ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

Amharic Document Categorization Using Itemsets Method

By

Abraham Hailu Birkute

A thesis submitted to

The School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Computer Science

February 2013

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

Amharic Document Categorization Using Itemsets Method

By

Abraham Hailu Birkute

Signature of the Board of Examiners for Approval

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

DECLARATION

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

ABRAHAM HAILU BIRKUTE

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

YAREGAL ASSABIE (PHD)

ABSTRACT

Document categorization or document classification is the process of assigning a document to one or more classes or categories. Many researches are conducted in the area of Amharic document categorization. The main focus of those studies is to examine different document categorization techniques and measuring their performance however itemsets method is not so far examined. This study focused to extend Apriori algorithm which is traditionally used for the purpose of knowledge mining in the form of association rules.

The research focused on the basic principles of applying itemsets method to categorize Amharic documents. In addition to that the implementation of all the required tools which helps to carry out automatic Amharic Document categorization using itemsets method is developed and the algorithm is examined. Experiment results show itemsets method is an efficient method to categorize Amharic documents. The effectiveness and accuracy of the method to categorize Amharic documents is also evaluated and reported. Finally, factors affecting the performance of the proposed system and the importance of preprocessing training dataset in finding useful information are discussed.

DEDICATION

This thesis is dedicated to my best friend and partner, Wabi Wolde;

to my Dad and Mom, for understanding my goals;

to my spiritual father apostle Barnabas Paulos who has supported me throughout the process;

and to all of my friends who were helping me in many ways.

ACKNOWLEDGMENT

It would not have been possible to write this thesis without the help and support of the almighty God. I also would like to express my sincere gratitude to my advisor Dr. Yaregal Assabie for the continuous support of my MSc study and research, for his patience, motivation, interest, and immense knowledge.

Amongst my friends, I would also like to thank Wabi Wolde for his kindness, friendship and support.

Last, but by no means least, I thank my friends in Ethiopia, Norway, America and elsewhere for their support and encouragement throughout my work.

Abraham Hailu Birkute

TABLE OF CONTENTS

LIST OF TABLES, FIGURES, LISTING, AND GRAPH

Tables

Figures

Listing

Graph

LIST OF ANNEXES

ABBREVIATIONS USED

Abbreviations

TC	Text Categorization
EM	Expectation Maximization
SVM	Support Vector Machines
LMT	Logic Model Tree
LibSVM	Library of SVM
NB	Naive Bayes
ERTA	Ethiopian Radio and Television Agency
NMA	National Meteorology Agency
VOA	Voice of America
MoH	Ministry of Health
MoE	Ministry of Education
ECTM	Ethiopian Culture and Tourism Ministry
AACTB	Addis Ababa Culture and Tourism Bureau

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Due to the ongoing expansion of the Internet and electronic technologies, availability of electronic documents is dramatically increasing. The presence of large volume of electronic information requires automatic categorization of documents for the purpose of organizing the information, knowledge and pattern identification [26].

Manual Organization of very large volume of electronic information will not be feasible. To overcome this problem automatic document categorization is required.

A definition for text categorization can be found at Sebastini [29]: *"text categorization (TC- also known as text classification or topic spotting) is the task of automatically sorting a set of documents into categories (or classes, or topics) from a predefine set"*.

Hence document categorization is the process of assigning a document to one or more classes or categories. The documents to be categorized may be texts, images, music, etc. Each kind of document possesses its special categorization problems.

Documents may be categorized according to their subjects or according to other attributes (such as document type, author, printing year etc.) [45]. In this research only subject categorization of electronic text documents is considered. There are two main types of subject categorization of documents: The content based approach and the request based approach.

Content based categorization is categorization in which the weight given to particular subjects in a document determines the class to which the document is assigned.

Request oriented categorization (or indexing) is categorization in which the anticipated request from users is influencing how documents are being categorized.

Request oriented categorization is a kind of policy based categorization targeted towards a particular audience or user group. The categorization is done according to some ideals and reflects the purpose of doing the categorization.

Automatic document categorization tasks can be divided into two sorts [46]: supervised document categorization where some external mechanism (such as human feedback) provides

information on the correct categorization for documents, and unsupervised document categorization (also known as document clustering), where the categorization must be done entirely without reference to external information. There is also a semi-supervised document categorization, where parts of the documents are labeled by the external mechanism.

Automatic document categorization techniques [47] include Expectation maximization (EM), Naive Bayes categorizer, Latent semantic indexing, Support vector machines (SVM), Artificial neural network, K-nearest neighbour algorithms, Decision trees such as ID3 or C4, Concept Mining, Rough set based categorizer, Soft set based categorizer, Multiple-instance learning, and Natural language processing approaches.

Many researches are conducted in the area of Amharic document categorization [8, 9, 10].The main focus of those studies is to examine available document categorization techniques and measuring their performance.

Among those available techniques researchers used some of the methods to categorize Amharic document. For instance, Samuel Eyassu and Bjorn Gamback [8] used Artificial Neural Network to categorize Amharic documents. Their findings showed a precision of 60.0% when trying to cluster unseen data, and a 69.5% precision when trying to categorize it. Whereas to categorize Amharic documents Naïve Bayes and k-Nearest Neighbor categorizers techniques were used by Surafel Teklu [9]. His research findings showed that Naïve Bayes is more applicable to automatic categorization of Amharic documents. On the other hand Yohannes Afework [10] used two other techniques for categorization of Amharic documents namely Logic Model Tree (LMT) and Library of SVM (LibSVM). He stated that LMT and LibSVM categorizers showed good categorization accuracy; however, the computational cost was very high.

In this research, a different approach is introduced to categorize Amharic documents instead of using the existing and commonly used categorization approaches. The study focuses to extend Apriori Algorithm [2, 3, 4, 5] which is traditionally used for the purpose of knowledge mining in the form of association rules.

This thesis deals with Amharic document categorization using Apriori Algorithm. The Apriori algorithm was initially developed for data mining and basket analysis applications in the relational databases [1, 6, 7].

However in this research the algorithm is studied for the purpose of categorizing Amharic documents. This technique is a type of logic oriented approach [39] in which it relies on the statistical nature of the data which is going to be categorized.

This method of categorizing documents is preferred for the study because current researches findings [11, 13, 14] shows that the method is more applicable for short document (such as abstracts, summaries, news etc) categorization and also up to the knowledge of the researcher no researches are conducted to examine the performance of the method in Amharic document categorization yet. The research focused on the basic principles of applying itemsets method to categorize Amharic documents. In addition to that the implementation of all the required tools which helps to carry out automatic Amharic Document categorization using Apriori algorithm is developed. The effectiveness of the method to categorize Amharic documents is also studied, evaluated and reported.

1.2 STATEMENT OF THE PROBLEM

Automatic document categorization is very useful text mining method especially with the rapid growth of the number of available documents online [34]. In such cases, users to use their native language such as Amharic language without fear of inconvenience (e.g. as enough as English language) the presence of data mining tools which can aid them is very important. Doing so can help user in many ways such as searching, categorization and summarization of documents.

Performances [8, 9, 10] of currently examined text categorization methods don't show very high performance. This research focused on how Amharic documents can be automatically categorized by using Itemsets methods with high performance. By its nature automatic text categorization involves preprocessing of the source data so that the entire work involves the following two broad tasks; (1) Data preprocessing (data cleaning, data integration and transformation, data reduction, etc), (2) developing a prototype for automatic categorization of the preprocessed Amharic text.

1.3 JUSTIFICATION OF THE STUDY

The rapid growing phenomenon of the textual data needs text processing, text mining, machine learning and natural language processing techniques and methodologies to manage and dig out pattern and knowledge from the documents. Text processing is a crucial issue.

Several document categorization algorithms or group of algorithms as hybrid approaches are proposed for the automatics categorization of documents, among these SVM and NB categorizer are shown most appropriate in the existing literature [15]. However, it is believed that more researches are required for the performance improvement and accuracy of the documents categorization and new method to solutions are required for useful knowledge from the increasing volume of electronics documents.

1.4 APPLICATIONS OF THE STUDY

Some of the major applications of automatic document categorization are summarized as follows:

- Easy to use automatic categorization system for improved document organization
- Faster and more reliable than manual categorization of documents
- Possibility of processing large volumes of documents at a time
- Significantly decrease manually organizing documents
- Easily integrates new documents with the available document workflow.

1.5 OBJECTIVES OF THE STUDY

1.5.1 GENERAL OBJECTIVE

The general objective of this research is to investigate the means to extend application of Apriori Algorithm (which is traditionally used for the purpose of mining association rule) to automatically categorize Amharic documents.

