ዜና-ዋዜማ
አአገ	ዜት	ዜና ትንኔ
አየጸ	ዕት	ዕለዊ ትንበያ
አየጸ	ረጊት	የረዥም ጊዜ ትንበያ
አየጸ	አጊት	የአጭር ጊዜ ትንበያ

Annex 3: Amharic Fidel

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ			ሟ		
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ					
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኍ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	኱	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ			ዧ		
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጕ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ					
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ

Numerals

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

Punctuation marks

:	÷
፣	።
፤	?
!	( )

Source: Bender et al. 1976

Annex 4: List of Suffixes

Suffixes		
ና	ናዉ	ንና
ናና	ናዉና	ንና
ዎች	ናዉና	ንናም
ዎቹ	ናዉም	ንናም
ዎቹና	ናዉም	ንም
ዎቹናም	ናዉን	ንምና
ዎቹናን	ም	ንምን
ዎቹም	ምና	ንምዉ
ዎቹምና	ምናም	ንን
ዎቹምን	ምናን	ካት
ዎቹን	ምቹ	ዉ
ዎቹንና	ምን	ዉና
ዎቹንም	ምንና	ዉናም
ዎችና	ምንም	ዉናን
ዎችናም	ምንም	ዉናዉ
ዎችናን	ምንን	ዉም
ዎችም	ምንን	ዉምና
ዎችምና	ምንዉ	ዉምን
ዎችምን	ምንዉ	ዉን
ዎችን	ምዉ	ዉንና
ዎችንና	ምዉ	ዉንም
ዎችንም	ምዉና	ዉንን
ናም	ምዉና	
ናምና	ምዉም	
ናምን	ምዉም	
ናን	ምዉን	
ናንና	ምዉን	
ናንም	ን	
ናንን		