1.5.2 SPECIFIC OBJECTIVES

The specific objectives of this research are:

- To review literature on the concept of Apriori algorithm and text categorization.

- To collect, preprocess Amharic documents to make them suitable for automatic categorization.
- To build and train models using Apriori algorithm.
- To test the performance of the model using a prototype that automatically categorizes documents according to content.

1.6 SIGNIFICANCE OF THE STUDY

The final findings of this research will have great contribution in the era of automatic Amharic document categorization. This is because; the findings will provide the opportunity for the user to have alternate approach for automatic Amharic document categorization. Also it opens a door for researchers to find research ideas about the possibilities of extending available data mining algorithms for categorization of Amharic documents.

1.7 SCOPE OF THE STUDY

This research mainly focused on the application of Apriori algorithm for the purpose of categorizing Amharic documents so that the research did not considered categorizing English or other languages documents. In this study only Amharic text documents are considered for categorization.

1.8 APPLICATION

Text categorization is the most frequently used approach to automated categorization. The main reason for document categorization is to access the desired document in a sophisticated manner so that in future the data or the document itself can be modified and retrieved without losing any information.

While a large portion of research is aimed at improving algorithm performance, it has been applied in operative information systems such as website content catalogue system.

Apart from organizing web pages contents into categories, text categorization has been applied for categorizing web search engine results. It also finds its application in document filtering, word sense disambiguation, speech categorization, multimedia document categorization, language identification, text type identification, and automated essay grading[6,9].

Moreover text categorization has many different applications. Indexing of scientific articles according to predefined tags or keywords of technical terms, automated population of hierarchical catalogues of web resources, spam filtering, categorization of news paper ads, categorizing news stories as sport, politics, technology and Economy [44].

1.9 METHODOLOGY

The methodologies which are used to achieve the objectives of the research are the following

1.9.1 Literature Review

Literature review has been conducted to understand text categorization techniques and particularly prior studies in the area of associating Apriori Algorithm with data categorization. In addition to that literature review also was conducted to understand Amharic text features [12] on the subject of computer representation and categorization.

1.9.2 Data Collection

As source of Amharic documents the following organizations are used:

- Ethiopian Radio and Television Agency (ERTA) website [16]
- Reporter Online Amharic Magazine [17]
- National Meteorology Agency (NMA) website [18]
- Ethiopian Orthodox Tewahedo Church Sunday Schools Department Mahibere Kidusan USA Center website [19]
- Deutsche Welle website [20]
- Voice of America (VOA) website [21]
- Ministry of Health (MoH) website [22]
- Ministry of Education website (MoE) [23]
- Ethiopian Culture and Tourism Ministry (ECTM) website [24]
- Addis Ababa Culture and Tourism Bureau (AACTB) website [25]

1.9.3 Program Development Tools

Initially, data mining software named Weka was used to categorize documents. However when the volume of the training documents increases the software was not able to function properly. Due to this The Apriori algorithm is implemented by using PHP and MySQL programming language and database system respectively. In addition to that Microsoft Office Excel 2007 is used.

1.9.4 Document Analysis

For further understanding of the training and test Amharic documents how Amharic texts were used was analyzed for all websites which were used as a source. And all the training as well as testing documents were converted to Unicode text format for consistency purpose.

1.9.5 Sampling Technique

The population of this study was a total of 637 training data (documents) and 348 test data out of the total 985 documents. The training data is approximately 2/3 of the total corpus and the rest, 1/3 hand used for testing purpose.

1.9.6 Experiment

Experiment was conducted both for preprocessing the raw documents and to examine categorization of the documents using itemsets method. To increase the performance of the model further experiments has been done. Moreover to demonstrate the application of the method for document categorization, Ethiopian Video Grabbing website [49] named "ሀሁ· Tube" is developed and how it works is explained.

1.9.7 Testing Technique

To measure the performance of The Apriori algorithm for automatic document categorization, a prototype is developed and tested for the effectiveness of categorizing the test documents.

1.10 ORGANIZATION OF THE THESIS

This thesis is organized in six chapters. Chapter one is a general introduction to the problem and the justification of the research and methodology used for the research. The main focus of chapter two is literature review mainly focused on automatic document categorization. It discusses concepts in automatic document categorization using itemsets method.

In chapter three, the characteristics of the Amharic language which are relevant to the research area are discussed briefly. The experimental settings, the process of the experimentation and the findings are presented in chapter four. In chapter five results are evaluated and factors which affect the performance of the model are discussed. Finally, in chapter six conclusions and recommendations are made based on observations and results from the experiment.

CHAPTER TWO

LITERATURE REVIEW

2.1 AUTOMATIC DOCUMENT CATEGORIZATION

There is a huge amount of information available on the World Wide Web. As the information flows in the WWW at a very high speed, there is a need to organize it in the right manner so that user can access it very easily. Currently most organizations and even search engines (e.g., Google [27], Yahoo [28]) categorize the documents manually [26].

With today's document processing technology, organizations do not need to rely on manual categorization or processing of documents. Some of the reasons why organizations that overcome manual document categorization in favor of an automated document categorization and processing system can realize a significant reduction in manual entry costs, and improve the speed and turnaround time for document processing.

Text categorization has many applications. Among those applications some of them are spam filtering, indexing of electronic documents, automatic online cataloging of web resources, categorizing news stories etc. The applications of automatic categorization also extend to fraud detection. [1,11,14,40]. This chapter discusses the basic steps of automatic Amharic document categorization.

Automatic document categorization tasks can be divided into three sorts[30]: supervised document categorization where some external mechanism (such as human feedback) provides information on the correct categorization for documents, unsupervised document categorization (also known as document clustering), where the categorization must be done entirely without reference to external information, and semi-supervised document categorization, where parts of the documents are labeled by the external mechanism.

2.2 PROCEDURES OF DOCUMENT CATEGORIZATION

The three main parts of document categorization processes are discussed in brief as follows;

(1) Initially it involves manual categorization of a number of documents to pre-defined categories. These documents are called training documents. This is because, based on those documents, the characteristics of categories in which documents belong to are learnt.

(2) After learning the characteristics of training documents, for each category a program called categorizer will be constructed. After the construction of the categorizers and before automated categorization of new documents takes place, categorizers will be tested with a set of test documents, which were not used in the first step.
(3) The third step is about applying the categorizer to new documents and predicting the category of new documents based on the categorizers.

2.3 APPROACHES TO AUTOMATIC DOCUMENT CATEGORIZATION

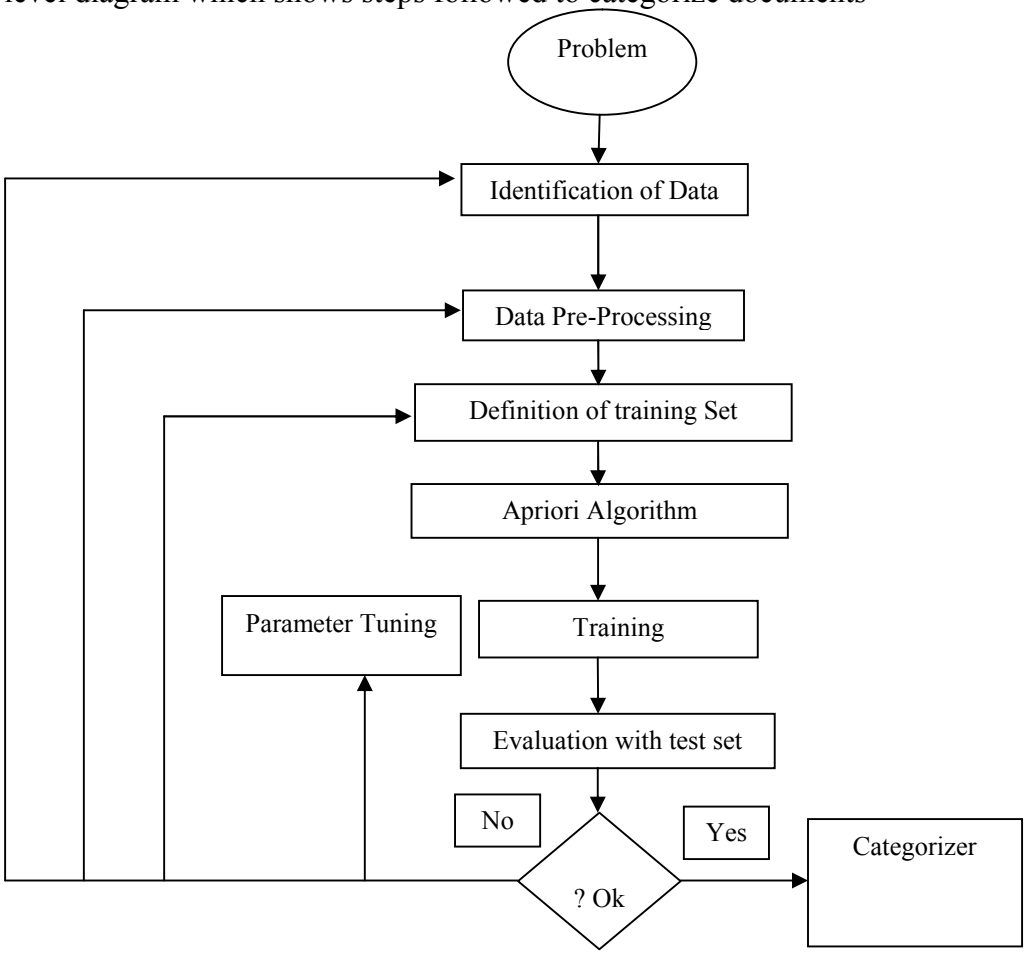
Text categorization is a machine-learning approach [29], in which also information retrieval methods are applied. It consists of three main parts. Those are;