Annex 5. ANC Program Code

```
' Global variables used by the different procedures
Dim db As Database
Dim newdoc As Integer
Dim rst, rst1, rst2, rst3, rst4, rst5, rst6, rst7, rst8, rst9 As Recordset
Dim rsta, rstc, rste, r As Recordset
Dim found5, found4, found3, found2  As Boolean

Private Sub Classify_Click()
'This procedure classifies Amharic text given the text or the file where the text is stored
Screen.MousePointer = vbHourglass  'Change mouse pointer to hourglass.
Call GnerateWord
Call CalculateDf
Call CalculateWeight
Dim classa, classc, classe As Single
Dim dena, denc, dene, dendoc As Single
'denominators to hold sum of square for the classes and document
   classa = 0
   dena = 0
   classc = 0
   denc = 0
   classe = 0
   dene = 0
   dendoc = 0
   While Not rst8.EOF
   st = Trim(rst8!word)
   dendoc = dendoc + rst8!Weight * rst8!Weight
   rst7.FindFirst "word = '" & st & "'"
   If Not rst7.NoMatch Then
   classa = classa + rst8!Weight * rst7!Weightinclassa
   classc = classc + rst8!Weight * rst7!Weightinclassc
   classe = classe + rst8!Weight * rst7!Weightinclasse
   End If
```

```
    rst8.MoveNext
    If Not rst7.BOF Then rst7.MoveFirst
    Wend
     If Not rsta.BOF Then rsta.MoveFirst
    While Not rsta.EOF
    dena = dena + rsta!Weightinclassa * rsta!Weightinclassa
    rsta.MoveNext
    Wend
    If Not rstc.BOF Then rstc.MoveFirst
    While Not rstc.EOF
    denc = denc + rstc!Weightinclassc * rstc!Weightinclassc
    rstc.MoveNext
    Wend
    If Not rste.BOF Then rste.MoveFirst
    While Not rste.EOF
    dene = dene + rste!Weightinclasse * rste!Weightinclasse
    rste.MoveNext
    Wend
    If dena <> 0 Then classa = classa / Sqr(dena * dendoc)
    If denc <> 0 Then classc = classc / Sqr(denc * dendoc)
    If dene <> 0 Then classe = classe / Sqr(dene * dendoc)
    Max = classa
    tclas = "accident"
    If classc > Max Then
       Max = classc
       tclas = "culture"
    End If
    If classe > Max Then
       Max = classe
       tclas = "economy"
    End If
If Not rst8.BOF Then rst8.MoveFirst
While Not rst8.EOF
    rst8.Delete
```

```
    If Not rst8.BOF Then rst8.MoveFirst
Wend
  Screen.MousePointer = vbDefault    ' Return mouse pointer to normal.
  MsgBox "the document belongs to class " & tclas     ' display the result to the user
Rst.Close
Rst1.Close
Rst2.Close
Rst3.Close
Rst4.Close
Rst5.Close
Rst6.Close
Rst7.Close
Rst8.Close
Rsta.Close
Rstc.Close
Rste.Close
End Sub

Private Sub Exit_Click()
'Exits the program
End
End Sub

Private Sub Dir1_Change()
' This procedure synchronizes the directory and file list controls.
File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
' This procedure synchronizes the drive control and directory control.
On Error GoTo l1
Dir1.Path = Drive1.Drive
GoTo l2
l1:
```

```
    MsgBox "You may not have inserted a disk in the disk drive. Try again"
    Drive1.Drive = "c:"
l2:
End Sub

Private Sub Form_Load()
' This procedure sets default value for global variables.
Dir1.Path = Drive1.Drive
File1.Pattern = "*.doc;*.rtf"
File1.Path = Dir1.Path
Set db = OpenDatabase("e:\automatic classificatin\automatic classification.mdb", True)
Set rst1 = db.OpenRecordset("words of each class", dbOpenDynaset) 'Used to hold all
‘document terms used to produce the vector table
Set rst2 = db.OpenRecordset("stopwordlist", dbOpenDynaset)
Set rst3 = db.OpenRecordset("words of new documents", dbOpenDynaset) 'used to hold
words ‘of documents that are classified after the production of the vector table
Set rst4 = db.OpenRecordset("PrefixTable", dbOpenDynaset)
Set rst5 = db.OpenRecordset("distinct all", dbOpenDynaset) 'used to hold terms of document
‘frequency information
Set rst6 = db.OpenRecordset("NumberofNewDocuments", dbOpenDynaset)
Set rst7 = db.OpenRecordset("real vector table", dbOpenDynaset) 'the vector table
Set rst8 = db.OpenRecordset("new document vector", dbOpenDynaset) 'table used to hold the
‘vector of the document to be classified
Set rsta = db.OpenRecordset("select word, weightinclassa from [real vector table] where
weightinclassa>0", dbOpenDynaset)   'used to hold vector of accident class
Set rstc = db.OpenRecordset("select word, weightinclassc from [real vector table] where
weightinclassc>0", dbOpenDynaset)   'used to hold vector of culture class
Set rste = db.OpenRecordset("select word, weightinclasse from [real vector table] where
weightinclasse>0", dbOpenDynaset)  'used to hold vector of economy class
newdoc = 0
End Sub

Private Sub GnerateWord()
```

```
'This procedure generates a stemmed term frequency for a given document. It removes stop
‘words. It also store high frequency terms in a new document vector
Dim st1, st2, st3 As String
Dim i As Long
Set rst = db.OpenRecordset("temporarytable", dbOpenDynaset)
If RichTextBox1.Text = "" And File1.ListIndex = -1 Then
   MsgBox "You have to either enter the text to classify or choose a file"
   Exit Sub
End If
found5 = False
found4 = False
found3 = False
found2 = False
st3 = ""      'used to hold words extracted from the document
countstwrd = 0
   newdocn = rst6!NewDocnumber + 1 'variable used to hold
   'number of new documents
   rst6.Edit
   rst6!NewDocnumber = rst6!NewDocnumber + 1
   rst6.Update
fname = "ENA" & newdocn   'fname is unique identifier of documents classified
' and holds the newdocument id or the filename of the text to be classified
If RichTextBox1.Text = "" And File1.ListIndex <> -1 Then
   fname = File1.List(File1.ListIndex)
   Set fs = CreateObject("Scripting.FileSystemObject")
   d = Dir1.Path
   If Not (Dir1.Path Like "?:\") Then
      d = Dir1.Path & "\"
   End If
   st1 = d & fname
   RichTextBox1.LoadFile st1
End If
   st2 = RichTextBox1.Text
   st2 = Trim(st2)
```

```
    i = Len(st2)
    'i holds the length of the news item
For j = 1 To i
 ' delimeters are space, netelaserez(comma), direbserez, aratnetib, hulet netib, '', ', /, (, ), ?
ch = Mid(st2, j, 1)
Select Case ch
Case " ", "½", "Ý", "¿", Chr(188), vbTab, Chr(41), Chr(171), Chr(180), vbCr, vbLf, Chr(39)
If (Not IsNumeric(st3)) And (Len(st3) > 1) Then
    If Not isvalid(st3) Then GoTo l1
    st3 = RemoveDotAndHyphen(st3)
    st3 = change2commonform1(st3)
    st3 = change2commonform2(st3)
    st3 = change2commonform3(st3)
    st3 = change2commonform4(st3)
    st3 = change2commonform5(st3)
    st3 = removepreffix(st3)
    st3 = removesuffix5(st3)
    If Not found5 Then st3 = removesuffix4(st3)
    If Not found4 Then st3 = removesuffix3(st3)
    If Not found3 Then st3 = removesuffix2(st3)
    If Not found2 Then st3 = removesuffix1(st3)
'Check for stop word list. If it is stop word then process the next word
    If Not rst2.BOF Then rst2.MoveFirst
      rst2.FindFirst "word = '" & st3 & "'"