(1) Categorizing a number of documents to pre-defined categories,
(2) Understanding the characteristics of those documents, and
(3) Categorizing new documents.

Using input dataset categorization models could be built using categorization techniques. This process requires a learning algorithm. After this using the learning algorithm building a model that best matches the relationship between item set (in this case word set) and categories of the input data. After training the model is supposed to predict the categories for new documents which are previously unknown. Figure 2.1 shows high level architecture adopted from [41] which shows the procedures followed for the document categorization.

For document categorization problem a corpus is required. This corpus which is considered as a data set is further divided into a training data set and a test data set. A training set is a collection of records which have categories already known. This set is used to build categorization model. Whereas a test set is a collection of records whose categories are known and the model used must predict categories for these known records. Through this approach accuracy of the categorization model will be measured. [13].

Fig 2.1 High level diagram which shows steps followed to categorize documents



Different terms can be extracted based on different principles. The number of terms per document needs to be reduced not only for indexing the document with most representative terms, but also for computing reasons. This is called dimensionality reduction of the term space.

Dimensionality reduction [50] methods could include removal of non-informative terms (not only stop words); also, taking only parts of the document, its summary has been explored.

To evaluate different aspects of text categorization performance various measures are used. While performing categorization, representative itemsets utilized to categorize documents into corresponding topics. The categorization algorithm is evaluated in terms of accuracy (precision, recall, and F1 parameters) and also efficiency is evaluated in terms of computing time spent on different parts of the process. Accuracy is measured by means of a test-set, the members of which have a priori known categorization.

Precision: P = p/q;

Recall: Q = p/r,

where p is the number of classes determined correctly by the categorizer (automatically); q is total number of classes determined automatically; r is the number of classes determined by a domain expert (manually, i.e. correctly). For every search result all retrievable items fall into any of the four cells in a matrix (Table 2.1) defined by the two characteristics [48]:

(i) Retrieved or Not Retrieved; and

(ii) Relevant or Not Relevant.

Table 2.1 Retrieval matrix

	Relevant	**Not Relevant**	**Total**
Retrieved	$N_{ret \cap rel}$	$N_{ret \cap \overline{rel}}$	N_{ret}
Not Retrieved	$N_{\overline{ret} \cap rel}$	$N_{\overline{ret} \cap \overline{rel}}$	$N_{\overline{ret}}$
Total	N_{rel}	$N_{\overline{rel}}$	N_{tot}

For any given retrieved set, Recall is the number of retrieved Relevant items as a proportion of all Relevant items i.e $N_{ret \cap rel}/N_{rel}$. Recall is, therefore, a measure of effectiveness in retrieving (or selecting) performance and can be viewed as a measure of effectiveness in including relevant items in the retrieved set. One hundred percent Recall can always be achieved by examining the entire database, but this may challenges the purpose of a retrieval system. High Recall may not be always needed because people commonly may not want all relevant items, often preferring only one or a few relevant items.

For a given retrieved set, Precision is the number of retrieved Relevant items as a proportion of the number of retrieved items i.e $N_{ret \cap rel}/N_{ret}$. Precision is, therefore, a measure of purity in retrieval performance, a measure of effectiveness in excluding non-relevant items from the retrieved set. The ideal Precision which can be achieved is 100%.

As discussed above there are many text categorization approaches. Some of them are:

1. Decision tree categorization
2. Neural networks
3. K nearest neighbor
4. Rule based categorization
5. Support vector machine
6. Bayesian categorization
7. Instance-based learning
8. Genetic algorithms
9. Apriori algorithm

A major difference among them is in how categorizers are built. Another difference within the text categorization approach [63] is in the document pre-processing and indexing part, where documents are represented as vectors of term weights. Calculating the term weights can be based on a variety of heuristic principles which is enabling a computer to discover or learn something for itself.

Among those mentioned categorization methods listed above, frequently used techniques [34,63] are discussed in brief as follows;

1. Decision tree categorization

   Fabrication of information using data mining techniques can be characterized in many different methods. In categorizing tasks, decision trees[62] helps to visualize what steps are taken to arrive at a categorization. Every decision tree begins with a root node. This root node considered as the "parent" of every other node. Each node in the tree evaluates an attribute in the data and determines which path it should follow. Usually, the decision test uses comparison of values against some constants. Categorization using a decision tree is performed by routing from the parent node until arriving at a leaf node.

2. Neural networks

   An artificial neural network [31] consists of a network of many simple units, usually positioned in successive layers. Communication channels that carry numeric data connect the units, with varying connection strengths. A network layer receives input, in the form of a collection of terms and weights representing a document, intermediate

layers process the weights, and an output layer suggests a relevant category. A large number of variations exist in this architecture.

3. Genetic algorithms

   Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of evolution.

4. Nearest neighbor method

   A technique that categorizes each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset. Sometimes called the k-nearest neighbor technique.

5. Apriori

   It is a model algorithm used in data mining for learning association rules. Apriori is designed to operate on databases containing transactions.

2.4 DOCUMENT CATEGORIZATION USING ITEMSET METHOD

2.4.1 Terminologies

Association rule can play an important role to categorize documents in many document categorization algorithms [13]. Association rules and the related concept of itemsets constitute the motivation for developing a new document categorizer. Common terminologies used in association rules are defined as follows [33].

Item: In text document categorization, each word contained in a document is referred as item.

Let I = (i1, i2, …, im) be a set of m distinct items.

Transaction: A transaction T is defined as any subset of items in I.

Database: A database D is a set of transactions.

Itemset: A set of items is called an Itemset.

Length of Itemset: is the number of items in the itemset. Itemsets of length ‘k’ are referred to as k-itemsets.

Minimum Support threshold: It is the minimum support threshold (minimum number of occurrence of a term) provided by the user, which is used as a standard to decide if an

itemset is sufficiently frequent to be determined as 'useful for the purpose of association rule mining.

Confidence: is defined as Support of $X \cup Y$ / Support of X.

Frequent Itemset: An itemset whose support count is greater than or equal to the minimum support threshold specified by the user.

Infrequent Itemset: An itemset, which is not frequent, is infrequent.

Association Rule: is the rule of the form $R : X \rightarrow Y$, where X and Y are two non-empty and non-intersecting itemsets. Support for rule R is defined as support of $X \cup Y$.

Mining association rules: consists of the following two stages:

- Stage 1: Finding all frequent itemsets.
- Stage 2: Generating association rules using the discovered frequent itemsets.

In the first stage at the beginning all candidate 1-itemsets shall be generated. Then by scanning the database, the support of each candidate 1-itemset should be found. Then all, which are frequent (having support greater than minimum support threshold) should be saved in a set. Then this set of frequent 1-itemsets is used to generate all possible candidate 2-itemsets. Again support of each is calculated and frequent nature is determined and process goes on, till no more candidate sets can be generated.

The cost of frequent itemset-discovery process comes from scanning of database and the generation of new candidate itemsets. According to Saxena[33], if it is possible to reduce database scanning time and time taken to generate new candidate itemsets needed by reducing the need of multiple scanning of database and number of candidates tested, it is possible to improve the performance of the mining process.

Interesting Association Rule: An association rule is said to be interesting if its support and confidence measures are equal to or greater than minimum support and confidence thresholds (specified by user) respectively.

Candidate Itemsets: It is a set of itemsets, which are to be tested to determine whether they are frequent or infrequent.

2.4.2 The Apriori Principle

The Apriori principle states that if an itemset is frequent, then all of its subsets must also be frequent [6]. For example suppose itemset {a, b, c} is a frequent itemset, then all of its subsets {a}, {b}, {c}, {a, b}, {a, c} and {b, c} are also frequent. This is because any transaction that contains {a, b, c} must also contain its subsets.

The Apriori algorithm is an efficient algorithm for knowledge mining in the form of association rules [11]. The algorithm is based on The Apriori principle, which states that every subset of a frequent item set is also frequent. First it calculates the support of every individual item by counting the instances that contain the item and then finds larger and larger frequent item sets, step by step.

The major steps in association rule mining are:

- Frequent Itemset generation
- Rules derivation

Apriori algorithm obeys the principle which says "every subset of a frequent itemset is also frequent". Hence it uses the downward closure property, to prune unnecessary branches for further consideration [51]. This rule essentially says that it is not required to find the count of an itemset, if all its subsets are not frequent. The support for an itemset never exceeds the support for its subsets.

Apriori algorithm needs two parameters, minimum Support and minimum confidence. The minimum support is used for generating frequent itemsets and minimum confidence is used for rule derivation [52].

Many researchers [1,2,11,13,33] recognized its convenience for document categorization. The original Apriori algorithm is applied to a transactional database of market baskets. In the context of a digital text documents, significant terms occurring in the text documents take place of items contained in market baskets and the transactional database is a set of documents (represented by sets of significant terms).

Apriori is a well known algorithm in data mining which was invented originally to analysis market basket transactions [1]. By extending the application of Apriori, this thesis work is

based on a basket of significant terms found from a collection of electronic Amharic documents.

This section shows how Apriori algorithm used to generate frequent itemsets by using a general transaction database example as shown in Table 2.2. The example is adopted from [1] Each row in the table represents a transaction, which contains distinctive transaction identification number (TID) along with items sold by a company represented as {A, B, C, D, E,F}.

Table 2.2 The transaction database

TID	Items
1	{A,B,D,E}
2	{A,B,C,D}
3	{A,C,D,E}
4	{B,C,D,F}
5	{A,C,D,F}
6	{B,C,D,F}

It is possible to represent the transaction database using binary form of 0's and 1's as shown in the Table 2.3. The rows correspond to transactions and columns correspond to an item. If an item exists in a transaction then it is represented as ‘1’ otherwise ‘0’ [42].

Table 2.3 A binary representation of transaction database

TID	A	B	C	D	E	F
Tr1	1	1	0	1	1	0
Tr2	1	1	1	1	0	0
Tr3	1	0	1	1	1	0
Tr4	0	1	1	1	0	1
Tr5	1	0	1	1	0	1
Tr6	0	1	1	1	0	1

Frequent itemset generation can be determined by Scanning D and count each itemset in C_k, if the count is greater than minSupp, then add that itemset to L_k.

Candidate itemset generation can be generated as follows [11]

For k = 1, C_1 = all itemsets of length = 1.

For k > 1, generate C_k from L_{k-1} as follows:

The join step:

C_k = k-2 way join of L_{k-1} with itself

If both $\{a_1,..,a_{k-2}, a_{k-1}\}$ & $\{a_1,.., a_{k-2}, a_k\}$ are in L_{k-1}, then add $\{a_1,..,a_{k-2}, a_{k-1}, a_k\}$ to C_k

The items are always stored in the sorted order (usually in ascending order).

The prune step:

Remove $\{a_1, \ldots,a_{k-2}, a_{k-1}, a_k\}$, if it contains a non-frequent (k-1) subset.

To generate Rules further steps are required. Frequent itemsets do no mean association rules. One more step is required to convert these frequent itemsets into rules.

Association Rules can be found from every frequent itemset X as follows:

For every non-empty subset A of X

Let B = X - A.

$A \Rightarrow B$ is an association rule if Confidence $(A \Rightarrow B) \geq$ minimum confidence.

Where;

confidence ($A \Rightarrow B$) = support (AB) / support (A), and

support($A \Rightarrow B$)=Support (BA)

One way to improve efficiency of the Apriori would be to prune without checking all k-1subsets.

The general steps of Apriori algorithm to find frequent itemsets is shown below [11].

Listing 2.1 The general steps of Apriori algorithm (pseudo code) to find frequent itemsets

1. Given a data set create a list of candidate item sets of size k=1, one item set for each item appearing in the data set.
2. For each candidate item set size k count the number of instances in the data set counting it. This is the support count of the item set.
3. Remove candidate item sets of size k not meeting the required minimum support count. The remaining item sets are the frequent item sets of size k.
4. Generate candidate item sets of size k+1 by joining together each pair of frequent item sets of size k that share k-1 item in common.
5. Prune candidate item sets that are known to be infrequent because of The Apriori principle.
6. Increment k by one.
7. Repeat steps 2 through 6 until there are no more frequent items or candidate item sets of size k.

The algorithm then generates association rules from the frequent itemsets. Association rules must meet the minimum confidence which is specified by the user. The confidence of an association rule is the probability of the predecessor given the subsequent.

Support counting which is found in the second step of the above pseudo code (Listing 2.1) is the process of determining the frequency of occurrence for all itemsets. Candidates that satisfy minimum support count are considered as frequent itemsets. When there are no more candidate itemsets to generate the algorithm will stop. The algorithm is illustrated with an example using transaction database as shown in Table 2.4 assuming that the minimum support count is 3.

According to the pseudo code the first step is listing each item in the transaction database as a candidate is shown in table 2.4.

Table 2.4 Candidate 1-Itemsets

Item
A
B
C
D
E
F

After generating candidate 1-itemsets their support counts are calculated. Only those itemsets are saved that satisfy the minimum support count known as frequent itemsets. Here itemset {E} is discarded and all the rest itemsets satisfy the minimum support count which is 3 as shown in Table 2.5.

Table 2.5 1-Itemsets those satisfied the minimum support

Item	Support Count
A	4
B	4
C	5
D	6
F	3

Candidate 2-itemsets are generated based on frequent 1-itemsets as shown in Table 2.6.

Table 2.6 Candidate 2-Itemsets

Itemsets
{A,B}
{A,C}
{A,D}
{A,F}
{B,C}
{B,D}
{B,F}
{C,D}
{C,F}
{D,F}

{A,E}, {B,E}, {C,E}, {D,E}, and {E,F} are pruned since those itemsets are not frequent.

Support count is calculated for each candidate 2-itemset and as {A, B}, {A, F}, and {B, F} are infrequent and all of its supersets are also infrequent from the property of support-based pruning. So, it is discarded and remaining itemsets are saved as frequent 2-itemsets. The frequent 2-itemsets are shown in Table 2.7.

Table 2.7 Support count for candidate 2-itemsets those satisfied the minimum support

Itemsets	Support Count
{A,C}	3
{A,D}	4
{B,C}	3
{B,D}	4
{C,D}	5
{C,F}	3
{D,F}	3

To generate candidate 3-itemsets apply step 4 of the pseudo code and generate candidate itemsets of size 3 by joining together each pair of frequent item sets of size 2 that share 1 item in common. The resulting candidate 3-itemsets are shown in Table 2.8.

Table 2.8 Candidate 3- itemsets

Itemsets	Support Count
{A,B,C}	?
{A,B,D}	?
{A,B,F}	?
{A,C,D}	?
{A,C,F}	?
{A,D,F}	?
{B,C,D}	?
{B,C,F}	?
{B,D,F}	?
{C,D,F}	?

During the candidate pruning step itemset {A, B, C} is eliminated because its subset {A, B} is infrequent. In the same way support count is calculated and frequent 3-itemsets are saved as shown in Table 2.9.

Table 2.9 Frequent 3- itemsets

Itemsets	Support Count
{A,C,D}	3
{B,C,D}	3
{C,D,F}	3

Candidate 4-itemsets are generated by checking whether 2 items of any 2 itemsets of frequent 3-itemsets are equal. Only two itemsets satisfy this property as shown in Table 2.10.

Table 2.10 Candidate 4- itemsets

Itemsets	Support Count
{A,B,C,D}	?
{A,C,D,F}	?
{B,C,D,F}	?