    If Not rst2.NoMatch Or st3 = "" Then
      GoTo l1
    End If
   'check if it is the first word or repeated word
   If Not rst.BOF Then rst.MoveFirst
   rst.FindFirst "word = '" & st3 & "'"
   If rst.NoMatch Then
    rst.AddNew
    rst!word = st3
    rst!frequency = 1
```

```
    rst!FileName = fname
    rst.Update
   Else
    rst.Edit
    rst!frequency = rst!frequency + 1
    rst.Update
   End If
End If
l1:
  ' initialize the variables again
   st3 = ""
   found5 = False
   found4 = False
   found3 = False
   found2 = False
Case Else
   st3 = st3 + ch
End Select
Next
If Not rst.BOF Then rst.MoveFirst
While Not rst.EOF
    rst3.AddNew
    rst3!word = rst!word
    rst3!frequency = rst!frequency
    rst3!FileName = rst!FileName
    rst3.Update
    If rst!frequency > 1 Then
        rst8.AddNew
        rst8!FileName = rst!FileName
        rst8!word = rst!word
        rst8!frequency = rst!frequency
        rst8.Update
    End If
    rst.Delete
```

```
    If Not rst.BOF Then rst.MoveFirst
Wend
'Since the tables opened here will be used in the subsequent procedures the following codes
‘moves the pointers to the 1st record
If Not rst3.BOF Then rst3.MoveFirst
If Not rst8.BOF Then rst8.MoveFirst
End Sub

Function RemoveDotAndHyphen(st3)
l1: dot1 = InStr(st3, ".")
    dot2 = InStrRev(st3, ".")
    If dot1 <> 0 And dot1 = dot2 Then 'only one dot is inside
        st3 = Left(st3, dot1 - 1) & Mid(st3, dot1 + 1)
    ElseIf dot1 <> 0 And dot1 <> dot2 Then
        l = Left(st3, dot1 - 1)
        m = Mid(st3, dot1 + 1, dot2 - dot1 - 1)
        r1 = Mid(st3, dot2 + 1)
        st3 = l & m & r1
        GoTo l1
    End If
l2: dash1 = InStr(st3, "(")
    dash2 = InStrRev(st3, "(")
    If dash1 <> 0 And dash1 = dash2 Then    ' one dash is only inside
        st3 = Left(st3, dash1 - 1) & Mid(st3, dash1 + 1)
    ElseIf dash1 <> 0 And dash1 <> dash2 Then
        l = Left(st3, dash1 - 1)
        m = Mid(st3, dash1 + 1, dash2 - dash1 - 1)
        r1 = Mid(st3, dash2 + 1)
        st3 = l & m & r1
        GoTo l2
    End If
 RemoveDotAndHyphen = st3
 End Function
```

```
Private Sub CalculateDf()
' rst8 holds the vector of the document to be classified
While Not rst8.EOF
df = 0
st = rst8!word
' check if the term has already document frequency information
rst5.FindFirst "word = '" & st & "'"
If Not rst5.NoMatch Then
 df = rst5!dfrequency + 1
 rst5.Edit
 rst5!dfrequency = df
 rst5.Update
 rst8.Edit
 rst8!dfrequency = df
 rst8.Update
Else
' otherwise create a document frequency information for the term from the new documents
‘term table
 rst3.FindFirst "word = '" & st & "'"
 If Not rst3.NoMatch Then
 While Not (rst3.NoMatch Or rst3.EOF)
   df = df + 1
   rst3.FindNext "word = '" & st & "'"
 Wend
 End If
 ' if the document frequency information calculated above satisfies the threshold for df register
‘the value else remove the term from the vector table
 If df > 1 Then
   rst5.AddNew
   rst5!word = st
   rst5!dfrequency = df
   rst5.Update
   rst8.Edit
   rst8!dfrequency = df
```

```
   rst8.Update
 Else
   rst8.Delete
 End If
End If
' move the pointers to the first record of tables processed above
rst8.MoveNext
rst3.MoveFirst
rst5.MoveFirst
Wend
If Not rst8.BOF Then rst8.MoveFirst
If Not rst5.BOF Then rst5.MoveFirst
If Not rst3.BOF Then rst3.MoveFirst
End Sub

Private Sub CalculateWeight()
' number of documents processed to produce the vector table from documents of the three
‘classes
cultdoc = 100
ecodoc = 850
accdoc = 210
If Not rst6.BOF Then rst6.MoveFirst
rst8.MoveFirst
' number of new documents processed so far after the generation of the vector table
newdoc = rst6!NewDocnumber
' total documents in the collection (all documents proccessed so far)
totdoc = culdoc + ecodoc + accdoc + newdoc
' calculate weight for each term in the document vector
While Not rst8.EOF
wrd = rst8!word
rst8.Edit
rst5.FindFirst "word = '" & wrd & "'"
rst8!Weight = rst8!frequency * Log(totdoc / rst5!dfrequency)
rst8.Update
```

```
rst8.MoveNext
If Not rst5.BOF Then rst5.MoveFirst
Wend
If Not rst8.BOF Then rst8.MoveFirst
If Not rst7.BOF Then rst7.MoveFirst
End Sub

Function RemoveSuffix1(st)
'This procedure stems the sufixes w, n, m, and na
l = Len(st)
If l - 1 <= 0 Then GoTo l1    ' if word length is 1, no processing is made
st2 = Left(st, l - 1)
l3 = Len(st2)
'check if diactric mark exists in the remaining word, which helps to determine the required
‘word length
d2 = InStr(1, st2, "#") + InStr(1, st2, "!") + InStr(1, st2, "@") _
  + InStr(1, st2, "Ö") + InStr(1, st2, "$")
If Not rst4.BOF Then rst4.MoveFirst
s = Right(st, 1)
rst4.FindFirst "suffix = '" & s & "'"
If Not rst4.NoMatch Then
    If (d2 > 0 And l - 1 > 2) Or (d2 = 0 And l - 1 > 1) Then st = st2
End If
l1: removesuffix1 = st
End Function

Function RemoveSuffix2(st)
'check the size before stripping
l = Len(st)
If l - 2 <= 0 Then GoTo l1
st2 = Left(st, l - 2)
d2 = InStr(1, st2, "#") + InStr(1, st2, "!") + InStr(1, st2, "@") _
    + InStr(1, st2, "Ö") + InStr(1, st2, "$")
If Not rst4.BOF Then rst4.MoveFirst
```

```
s = Right(st, 2)
rst4.FindFirst "suffix = '" & s & "'"
If Not rst4.NoMatch Then
   If (d2 > 0 And l - 2 > 2) Or (d2 = 0 And l - 2 > 1) Then
   l2 = rst4!number2strip
   st = Left(st, l - l2)
   found2 = True
   End If
End If
l1: removesuffix2 = st
End Function

Function RemoveSuffix3(st)
l = Len(st)
If l - 3 <= 0 Then GoTo l1
st2 = Left(st, l - 3)
l3 = Len(st2)
d2 = InStr(1, st2, "#") + InStr(1, st2, "!") + InStr(1, st2, "@") _
   + InStr(1, st2, "Ö") + InStr(1, st2, "$")
If Not rst4.BOF Then rst4.MoveFirst
s = Right(st, 3)
rst4.FindFirst "suffix = '" & s & "'"
If Not rst4.NoMatch Then
   If (d2 > 0 And l - 3 > 2) Or (d2 = 0 And l - 3 > 1) Then
   l2 = rst4!number2strip
   st = Left(st, l - l2)
   found3 = True
   End If
End If
l1: removesuffix3 = st
End Function

Function RemoveSuffix4(st)
l = Len(st)
```

```
If l - 4 <= 0 Then GoTo l1
st2 = Left(st, l - 4)
l3 = Len(st2)
d2 = InStr(1, st2, "#") + InStr(1, st2, "!") + InStr(1, st2, "@") _
   + InStr(1, st2, "Ö") + InStr(1, st2, "$")
s = Right(st, 4)
rst4.FindFirst "suffix = '" & s & "'"
If Not rst4.NoMatch Then
   If (d2 > 0 And l - 4 > 2) Or (d2 = 0 And l - 4 > 1) Then
   l2 = rst4!number2strip
   st = Left(st, l - l2)
   found4 = True
   End If
End If
l1: removesuffix4 = st
End Function

Function RemoveSuffix5(st)
l = Len(st)
If l - 5 <= 0 Then GoTo l1
st2 = Left(st, l - 5)
l3 = Len(st2)
d2 = InStr(1, st2, "#") + InStr(1, st2, "!") + InStr(1, st2, "@") _
   + InStr(1, st2, "Ö") + InStr(1, st2, "$")
s = Right(st, 5)
rst4.FindFirst "suffix = '" & s & "'"
If Not rst4.NoMatch Then
   If (d2 > 0 And l - 5 > 2) Or (d2 = 0 And l - 5 > 1) Then
   l2 = rst4!number2strip
   st = Left(st, l - l2)
   found5 = True