In the candidate pruning step all are eliminated because for itemset {A, B, C, D} its subset {A, B} is not frequent. In the same way for itemset {A, C, D,F} its subset {A, C,F} is not frequent and also for itemset {B, C, D,F} its subset { B,D,F} is not frequent. Because of this no frequent 4-itemsets are generated hence the algorithm halts.

In this research, the same technique is used to determine the frequent itemsets.

CHAPTER THREE

AMHARIC DOCUMENT CATEGORIZATION

3.1 INTRODUCTION

Vast amount of textual information is currently available in electronic form. This large volume of textual data has led to the task of mining useful or interesting frequent itemsets (words/terms) from very large text databases and still it seems to be quite challenging [53]. The use of frequent itemsets for text categorization has received a great deal of attention in research community since the mined frequent itemsets reduce the dimensionality of the documents drastically [3,4,50]. This chapter will discuss the basic principle of Amharic document categorization using itemsets method.

Since Amharic is work language of Ethiopia, resent days many organizations prefer to produce Amharic documents. The focus of this thesis is that how to develop a system which can automatically categorize Amharic documents using itemsets method.

3.2 AN EFFICIENT APPROACH FOR AMHARIC DOCUMENT CATEGORIZING BASED ON FREQUENT ITEMSETS

The huge quantity of documents existing in electronic form has motivated the exploration for hidden knowledge in text collections. Therefore, there is an increasing research concentration in the topic of text mining. For grouping text documents, researchers have used various data mining techniques and approaches, in which categorization is one of the popular technique. Text categorization is to group a collection of documents into different category groups so that documents in the same category group describe the same subject.

Many researches [8,9,10,34] have examined possible ways to enhance the performance of Amharic text document categorizing based on the popular categorization algorithms and frequent term based categorization. Here, in this research an effective approach for categorizing a text corpus with the aid of frequent itemsets is devised. The devised approach consists of the following major steps:

1) Preprocessing Amharic text :
   - Abbreviations have been expanded;
   - Foreign words have been eliminated;
   - Non Amharic characters and symbols have been removed; and
   - Words have been reduced to root word form;
2) Mining of frequent itemsets
3) Categorizing the text documents based on frequent Itemsets

3.3 AMHARIC DOCUMENT STRUCTURE

Documents from different sources have been used in this research. Based on the categories which were set by each documents sources the training documents have be used to develop categorizers.

The following table (Table 3.1) shows what possible specific major pre-defined categories of documents have been available on selected websites (accessed in between December 11, 2011 and October 18, 2012) and used for the experiment from each of the sources.

Table 3.1 List of Amharic text sources with accessed pre-defined categories of documents

No	Source	Predefined Categories (Accessed for This Research)
1	ERTA	Sport
2	Reporter online newspaper	Politics, Art and Culture, Sport
3	NMA	Meteorology
4	Mahibere Kidusan	Religion (Christian)
5	Deutsche Welle	Politics, sport, Culture and Youth
6	VOA	Politics
7	MoH	Health
8	MoE	Education
9	ECTM	Culture and Tourism
10	AACTB	Culture and Tourism

This thesis work concentrates on Amharic document categorization and for this purpose 7 categories are selected to run the experiment.

The categories are:

1. Sport,
2. Politics,
3. Religion (Christian),
4. Meteorology,
5. Tourism,
6. Education, and
7. Health

A total of 985 documents collection are mapped to these seven categories. These documents are further divided into two sets. They are

1. Training set with 637 documents and

2. Test set with 348 documents.

Training set collection and testing set collection are shown in Table 3.2.

Table 3.2 Training set collection and testing set collection

No	Category	Training set	Test set
1	Sport	84	45
2	Politics	203	109
3	Religion	71	38
4	Meteorology	68	45
5	Tourism	63	33
6	Education	78	41
7	Health	70	37

All texts from each document are represented in Unicode format. Length of each document varies such as five to more than 65 lines. All the training documents are assigned to only one category.

3.4 THE AMHARIC WRITING SYSTEM

Amharic [32] (Amharic: አማርኛ amarəññа) is a Semitic language spoken in Ethiopia. It is the second most-spoken Semitic language in the world, next to Arabic. It is the official work language of the Federal Democratic Republic of Ethiopia. Thus, it has official status and is used nationwide. Amharic is also the official or working language of several of the states within the federal system. It is written using Amharic Fidel, ፊደል, which grew out of the Ge'ez abugida.

Amharic Fidel is an abugida, in that each character represents a consonant and vowel combination. The alphabets are organized in groups of similar symbols on the basis of both the consonant and the vowel. In order to view the fidel, it is required a font that supports Ethiopic, such as Visual Geez Unicode.

The Amharic script has 33 basic characters [54]. There are six orders derived from the basic forms. The first five orders represent a combination of a consonant and vowel. The sixth order may represent either a consonant alone or a consonant followed by a vowel. Therefore, there are 231 (7 × 33 = 231) core characters in Amharic writing system. Besides these, there are over forty others which contain a special feature usually representing labialization.

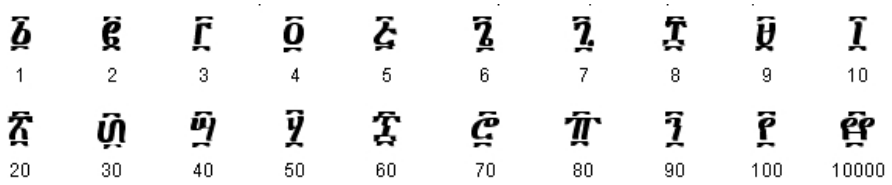
The list of these 33 basic Amharic characters is shown in Annex 1.

3.4.1 Amharic Punctuation Marks

Amharic punctuation marks [35] includes word separator (፡), full stop (።), comma (፣), semicolon (፤) colon (፥), preface colon (፦), question mark (፧), paragraph separator (፨)

3.4.2 Amharic Numbers

Amharic numbers are basically developed from Greek alphabet [43]. The Amharic number system consists of twenty (20) single characters. They represent numbers one to ten, multiples of ten (twenty to ninety), hundred, and thousand. In order to make them look like the Amharic characters the symbols are modified by adding a horizontal stroke above and below. The following are the Amharic numbers.



Recent time a new Amharic zero character (i.e) is introduced [55] but it is not yet included in Unicode System.

3.4.3 Major Problems of Amharic Texts

There is a considerable amount of systemic redundancy [36], which lacks several consonant sounds found in the phonology of Geez. Thus, 4 distinct sets of 7 can represent the sound /h/ + vowel: **ሀ ሐ ኀ ኸ.** 2 sets represent /s/: **ሰ ሠ** and 2 /s'/ (ejective) **ጸ ፀ**.

These different fidels can be used interchangeably without meaning change. For example, the word "secretary" can be written as, **ጸሀፊ**, **ጸሃፊ**, **ፀኃፊ**, **ፀሓፊ**, etc … all mean the same, without change in meaning.

In this research the above mentioned problem doesn't affect the performance of the model. This is because each Geez character is converted to Latin equivalent form so that those different characters would have the same form. E.g. Both **ሀ** and **ሐ** are translated to "ha".

The other problem is in the formation of compound words. Compound words are sometimes written as a single word and sometimes as two separate words. For example, the word "School" can be written as "**ትምህርት ቤት**" or "**ትምህርትቤት**".

In Amharic language, it is common to write some words in shorter form using "/" (forward slash) or "." (dot). The short form of words can be expanded as single or a combination of words. **ጠ/ሚ**, which is expanded as **ጠቅላይ ሚንስትር** (means Prime Minister), is an example for the former. **እ/ር** is a short form of the single word **እግዚአብሔር** (means God). **እ.ኤ.አ.** is a short form of **እንደ ኤሮፓውያን አቆጣጠር** (Gregorian Calendar).

Another problem of the language is there can be different ways of writing a single word. Among many possible reasons, one major reason for this can be regional dialects [56] that can impact word formation in the basic level where the words are more likely to be written following their spoken form; "**ማነው**" vs. "**ምነው**","**ሆኗል**" vs. "**ሆኖአል**","**ሀገር**" vs. "**አገር**" ,

etc. Another one is, in Amharic there are many ways of writing loan words, i.e words that are taken from foreign languages. For example, the word minster can be written as "ሚንስትር", "ሚኒስቱር", "ምንስቱር", "ምንስትር" etc.

3.4.4 AMHARIC STEMMER

Stemming is the process of reducing morphological variants of a word into a common form [37]. For languages like Amharic or Arabic, that have a much richer morphology, this process involves dealing with prefixes, infixes and derivatives in addition to the suffixes. It is applied during indexing and is used to reduce the vocabulary size, and it is used during query processing in order to ensure similar representation as that of the document collection. Even though it is believed that using stemmer is very important for reduction of total number of important words for the categorization of documents due to inability to get it from the shelf in time, simply prefixes and suffixes are removed from texts.