   End If
End If
l1: removesuffix5 = st
```

```
End Function

Function RemovePreffix(st)
'This function stems the sufix woch, wochu and their combination with n, m, na
a = Mid(st, 2, 1)
b = Mid(st, 3, 1)
c = Mid(st, 4, 1)
d = a = "#" Or a = "!" Or a = "@" Or a = "Ö"
e = b = "#" Or b = "!" Or b = "@" Or b = "Ö"
f = c = "#" Or c = "!" Or c = "@" Or c = "Ö"
d2 = InStr(2, st, "#") + InStr(2, st, "!") + InStr(2, st, "@") _
   + InStr(2, st, "Ö") + InStr(2, st, "E") + InStr(2, st, "$")
e2 = InStr(3, st, "#") + InStr(2, st, "!") + InStr(2, st, "@") _
   + InStr(2, st, "Ö") + InStr(2, st, "E") + InStr(2, st, "$")
f2 = InStr(4, st, "#") + InStr(2, st, "!") + InStr(2, st, "@") _
   + InStr(2, st, "Ö") + InStr(2, st, "E") + InStr(2, st, "$")
g = Len(st)
  If (Left(st, 1)) = "b" And Not d Then
     If d2 > 0 and Len(st) > 3 Then st = Right(st, g - 1) 'be
     Else If Len(st) > 2 Then st = Right(st, g - 1)
     End If
  ElseIf (Left(st, 1)) = "y" And Not d Then
     If d2 > 0 and Len(st) > 3 Then st = Right(st, g - 1) 'be
     Else If Len(st) > 2 Then st = Right(st, g - 1)
     End If
  ElseIf (Left(st, 1)) = "k" And Not d Then 'be
     If d2 > 0 and Len(st) > 3 Then st = Right(st, g - 1) 'be
     Else If Len(st) > 2 Then st = Right(st, g - 1)
     End If
  ElseIf (Left(st, 1)) = "l" And Not d Then 'be
     If d2 > 0 and Len(st) > 3 Then st = Right(st, g - 1) 'be
     Else If Len(st) > 2 Then st = Right(st, g - 1)
     End If
  ElseIf (Left(st, 2)) = "Sl" And Not e Then 'be
```

```
    If e2 > 0 and If Len(st) > 4 Then st = Right(st, g - 2)
    Else If Len(st) > 3 Then st = Right(st, g - 2)
    End If
  ElseIf (Left(st, 3)) = "XNd" And Not f Then
    If f2 > 0 and Len(st) > 5 Then st = Right(st, g - 3)
    Else If Len(st) > 4 Then st = Right(st, g - 3)
    End If
  End If
  removepreffix = st
End Function

Function Change2CommonForm1(st)
'This function converts the different ha forms into the first form
a = InStr(st, "/") 'used for hamerewuha 1st form
b = InStr(st, "^") '4th form of hamerewuha
c = InStr(st, "¦") '4th form of ha the first
d = InStr(st, "^") 'used for 4th form of hamerewuha
e = InStr(st, "?") '6th form of hamerewuha
f = InStr(st, "‡") '7th form of hamerewuha
g = InStr(st, "^") 'used for hailesilasie ha 1st and 2nd form
h = InStr(st, "`") 'used for hailesilasie 4th form
i = InStr(st, "~") 'used for hailesilasie 6th form
j = InStr(st, "@") 'diactric for 5th form of hamerewuha and hailesilasie ha.

If a > 0 Then   '
  If Mid(st, a + 1, 1) = "!" Then
     st = Left(st, a - 1) & "£" & Mid(st, a + 2) 'change hi
  ElseIf Mid(st, a + 1, 1) = "@" Then
     st = Left(st, a - 1) & "ÿ" & Mid(st, a + 2) 'change hie
  Else
     st = Left(st, a - 1) & "h" & Mid(st, a + 1) 'change ha and hu
  End If
ElseIf d > 0 Then
   st = Left(st, d - 1) & "h" & Mid(st, d + 1) 'change ha (z 4th)
```

```
ElseIf e > 0 Then
   st = Left(st, e - 1) & "H" & Mid(st, e + 1) 'change h
ElseIf f > 0 Then
   st = Left(st, f - 1) & "ç" & Mid(st, f + 1) 'change ho
ElseIf g > 0 Then
  If Mid(st, g + 1, 1) = "!" Then
     st = Left(st, g - 1) & "£" & Mid(st, g + 2) 'change hi
  ElseIf Mid(st, g + 1, 1) = "@" Then
     st = Left(st, g - 1) & "ÿ" & Mid(st, g + 2) 'change hie
  ElseIf Mid(st, g + 1, 1) = "Ö" Then
     st = Left(st, g - 1) & "ç" & Mid(st, g + 2) 'change hie
  Else
     st = Left(st, g - 1) & "h" & Mid(st, g + 1) 'change ha and hu
  End If
ElseIf h > 0 Then
   st = Left(st, h - 1) & "h" & Mid(st, h + 1) 'change ha (z 4th)
ElseIf i > 0 Then
   st = Left(st, i - 1) & "H" & Mid(st, i + 1) 'change h
ElseIf c > 0 Then
   st = Left(st, c - 1) & "h" & Mid(st, c + 1) 'change ha
End If
change2commonform1 = st
End Function

Function Change2CommonForm2(st)
'This function converts the different se forms into the first form
a = InStr(st, "\") 'used 4 nigusu se 1st & 2nd form
b = InStr(st, "œ") ' 3rd form of nigusu se
e = InStr(st, "|") '6th form of nigusu se
f = InStr(st, "ƒ") '7th form of nigusu se

If a > 0 Then   'su don't have pb. changing se will change it
   st = Left(st, a - 1) & "s" & Mid(st, a + 1) 'change se and su
ElseIf b > 0 Then
```

```
    If Mid(st, b + 1) = "!" Or Mid(st, b + 1) = "@" Then
      st = Left(st, b - 1) & "s" & Mid(st, b + 1) 'change  si, sie
    Else
       st = Left(st, b - 1) & "ú" & Mid(st, b + 1) 'change  sa
    End If
ElseIf e > 0 Then
    st = Left(st, e - 1) & "S" & Mid(st, e + 1) 'change  (z 4th)
ElseIf f > 0 Then
    st = Left(st, f - 1) & "î" & Mid(st, f + 1) 'change  (z 4th)
End If
change2commonform2 = st
End Function

Function Change2CommonForm3(st)
'This function converts the different aa forms into the first form
a = InStr(st, ";") 'used for ayinu aa 1st & 2nd form
b = InStr(st, "›") ' 4th form of ayinu aa
e = InStr(st, ":") '6th form of ayinu aa
f = InStr(st, "â") '7th form of ayinu aa
i = InStr(st, "¨") '4th form of the first aa
If a > 0 Then   'ou don't have problem changing aa will change it
    st = Left(st, a - 1) & "x" & Mid(st, a + 1) 'change aa 1st and 2nd
ElseIf b > 0 Then
    If Mid(st, b + 1) = "!" Or Mid(st, b + 1) = "@" Then
      st = Left(st, b - 1) & "x" & Mid(st, b + 1)
    Else
       st = Left(st, b - 1) & "¨" & Mid(st, b + 1)
    End If
ElseIf e > 0 Then
    st = Left(st, e - 1) & "X" & Mid(st, e + 1)
ElseIf f > 0 Then
    st = Left(st, f - 1) & "å" & Mid(st, f + 1)
ElseIf i > 0 Then
     st = Left(st, i - 1) & "x" & Mid(st, i + 1)
```

```
End If
change2commonform3 = st
End Function

Function Change2CommonForm4(st)
'This function converts the different tse forms into the first form
a = InStr(st, "]") 'used 4 ste 1st & 2nd form
b = InStr(st, "i") 'used 4 ste 1st & 2nd form in another code
c = InStr(st, "É") ' 4th form of ste
d = InStr(st, "}") '6th form of ste
e = InStr(st, "ò") '7th form of ste
If a > 0 Then   'su don't have pb. changing se will change it
   st = Left(st, a - 1) & "o" & Mid(st, a + 1)
ElseIf b > 0 Then
   st = Left(st, b - 1) & "o" & Mid(st, b + 1)
ElseIf c > 0 Then
   If Mid(st, c + 1) = "!" Or Mid(st, c + 1) = "@" Then
     st = Left(st, c - 1) & "o" & Mid(st, c + 1)
   Else
      st = Left(st, b - 1) & "Ú" & Mid(st, b + 1)
   End If
ElseIf d > 0 Then
   st = Left(st, d - 1) & "A" & Mid(st, d + 1)
ElseIf d > 0 Then
   st = Left(st, d - 1) & "Û" & Mid(st, d + 1)
End If
change2commonform4 = st
End Function

Function Change2CommonForm5(st)
'this function converts the 6th for of w to its 2nd form
e = InStr(st, "W")
If a > 0 Then
st = Left(st, a - 1) & "ý" & Mid(st, a + 1)
```

```
End If
change2commonform5 = st
End Function

Function IsValid(st)
' this funciton checks the validity of Amharic word
l = Len(st)
isvalid = True
For i = 1 To l
If IsNumeric(Mid(st, i, 1)) Then
   isvalid = False
   Exit For
End If
Next
End Function
```

Annex 6: Sample Vector Table

Word	WeightInClassA	WeightInClassC	WeightInClassE
ጢንዚዛ	0.4444896	0	0.5778365
ጢንዝዛ	0	0	0.2014329
ጢስ	0	0	0.5984692
ጤና	0.2823364	0.2823364	4.235047
ጤናማ	0.3111427	0	0
ጤፍ	0	0	2.313637
ጤንነቱ	0	0	0.0968913
ጸድቀ	0	0	0.09207219
ጸሀፊ	0	0.1245869	0.08305793
ጸሀይ	0	0	0.3111427
ጸጋዬ	0	0	0.08889792
ጥጥ	0	0	0.5501204
ጥሩ	0	0	0.1523255
ጥላሁ	0	0.5447848	0.08381305
ጥሪ	0	0.2295564	0.1147782
ጥራጥሬ	0	0	0.3999395
ጥራታቸ	0	0	0.2629146
ጥራት	0	0	2.658606
ጥራቱ	0	0	0.2415814
ጥናታዊ	0	0.4272149	0.07767543
ጥናቶች	0	0	0.07998791
ጥበብ	0	0.3623721	0.08052713
ጥብቅ	0.08235879	0.1235382	0.08235879
ጥበ	0.3204896	0.1602448	1.922937
ጥድ	0	0	0.0968913
ጥሬ	0.07426964	0	0.5570223
ጥገ	0	0.1369475	1.985738
ጥጆች	0.09417732	0	0.141266
ጥፋት	0.1610543	0	0.1610543
ጥልቅ	0	0	0.07947784
ጥልቀት	0.07727393	0	0.1545479
ጥንታዊ	0	0.3737606	0.3322317

Word	WeightInClassA	WeightInClassC	WeightInClassE
ጥንታዊት	0	0.1007164	0
ጥንቁ	0.4188952	0	0.3046511
ጥንዚዛ	0.09207219	0	0.2301805
ጥቆማ	0.07998791	0	0.3999395
ጥቁር	0	0	0.6151105
ጥቅማቸ	0	0	0.0968913
ጥር	0.07688881	0	0
ጥርጣሬ	0	0	0.1510747
ጥርጊያ	0	0	0.2825319
ጥረት	0.170299	0.2838317	2.15712
ጥት	0	0.08652703	0.08652703
ጥይት	0.1777958	0	0
ጽጌረዳ	0	0	0.1007164
ጽሁፎች	0	0.2539017	0
ጽሁፍ	0	0.2415814	0.08052713
ጎጃ	0.07727393	0	0.5795544
ጎጂ	0.2354433	0	0
ጎራ	0	0.2710565	0
ጎዴ	0	0	0.2354433
ጎዳ	0.08109907	0.1621981	0.6893421
ጎብኚ	0	0	0.08889792
ጎች	0	0	0.3893716
ጎሬ	0	0	0.1007164
ጎግ	0	0	0.0968913
ጎጆ	0.2399637	0.2799577	0.5199214
ጎጅ	0	0.2014329	0
ጎመ	0	0.1269509	1.015607
ጎመንዘር	0	0	0.0968913
ጎን	0.1485393	0	1.039775
ጎኑ	0	0	0.09417732
ጎንደር	0.06945053	0	0.8334063
ጎረቤት	0	0	0.08463391
ጎርፍ	0.6755616	0.07947784	0.2781724
ጎርፉ	0.1510747	0	0

Word	WeightInClassA	WeightInClassC	WeightInClassE
ኅተራ	0	0	0.1007164
ኅሳ	0	0.8286497	0
ሚኅዳ	0	0	0.0968913
ሚላከ	0	0	0.3028446
ሚታወቅ	0	0	0.1553509
ሚታይ	0.07199453	0	0.07199453
ሚታየባቸ	0	0	0.08763819
ሚካኤል	0	0	0.1661159
ሚካሂዱት	0	0	0.1634158
ሚካሂደ	0.06201226	0.2790552	1.36427
ሚካሂድባቸ	0	0	0.09035218
ሚሹ	0	0.1007164	0
ሚሸጥበት	0	0	0.1007164
ሚሸፍ	0.1165131	0	0.07767543
ሚሸፍነ	0	0	0.08553306
ሚዛ	0	0	0.5398765
ሚዛኖች	0	0	0.2014329
ሚና	0	0	0.1530376
ሚያካሂዱት	0.1333469	0	0
ሚያካሂድ	0	0	0.07899394
ሚያጠናቅቁ	0	0	0.1007164
ሚያደርገ	0	0	0.403867
ሚያደርስ	0	0.08305793	0
ሚባክ	0	0	0.1841444
ሚባሉት	0	0	0.08652703
ሚያገናኝ	0	0	0.09035218
ሚያገናኙ	0	0	0.09417732
ሚያገለግለ	0	0	0.0765188
ሚያገኙ	0	0	0.2415814
ሚያገኙበት	0	0	0.08381305
ሚያገኙት	0	0	0.07998791
ሚያገኟ	0	0	0.08553306
ሚያቋቁ	0	0	0.1007164
ሚያከናውናቸው	0	0	0.08889792