At the time of pre-processing the following prefixes were removed. **በ, ለ, ከ, የ, ስለ, እንድ.** Also suffixes **ን, ም, ና, ዎች** were removed to get stem words. In addition to that possible combinations of suffixes were also removed. For example a word "**እቃዎችና**" both **ዎች** and **ና** removed to get stemmed word **"እቃ".** List of all suffixes and prefixes removed to get stem words are attached as Annex 4.

3.4.5 AMHARIC FONT REPRESENTATION

The Unicode Standard [38] is a widely used character coding system designed to support the worldwide interchange, processing, and display of the written texts of the diverse languages and technical disciplines of the modern world. To resolve any ambiguity and inconvenience which can be made due to variation of fonts used by the different source of the training and test data, all Amharic documents (both the training and test documents) converted to Unicode format before stored in the database. In this research each Amharic character is translated to English equivalent form and the translated equivalent form of each term is used for further process.

3.5 RELATED WORK

Researchers [8, 9, 10, 34] studied different document categorization techniques to categorize Amharic documents and they have reported different results and there research findings are

summarized in this section. There experimental results show that different categorization techniques results different performance.

Yohannes [10] has done a research to categorize Amharic news documents from Ethiopian News Agency (ENA). He used Decision Tree and Support Vector Machine (SVM) classifiers. In his research work he stated that to represent documents he identified relevant feature words. As he said the identification of relevant words determines the efficiency and accuracy of the classification. Standard pre-processing tools and methods are therefore very important for automatic classification.

Because of the lack of standard in the Amharic writing system and unavailability of Amharic text processing tools, the focus of his research was on developing a document-pre-processing scheme which facilitates for an efficient automatic classification of Amharic documents.

To this end much attention was given to the processing of the source data by developing and enhancing the following tools. The tools are specific to the source data – Amharic news documents from ENA.

- A tool to correct word spelling variations. Focusing on spelling variation due to pronunciation differences.
- Enhancement to the suffix and prefix removal tool developed in a previous study, so that it can perform semantic analysis before stripping-off affixes from words.
- A tool to correct word variations due to gender marker suffixes.
- A tool to correct word variations due to number marker suffixes.
- A tool to merge compound words (when they may result in semantic loss if separated) written as separate words.

The use of these tools (which enabled 10 to 30 % feature reduction) in addition to other tools and data reduction methods helped to analyze the huge source data (69,684 news items after data cleaning) and measure classifier performances.

Because of the high dimensionality of the source data, classifier algorithms that are suitable for high-dimensional data, Decision Tree and Support Vector Machine (SVM) classifiers were selected for the research experiment. The open source Weka package is used for the automatic classification of the preprocessed data. Out of the many classifier algorithms available in

Weka, the Logic Model Tree (LMT) and the Library of SVM (LibSVM) classifiers were used for performance testing.

Both LMT and LibSVM classifier showed good classification accuracy correctly classifying 79.72% and 81.15% of the test instance into the 15 news categories, respectively. However, the computational cost of the automatic classification was very high - taking several hours in high capacity computers (Computers with 512 MB RAM and 3.7 GHz speed).

The classification performance measures indicate the need for additional works in developing tools and methods for mining Amharic data.

Samuel Eyassu and Björn Gambäck [8] addressed in their research classification of Amharic news items using artificial neural networks. Their experiment results showed that the best ANN model showed a precision of 60.0% when trying to cluster unseen data, and a 69.5% precision when trying to classify it.

The objective of Surafel Teklu's [9] research was to investigate the application of machine learning techniques to automatic categorization of Amharic news items. 11, 024 news articles were used to do his research. He has done text preparation and preprocessing was done. Stop-word and words that occur in 3 or less documents were removed from the collection. Thirty-three percent of the data was used for testing purposes. Machine learning techniques, Naïve Bayes and k Nearest Neigbor classifiers, were used to categorize the Amharic news items.

The result of his research indicated that such classifiers are applicable to automatically classify Amharic news items. However, the classifiers work well when the categories contain almost evenly distributed news items. The best result obtained by the naïve Bayes and kNN classifiers is on three categories data (95.80% vs. 89.61%) and the least performance is shown on the 16 categories (78.48% vs. 64.50%) respectively. The 16 categories contain unevenly distributed data than the three categories and it is learnt that unevenly distributed numbers of documents over the categories decreases the performance of both classifiers; K nearest Neighbor dramatically decreases than naïve Bayes. His research indicated that Naïve Bayes is more applicable to automatic categorization of Amharic news items.

Zelalem Sintayehu [34] conducted a research to categorize Amheric documents of Ethiopian News Agency (ENA). In his research he used statistical techniques of automatic classification in all the steps (i.e. 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).

In the preprocessing activities he has done stemming and stopword removal. In his 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.

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 his 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.

CHAPTER FOUR

DESIGN AND DEVELOPMENT OF THE AUTOMATIC CATEGORIZER

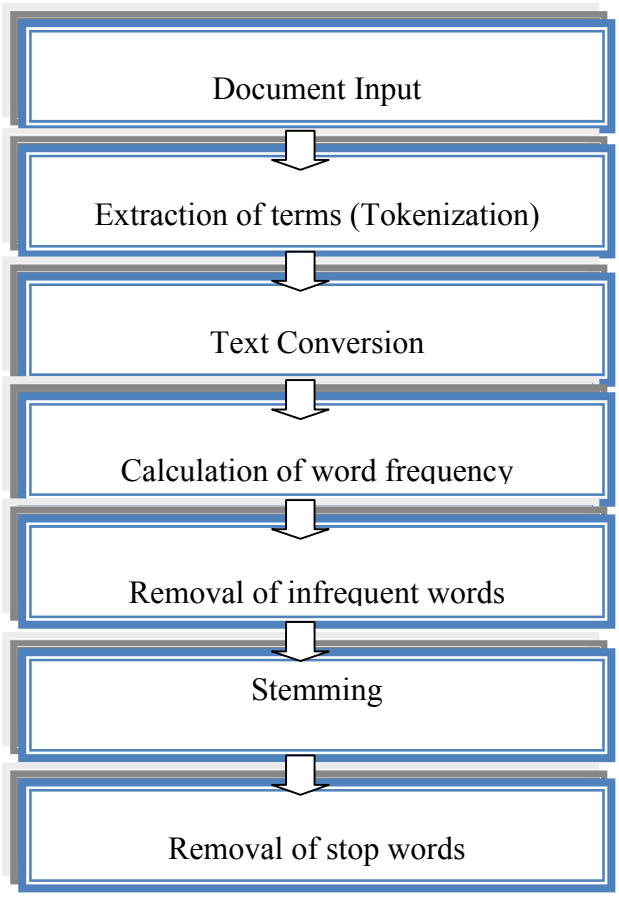
4.1 Introduction

In this chapter the design and development of the automatic Amharic documents categorizer which uses Apriori algorithm is discussed in detail. Apriori algorithm implementation and testing for Amharic document categorization are also presented.

4.2 Data Pre-processing

In order to convert both training as well as test data from the original data to suitable data as an input to Apriori algorithm, there are several operations required. Fig. 4.1 describes the block diagram of data processing phase which was applied in this research.

Fig 4.1 Block diagram of Amharic document preprocessing phase



First of all texts from the training documents were tokenized and stored in a database. Tokenization is the process of breaking parsed text into pieces, called tokens [57]. During this

phase punctuations and any non Amharic characters are removed and only Amharic words are selected from each document. For example consider the sentence **"በነገዉ ዕለት ዝናብ ሰጪ የሆኑ የአየር ሁኔታ ክስተቶች በአብዛኛዉ ዝናብ በማግኘት ላይ ባሉ የሀገሪቱ አካባቢዎች ላይ ከሞላ ጎደል ቀጣይነት እንደሚኖረዉ ይጠበቃል።"** from a document which belongs to category Meteorology is tokenized as shown in Table 4.1.