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሚያከናዉኑት	0	0	0.0968913
ሚያመለክት	0	0.07616276	0.1523255
ሚያመርታቸ	0	0	0.2354433
ሚያመርት	0	0	0.08652703
ሚያንቀሳቅሰ	0	0	0.1007164
ሚያቁሩ	0	0	0.0968913
ሚያቀርብ	0	0	0.1599758
ሚያቀርቡ	0	0.08553306	0
ሚያቀርቡት	0	0	0.08381305
ሚያስችላቸ	0	0	0.2764788
ሚያስችል	0.1237152	0	1.268081
ሚያስችሉ	0.1094081	0	0.1458774
ሚያስፈልግ	0.06847373	0	0.787448
ሚያስፈልጋቸ	0.1257196	0	0
ሚያስገኟ	0	0	0.09035218
ሚያስከፍለ	0	0	0.1007164
ሚያስከትለ	0	0	0.1570673
ሚያስከትሉት	0	0	0.1333469
ሚያስቀምጡት	0	0	0.0968913
ሚያስተዳድራቸ	0	0	0.0968913
ሚያጋጥ	0	0.1269509	0
ሚያጠቻ	0	0	0.08553306
ሚያሳድረ	0	0	0.09035218
ሚያሳይ	0	0.1807044	0
ሚያዝያ	0	0	0.1752764
ሚበልጡ	0	0	0.2771222
ሚበልጥ	0.1515907	0	1.728134
ሚጠበቅ	0	0	0.5777749
ሚጠብቁ	0	0	0.1841444
ሚጠበቅበት	0	0	0.1883546
ሚችል	0.1270634	0	1.111805
ሚችሉ	0	0.06802081	0.7142186
ሚችሉበት	0	0	0.08305793
ሚደረገ	0.06371669	0.1274334	0.8601753

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሚደረጉ	0	0	0.08553306
ሚደረግለት	0	0	0.1007164
ሚደርሰ	0	0	0.2829392
ሚደርሱ	0.1153332	0	0
ሚደርስባቸ	0	0	0.09207219
ሚጣለ	0	0	0.1007164
ሚፈልገ	0	0	0.07455994
ሚፈልጉ	0	0	0.1579879
ሚፈልጉት	0	0.09207219	0
ሚፈለግባቸ	0	0	0.1752764
ሚፈጸ	0	0	0.08553306
ሚገባቸ	0	0	0.1257196
ሚገቡ	0	0	0.3706146
ሚገቡት	0	0	0.141266
ሚገድብ	0	0.1007164	0
ሚገለገሉባቸ	0	0	0.0968913
ሚገመት	0.5796385	0	0
ሚገኙበት	0	0	0.07293871
ሚገቢ	0.06274142	0	0.6901555
ሚገቢበት	0	0	0.08235879
ሚጠፋ	0.09417732	0	0
ሚቋቋ	0	0	0.1297905
ሚቋቋሙ	0	0	0.1710661
ሚከናወ	0	0.07809479	0
ሚከናወነ	0	0	0.08889792
ሚከናወኑ	0	0	0.07899394
ሚከናወኑት	0	0	0.2163176
ሚከበር	0	0.09207219	0
ሚከፋፈል	0.08553306	0	0
ሚከለክል	0	0	0.09417732
ሚሊዩ	0	0	0.09207219
ሚሊየ	0	0	0.5778365
ሚሌ	0	0	0.2190955
ሚልዮ	0.07177255	0	2.440267

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሚለማ	0	0	0.08763819
ሚመለከታቸ	0	0	0.2813987
ሚመረት	0	0	0.1599758
ሚመረቱ	0	0	0.1007164
ሚመረቱት	0	0	0.1007164
ሚኖሩ	0.1530376	0	0.1530376
ሚነገረ	0	0.1007164	0
ሚነቀሳቀስ	0	0	0.08109907
ሚንቀሳቀሱ	0	0	0.07853365
ሚኖረ	0	0	0.2829392
ሚኖርባቸ	0	0	0.09207219
ሚኖርበት	0	0	0.1692678
ሚቆጠሩ	0	0.08763819	0
ሚቻለ	0	0	0.3959699
ሚቻልበት	0	0	0.1599758
ሚቆጠር	0	0	0.1355283
ሚጠቁ	0	0	0.08763819
ሚቀጥሉ	0	0	0.09417732
ሚቀጥሉት	0	0.06928055	0.4503236
ሚቀንስ	0	0	0.1777958
ሚጠቀሙት	0	0	0.1453369
ሚቀርብ	0	0	0.5857109
ሚረዳ	0	0	0.1132329
ሚረዱ	0	0	0.3199516
ሚስ	0	0	0.0968913
ሚሰጠ	0	0	0.3599727
ሚስጢር	0	0.0968913	0
ሚሰጡ	0	0	0.2318218
ሚሰጡት	0	0	0.08052713
ሚሰጥ	0	0.06911357	0.4146814
ሚሰሩ	0	0	0.2255064
ሚሰማሩ	0	0	0.3243963
ሚሰራ	0	0	0.3046511
ሚሰጣቸ	0	0	0.08052713

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሚስት	0	0.09035218	0
ሚስቱ	0	0.2906739	0
ሚጠጋ	0	0	0.2888874
ሚሳተፉ	0	0	0.1545479
ሚሳተፉበት	0	0	0.0968913
ሚንንዙ	0	0	0.1007164
ሚወጣ	0	0	0.3199516
ሚወስደ	0	0	0.3107017
ሚወሰነ	0	0	0.09035218
ሚዉል	0.0707348	0	0.3183066
ሚዉሉ	0.07767543	0.07767543	0.1165131
ሚጠይቀ	0	0	0.07899394
ሚጠየቁ	0	0	0.2415814
ሚጠይቁት	0	0	0.09417732
ሚይዘ	0	0	0.08889792
ሚይዙ	0	0	0.1510747
ሚዘሩ	0	0	0.1883546
ኋላፊ	0.08652703	0.08652703	0.5191622
ሩላ	0	0.1007164	0
ሩብ	0	0	0.1570673
ሩዋንዳ	0	1.035951	0
ሩሲያ	0.09035218	0.3614087	0
ሩዝ	0	0.1777958	0.8445302
ቮልት	0	0	0.7228174
ፈጣሪ	0	0.1561896	1.171422
ፈጥጨሩ	0	0.5078034	0.3385356
ፈኖ	0.08763819	0	0
ፈንቅላቱ	0.1007164	0	0
ፈነት	1.101732	0	0.2937951
ጨሩ	0.1777958	0	0.08889792
ጨፌ	0	0	0.09035218
ጨረታ	0	0.1794314	1.686655
ጨርሶ	0	0.1937826	0
ጨርቅ	0	0	0.3943719

Word	WeightInClassA	WeightInClassC	WeightInClassE
ጨር፣	0	0	0.09035218
ጨመ	0.3943719	0	0.2190955
ሸጠ	0	0	0.1007164
ሸሚያ	0	0	0.0968913
ሸማግሌ	0.09207219	0	0
ሸታ	0.4384178	0.1686222	2.697955
ሸያጭ	0	0	1.244885
ሸያጩ	0	0	0.1297905
ሸብሩ	0	0	0.0968913
ሸሮ	0	0	0.2825319
ሸዋ	0	0.08652703	0
ሸፈራ	0	0.1355283	0
ሸፋ	0.1178005	0	0.3141346
ሸቦ	0.1007164	0	0.1007164
ሸፋኑ	0	0	0.1314573
ሸልማት	0	0.4152896	0.2491738
ሸመ	0	0	0.1510747
ሸምብራ	0	0	1.368529
ሸመልስ	0.08889792	0	0.1777958
ሸንኩርት	0	0	0.08652703
ሸንሌ	0.1453369	0	0
ፖኬጆች	0	0	0.1007164
ፖኬጅ	0	0	0.2014329
ፖሊዮ	0.1007164	0	0.1007164
ፖሊሶች	0	0.1453369	0
ፖሊስ	6.806531	0.4235173	1.028542
ፖሊሲ	0	0.2110491	1.02007
ፖለቲካ	0	0.08889792	0
ፖምፕ	0	0	0.0968913
ፖስታ	0	0.2014329	0
ሚላ	0.1007164	0	0
ሚታ	0	0	0.09417732
ሚዳ	0	0	0.07998791
ሚዲካል	0.1007164	0	0

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሜክሲኮ	0	0	0.1007164
ሜትሪክ	0.2565992	0	0.8125641
ሜትር	0.4695544	0.08804144	7.072663
ሚጋ	0	0	0.5778365
ሚጋዋት	0	0	0.1007164
ዮናስ	0	0	0.1007164
ዮሀንስ	0	0	0.1647176
ኮታ	0	0	0.1453369
ኮካኮላ	0	0	0.3021493
ኮሪያ	0	0	0.1381083
ኮሞሽ	0.0968913	0	0
ኮድ	0.4473597	0	0
ኮፊ	0	0.09417732	0
ኮከብ	0.0968913	0	0
ኮኮኒ	0	0	0.2422282
ኮሌጁ	0	0	0.4844565
ኮሌጅ	0	0	0.1333469
ኮልፌ	0	0	0.09417732
ኮምቦልቻ	0.1178005	0.07853365	0.7460697
ኮንጎ	0	0.1841444	0.09207219
ኮንታ	0.09035218	0	0.09035218
ኮንሶ	0	0	0.5191622
ኮንክሪት	0	0	0.4237979
ኮሙኒስት	0	0.0968913	0
ኮንስትራክሽ	0	0	0.5964795
ኮንቴነር	0	0	0.2517911
ኮንቴይነር	0	0	0.1007164
ኮንትራት	0	0	0.1297905
ኮንትሮባንድ	0	0	1.621981
ኮርፖሬሽኑ	0	0	1.47034
ኮርማ	0	0	0.506397
ኮርፌሽኑ	0	0	0.1381083
ኮተቤ	0	0	0.48201
ኮትቻ	0	0	0.9064479

Word	WeightInClassA	WeightInClassC	WeightInClassE
ሂደት	0	0.1435451	0.1435451
ሂሳብ	0	0	0.8417068
ሩኬጅ	0	0	0.1453369
ማጥፊያ	0.1661159	0	0
ማጥፋት	0.07767543	0	0.1553509
ማኅ	0	0	0.2422282
ማኅልበት	0	0	0.227459
ማሽ	0	0	0.2666938
ማሽላ	0.147976	0	3.181485
ማሽከርከር	0.2027477	0	0
ማሽኑ	0	0	0.1510747
ማሽኖች	0	0	0.08889792
ማሽቆልቆል	0	0	0.09035218
ማላዊ	0.09417732	0	0.09417732
ማጂ	0	0	0.2514392
ማካሄድ	0	0	0.4240172
ማራባት	0	0	0.0968913
ማራቢያ	0	0	0.1883546
ማራቶ	0	0.4844565	0
ማጠራቀሚያ	0	0	0.4276653
ማኛ	0	0	0.6324652
ማሸግ	0	0	0.2517911
ማና	0	0.1937826	0.0968913
ማዳ	0	0.07548862	0.1509772
ማሟላት	0	0	0.227459
ማዳበሪያ	0	0.06979998	3.245699
ማናቸ	0	0	0.08889792
ማባዛት	0	0	0.7598851
ማባጉ	0	0	0.2595811
ማጠናከሪያ	0	0	0.2190955
ማጠናከር	0.06945053	0.1041758	0.5903295
ማዳቀል	0	0	0.4493936
ማጠናቀቂያ	0	0	0.1297905
ማዳረስ	0	0	0.1589557

Word	WeightInClassA	WeightInClassC	WeightInClassE
ማያዉቅ	0	0	0.0968913
ማበጠሪያ	0	0	0.2163176
ማበደር	0	0	0.08889792
ማበረታታት	0	0	0.07947784
ማበረታቻ	0	0	0.4085396
ማብት	0	0	0.0968913
ማደራጃ	0	0	0.1661159
ማደራጀት	0	0	0.3159758
ማደያ	0	0	0.5987314
ማደበሪያ	0	0	0.0968913
ማደግ	0	0	0.08652703
ማደጉ	0	0	0.1497189
ማድለብ	0	0	0.1634158
ማድረግ	0	0.461563	4.208366
ማድረጉ	0.06802081	0.06802081	0.2720833
ማድረቂያ	0	0	0.2014329
ማድረሱ	0.07947784	0	0.07947784
ማድረጋቸ	0.07485946	0	0
ማደስ	0	0	0.09035218
ማጣታቸ	0	0	0.08763819
ማጣሪያ	0	0	1.08957
ማጣራት	0	0	0.3322317
ማዋል	0.06945053	0	0.7292306
ማዋሉ	0	0	0.08553306
ማገናኘት	0	0.08889792	0
ማገዶ	0.1199819	0	0.5599153
ማገልገል	0	0	0.08763819
ማጉልበት	0	0	0.1883546
ማግኘታቸ	0	0	0.3713482
ማግኘት	0	0	0.8882797
ማግኘቱ	0	0	0.702886
ማግስት	0	0.08889792	0
ማህበሩ	0.1468976	0	2.570707
ማህበራዊ	0.1285889	0.06429446	0.9001226