Table 4.1 List of tokens

በነገዉ	**ሁኔታ**	**ላይ**	**ከሞላ**
ዕለት	**ክስተቶች**	**ባሉ**	**ጎደል**
ዝናብ	**በአብዛኛዉ**	**የሀገሪቱ**	**ቀጣይነት**
ሰጪ	**ዝናብ**	**አካባቢዎች**	**እንደሚኖረዉ**
የሆኑ	**በማግኘት**	**ላይ**	**ይጠበቃል**
የአየር			

In Amharic language, it is common to write some words in shorter form using "/" (forward slash) or "." (dot). In this preprocessing phase the short forms of words are automatically identified and manual replacement to the appropriate form has been done. After text conversion, removing rare words has been done. To do this first of all frequency of each words determined and those words which have frequency less than 9 considered as rare words. Those words identified as rare words excluded from further processing.

Documents were contained several occurrences of words like **'ሰዎች', 'ሰዎቹ', 'ሰዎቻችን', 'ሰውየው', 'የሰውየው', 'ሰውየውና', 'ሰውየውን', 'በሰውየው', 'ሰዎቹ'**. Different words share the same word stem (i.e**'ሰው'**) and a module which was designed for this particular purpose used to convert those different representations to their stems.

Stemming is a technique for the reduction of words to their stems or root variant. Doing this reduce computing time and space as different forms of words are stemmed to a common word. In this thesis work a module which was developed to steam Amharic words designed to remove suffixes and prefixes from each word. Among those suffixes and prefixes some of them are listed below

Suffixes: ‘ዎች’, ‘ና’, ‘ቻችን’, ‘ንንም’, ‘ዎችና’, ‘ውናም’, ‘ውን’, ‘ዎችም’, ‘ዎችንም’, ‘ውንም’, ‘ውና’, ‘ናን’, ‘ናንና’, ‘ምውን’, ‘አቸው’

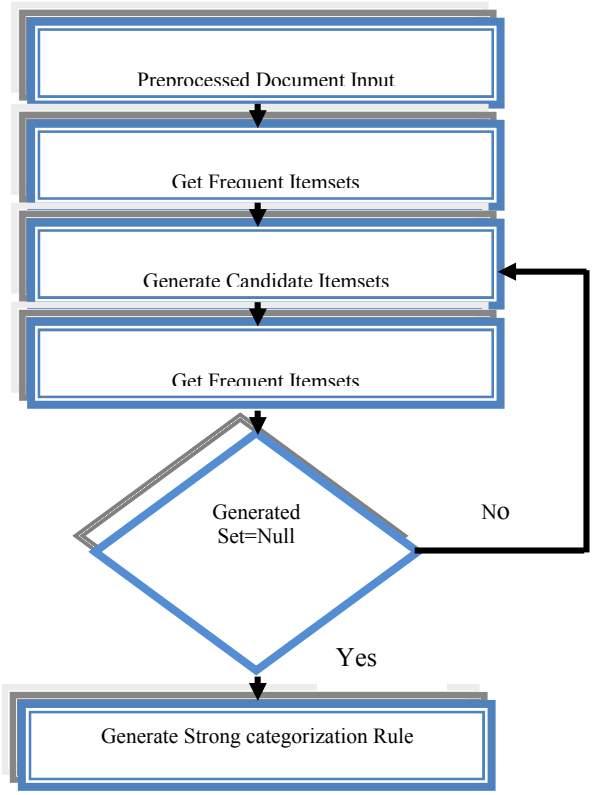
Prefixes: ‘የ’, ‘እንደ’, ‘እንዲ’, ‘ወደ’, ‘ሲያ’, ‘እየ’, ‘ከ’, ‘ለ’, ‘በ’. After the stemming sub-process, identification of rare words has been done. To do this, frequency of each word is determined. Words those having a frequency of less than 9 were considered as rare words and removed from further processing.

Common words such as **‘እንዲሁም ‘, ‘እና’, ‘በዚሁ’, ‘ነው’, ‘ስለዚህ’** etc that occur in almost all documents are removed and this is because those words do not help in deciding whether documents belong to a category or not. Such words are referred as stop words. To store stop words the system which is developed has a sub-system (module) which can load all the available Amharic documents so that manually the stop words are selected and stored in the database. Once those words are identified as stop words, they were not further processed. Since the influence of those irrelevant words for the categorization purpose is illuminated, it helps to save both time to process the texts and storage space.

4.3 Processes to Generate Frequent Itemsets

The document categorization method preferred in this research needs two basic steps; find all frequent itemsets and generate strong association rules from frequent itemsets. To do those two important activities the following diagrammatically shown high level design is applied.

Fig 4.2 High level design for Amharic document categorization using Itemsets method.



The following text describes the low level extended Apriori algorithm design to associate terms with document categories;

1. Scan the transaction database to get the frequency of each term.
2. If frequency of an item is greater than or equal to the minimum frequency (e.g. 9), calculate the frequency of the term in each category. Else remove the term from the database.
3. If a term has document frequency greater than or equal to 47 at least in 2 categories then remove the term from the database. 47 is taken in this research because at this point the categorization algorithm shows the most optimal result.
4. If document frequency of a term is greater than the minimum support threshold (e.g. 38) then add to frequent 1-itemset (L_1).
5. Use L_{k-1} join L_{k-1} to generate a set of candidate k-itemset.
6. Scan the transaction database to get the support S of each candidate K-itemset
7. If support is greater than or equal to the minimum support add to K-frequent itemsets
8. Repeat step 6 until generated set is Null.
9. Calculate weight for all frequent itemsets.
10. Link a term with a category which has maximum weight.

After association of terms with categories the next step is prediction of document category. To do this the low level design is described as follows.

1. Scan the transaction database to get each term.
2. Calculate the weight factors for all itemsets.
3. Calculate the sum of weight factors of all itemsets for each category.
4. Select the category which has highest weight factor and associate (link) the category with the document.

After the preprocessing phase building an inverted index was done. Inverted Index [43] is an indexing approach which can help to map a data with a given content. In this research inverted index is applied to produce data structure of each term in a way which indicates where it is located.

There are two types of inverted indexes [43], these are referred as record level inverted index and word level inverted index In this work, record level inverted index is applied. A record

level inverted index contains a list of references to documents for each term. Whereas a word level inverted index additionally contains the positions of each word within a document. The following simple example illustrates the concept of Inverted Indexing. Assume that there are 4 documents named D1, D2, D3 and D4. The content of each of the document is presented as follows:

D1: "ሰውየው መጽሀፍ እያነበበ ነው"

D2: "ሰውየው መጽሀፍ ማንበብ ይወዳል"

D3: "ተማሪው መጽሀፍ አለው"

D4: "ተማሪው እና ሰውየው መጽሀፍ ማንበብ ፍላጎታቸው ነው"

To build inverted index, it is required to select each term from all the documents and keeping record of in which documents the terms are found. Table 4.2 shows how a simple inverted index can be generated for the above four documents. For the term "**ሰውየው**" document identification code 1, 2 and 4 used. This is because documents D1, D2 and D4 contain the term "**ሰውየው**".

Table 4.2 A record level term index

Items	**Document Identification Code**
ሰውየው	{1, 2,4}
መጽሀፍ	{1, 2, 3, 4}
እያነበበ	{1}
ነው	{1, 4}
ማንበቡ	{2, 4}
ይወዳል	{2}
ተማሪው	{3, 4}
አለው	{3}
እና	{4}
ፍላጎታቸው	{4}

The inverted index also can help to find document frequency for each word in the database. Document frequency [58] is defined as the number of documents that contain a particular term.

In this thesis the document frequency for each term determined so that it is used to identify significant terms from the collection of documents.

For the above terms, document frequencies are determined and shown in Table 4.3. Document frequency for the term "**ሰውየው**" is 3 because the term occurs in three documents D1, D2 and D4.

Table 4.3 Terms with their document frequencies

Items	**Document Frequency**
ሰውየው	3
መጽሀፍ	4
እያነበበ	1
ነው	2
ማንበቡ	2
ይወዳል	1
ተማሪው	2
አለው	1
እና	1
ፍላጎታቸው	1

In the above example the issue of stemming the Amharic terms is not considered. However in the actual experiment while examining the Apriori algorithm to categorize Amharic documents stemming was done for each term.

The following table (Table 4.4) shows few terms from the training dataset along with their document frequencies

For instance, in Table 4.4 the document frequency of a term **"አገላለጽ"** is shown as 61 because it occurs in 61 documents out of 637. Here 61 is the sum of all terms (such as **"አገላለጹን", "አገላለጽና"** etc.) which results the same root or stem according to the stemmer used. The total number of terms obtained from the training dataset after building inverted index is 240,340 excluding all duplicates.