Word	WeightInClassA	WeightInClassC	WeightInClassE
ማህበራት	0.1666279	0.06665118	3.39921
ማህበራቱ	0	0	0.3505528
ማህበረሰብ	0	0	0.4493935
ማሩጠ	0	0	0.2491738
ማዶ	0	0	0.0968913
ማቋቋ	0.1414696	0	0.6719805
ማከማቸት	0	0	0.1007164
ማከማቻ	0	0	0.09207219
ማከናወ	0	0	0.7419378
ማከናወኛ	0	0	0.08763819
ማከናወናቸ	0	0	0.1355283
ማከናወኑ	0	0	0.1561896
ማክበር	0	0.2258804	0.09035218
ማከሩፈያ	0	0	0.4493936
ማከሩፈል	0	0	0.1589557
ማል	0	0	0.09417732
ማልማት	0.06758865	0.06758865	1.858688
ማለዳ	0.1453369	0	0
ማልበስ	0	0	0.1841444
ማለፉ	0.07899394	0	0
ማመላለሻ	0.6823771	0	0.4170082
ማምለጡ	0.1007164	0	0
ማመንጨት	0	0	0.4143248
ማመንጫ	0	0	0.6229345
ማመቻቸት	0	0	0.07767543
ማምረቻ	0.1523255	0.2665697	1.10436
ማምረት	0	0	2.311203
ማምረቱ	0	0.09417732	0
ማኔጅመ	0	0	0.2301805
ማኔጅመንቱ	0	0	0.4603609
ማንነ	0	0	0.09417732
ማንኛ	0.07727393	0.07727393	0
ማነስ	0	0	0.2342844
ማንሻ	0	0	0.09417732

Word	WeightInClassA	WeightInClassC	WeightInClassE
ማንሳት	0	0	0.1355283
ማነሳሳት	0	0.1453369	0
ማቆ	0	0	0.3555917
ማዞር	0.0968913	0	0
ማቆየት	0	0	0.1710661
ማቀዱ	0	0	0.07581967
ማቅረባቸ	0	0	0.09207219
ማቅረብ	0	0	1.681175
ማቅረቡ	0	0	0.2514392
ማር	0	0	1.029485
ማረሚያ	0	0	0.1007164
ማርያ	0	0.08553306	0.08553306
ማርቼሎ	0	0	0.1007164
ማረፊያ	0	0	0.4983475
ማርክ	0.09035218	0	0.09035218
ማረቆ	0	0	0.0968913
ማረት	0.1007164	0	0
ማረፉ	0	0.1007164	0
ማረጋገጥ	0	0	0.1871486
ማረጋገጫ	0	0	0.2993657
ማረጋጋት	0	0	0.1752764
ማረየ	0	0	0.1007164
ማስታወቂያ	0	0.1474291	0
ማሰሪያ	0	0	0.09035218
ማሰራጨት	0	0	0.07616276
ማሰራጨቱ	0	0	0.08553306
ማሰራት	0	0	0.2163176
ማሰባሰብ	0	0	0.1963341
ማሰባሰቡ	0	0.09207219	0
ማስያዝ	0	0	0.2222448
ማሰብ	0	0	0.09207219
ማስደረግ	0	0	0.1007164
ማሰፈጸሚያ	0	0	0.1692678
ማስፈጸሚያ	0	0.08381305	0.2933457

Word	WeightInClassA	WeightInClassC	WeightInClassE
ማስገባት	0	0	0.2245784
ማስገባቱ	0	0	0.1883546
ማስገር	0	0	0.09417732
ማስፋፊያ	0	0	0.3293563
ማስፋፋት	0	0.06652454	0.8315567
ማስፋት	0	0	0.1225619
ማስኬጃ	0	0	0.09417732
ማሰልጠ	0	0	0.08305793
ማሰልጠኛ	0	0	0.3790984
ማስመጣት	0	0	0.08763819
ማሰመለስ	0	0.08381305	0
ማስመርመሪያ	0	0	0.1007164
ማስመርመር	0.1883546	0	0.09417732
ማስመሰል	0	0	0.1381083
ማስጠንቀቂያ	0	0	0.1245869
ማስቀረት	0.235601	0	0.3141346
ማስረጃ	0	0	0.3111427
ማስረከብ	0	0	0.09417732
ማሻሻያ	0	0	0.9436077
ማሻሻል	0	0	1.551808
ማስተላለፊያ	0	0	0.3067337
ማስተካከል	0	0	0.08889792
ማስተዳደር	0	0.09207219	0.1841444
ማሰተባበሪያ	0	0	0.2907028
ማስተናገጃ	0	0	0.1007164
ማስተናገድ	0	0	0.3999395
ማስተዋወቅ	0	0.07199453	0.539959
ማስተር	0.3067337	0	0.2190955
ማስወገጃ	0.08763819	0	0.08763819
ማስወገድ	0	0	0.4924478
ማተኮር	0	0	0.09417732
ማትረፍ	0.08889792	0	0
ማትሱራ	0	0.0968913	0
ማሳ	0.221964	0	1.257796

Word	WeightInClassA	WeightInClassC	WeightInClassE
ማሳያ	0	0	0.08235879
ማሳደግ	0	0	1.514028
ማሳደጊያ	0	0	0.1225619
ማጠል	0.08763819	0	0
ማሳለፍ	0	0.0968913	0
ማለል	0	0.07134365	0.3210464
ማሳተፍ	0	0.07581967	0.2653688
ማሳተፉ	0	0	0.1007164
ማንንዝ	0	0	0.1171422
ማንንዙ	0	0	0.1510747
ማሳየታቸ	0	0	0.0968913
ማሳየት	0	0	0.08652703
ማሳየቱ	0	0	0.1931848
ማወቅ	0	0.07054028	0.07054028
ማእድ	0	0	3.332736
ማእድናት	0	0	0.6572865
ማእከላዊ	0	0	1.74284
ማእከላት	0	0	0.7419004
ማእከላቱ	0	0	0.2354433
ማእከል	0	0.1937915	1.776422
ማእከሉ	0	0.2630908	3.006752
ማእቀፎች	0	0	0.0968913
ማእረግ	0.09207219	0	0
ማይጨ	0	0	0.09035218
ማይበልጥ	0	0	0.1676261
ማዉጣት	0.0751688	0	0.0751688
ማይከሮ	0	0.0968913	0
ማየት	0	0	0.1730541
ማዘጋጃ	0	0	1.056841
ማዘጋጀት	0	0.1122892	0.2994378
ማዘጋጀቱ	0	0	0.08553306
ጠላ	0.1510747	0	0
ላሚ	0	0	0.2014329