One of the major challenges to categorize text is the large size of terms in the training dataset. This is referred as high dimensionality [50] of feature space. To save space and time and also

to higher the performance of the algorithm reducing the dimensionality is important. There are many known approaches to do dimensionality reduction. In this research works since Apriori algorithm is mainly focus on frequency of terms in the dataset, terms are filtered based on document frequency threshold.

Table 4.4 20 sample terms from the training dataset along with their document frequencies

Term (In Amharic)	**Term (Translated to Latin alphabet)**	**Document frequency**
አጂ	AJI	2
አገቡ	AGEBU	19
አጋቢ	AGABI	6
አጅበው	AJBEW	5
አጋፋሪነት	AGAFARINET	11
አጋዥ	AGAZH	19
አጓጊ	AGWAGI	13
አገዛዞች	AGEZAZOCH	67
አጋጠሙ	AGATEMU	161
አጉል	AGUL	11
አጋለጣቸው	AGALETACHEW	19
አጉልተው	AGULTEW	33
አጋልጠውታል	AGALTEWTAL	54
አገላለጽ	AGELALETS	61
አገለለ	AGELELE	28
አጋማሽ	AGAMASH	23
አጀንዳ	AJENDA	229
አጋንንት	AGANNT	93
አገሮች	AGEROCH	102
አጋሩ	AGARU	7

In this approach term selection or filtering has been made based on a predefined threshold value. Based on this concept words that occurred nine or fewer times within the training dataset are considered as useless features and therefore they are removed from the items list. After removing rare words, if terms have document frequency greater than 47 at least in two different categories, it is considered that the term cannot distinguish between two documents and hence can be removed. After removing these terms which does not satisfy the predefined threshold values, only 2698 unique terms out of 240,340 terms left. After doing several experiments, the minimum document frequency (i.e threshold) is set to 38. At the time of finding the optimal performance of the algorithm, different parameters (such as term frequency) were altered and the performance is measured.

After selection of the significant terms which are assumed to be important for the categorization of documents, the next step is to generate frequent itemsets. Frequent itemsets are generated using Apriori algorithm for the purpose of categorizing documents into categories.

In the previous chapter how Apriori algorithm works with illustration is clearly presented. In this thesis, instead of applying Apriori algorithm to find knowledge from basket of items, it is used to find knowledge from basket of significant terms obtained from training documents. Significant terms collected from the training documents are considered as items and basket of terms are considered as an itemset.

A table is created in the database to store document verses significant terms matrix which helps to determine the frequent itemsets. Each row used to store record of document ID and the columns to represent significant terms. If a term occurs in a document then it is represented as '1' otherwise '0' as explained in the previous chapter. The minimum support threshold (θ) finally set as $\approx$6% i.e. if an itemset occurs in at least 38 documents then only it is considered as a frequent itemset. However in this research different θ values are examined and the findings are presented under *Precision and Recall section* (Section 4.4). After this, all the steps discussed in the earlier chapter under the topic DOCUMENT CATEGORIZATION USING ITEMSET METHOD is applied.

The following is a sample output of Apriori program with frequent itemsets represented as shown below:

Frequent 1-itemsets:

ሀንጻ, ቤተ, ዝናብ, ውጤት, ስፖርት, ኮሚሽን, ጠቅላይ, ሚንስቴር, ሰላም, ህዝብ,.....

Frequent 2-itemsets:

{ስፖርት, ኮሚሽን}, {ጠቅላይ, ሚንስቴር}, {ሰላም, ህዝብ},.....

Frequent 3-itemsets:

{ስፖርት, ኮሚሽን, አትሌት},.....

For example top ten frequent 1-itemsets from Politics, Meteorology, and Religion categories are tabulated as follows.

Table 4.5 Top twenty four frequent 1-itemsets from Politics, Meteorology, and Religion categories

Politics	Meteorology	Religion
መንግስት	ዝናብ	እግዚአብሔር
አገር	እርጥበት	ቤተ
ፓርቲ	ደረቅ	ክርስቲያን
ፖለቲካ	በልግ	ቅዱሳን
ኤርትራ	ሙከት	ክርስቶስ
ምርጫ	ሰብል	ሲኖዶስ
ኢህአዴግ	ሽለቆ	እርሱ
መብት	ስምጥ	ማህበረ

4.3.1 Itemsets Categorization Method

There are two phases in automatic document categorization, the *learning phase* and the *categorization phase*. In the learning phase a set of documents along with their categories are defined by domain experts. After that, a phase of categorization continues which is creation of a categorizer by learning from the internal representations of the training documents.

4.3.2 The training Phase

In this phase, a set of documents along with their categories are defined by specialists. Then, a categorization model is built using frequent itemsets method.

In this research paper the notations are taken from [13] and frequent itemsets are represented using 'Π' based on their cardinalities such as

1-itemsets are represented as $\Pi_1, \Pi_2, \ldots\ldots \Pi_{N1}$.

2-itemsets are represented as $\Pi_{N1+1}, \Pi_{N1+2}, \ldots\ldots \Pi_{N1+N2}$.

3-itemsets are represented as $\Pi_{N1+N2+1}, \Pi_{N1+N2+2} \ldots\ldots \Pi_{N1+N2+N3}$.

etc.

For each frequent itemset Π, find all documents that contain this particular itemset. Let us represent these set of documents as D_Π. For example itemset Π_1 corresponds to $D_{\Pi 1}$, Π_2 corresponds to $D_{\Pi 2}$ etc.

For each category C_i, there are a certain number of documents that fall in to this category. Such as documents that fall into category Sport is represented as DC1, category Politics is represented as DC2, category Meteorology is represented as DC3, category Tourism is represented as DC4, category Religion is represented as DC5, category Education is represented as DC6 and category Health is represented as DC7. Documents labeled in each category are shown below.

Sport = DC1= {D1, D2, D3, ……. D84}

Politics = DC2 = {D85, D86, D87, …… D287}

Meteorology = DC3 = {D288, D289, D290,…… , D355}

Tourism =DC4 = {D356, D357, D358,…… D415}

Religion = DC5 = {D416, D417, D418, …….. D489}

Education = DC6 = {D490, D491, D492, ……., D567}

Health =DC7 = {D568, D569, D570,…… D637}

To find out which itemsets fall into which categories, itemset Π_j is mapped with category C_i based on the maximum value of $W_{\Pi j}$.

The weight $W_{\Pi j}$ is calculated [11] using the following formula:

$W_{\Pi j} = (D\Pi_j \cap DC_i) / DC_i$ where i = 1, 2, 3, 4, 5, 6, 7 categories.

The denominator DC_i is used for normalizing with the number of documents associated with category C_i. It takes into account whether an itemset occurs in other categories as well. Significance of terms occurring frequently in documents other than DC_i is thus suppressed [11].

The following example demonstrates how this can be determined for frequent 1-itemset:

Let us take a Frequent 1-itemset indexed in the fifth order (i.e Π_5) = {**አባል**}

It is found in 78 documents.

$D\Pi_5 \cap DC_1$ =11, $D\Pi_5 \cap DC_2$ =57, $D\Pi_5 \cap DC_3$ =0, $D\Pi_5 \cap DC_4$ =1,

$D\Pi_5 \cap DC_5$ =6, $D\Pi_5 \cap DC_6$ =2, and $D\Pi_5 \cap DC_7$ =1.

$D\Pi_5 \cap DC_1$ is to mean that the term "**አባል**" found in 11 documents which are found under Sport category.

To determine to which category this itemset can be mapped is by finding common documents between Π_5 and DC_1 , Π_5 and DC_2 , Π_5 and DC_3, Π_5 and DC_4, Π_5 and DC_5, Π_5 and DC_6, and Π_5 and DC_7 . Π_5 is mapped only with that category which has maximum $W_{\Pi j}$ value.

$$W_{\Pi 5} = (D\Pi_5 \cap DC_1)/DC_1 = 11/84 = 0.13$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_2)/DC_2 = 57/203 = 0.28$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_3)/DC_3 = 0/68 = 0.00$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_4)/DC_4 = 1/63 = 0.02$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_5)/DC_5 = 6/71 = 0.08$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_6)/DC_6 = 2/78 = 0.03$$
$$W_{\Pi 5} = (D\Pi_5 \cap DC_7)/DC_7 = 1/70 = 0.01$$

Hence, itemset Π_5 is associated with category Politics because it has the highest weight (i.e 0.28) when compared to associating this itemset with other categories. In the same way weights for all the frequent itemsets constructed. All categories are mapped with their representative itemsets based on $W_{\Pi j}$ values. Those filtered itemsets were used as a training dataset so that during the testing phase the proposed model used these representative itemsets to categorize the new test (unied) documents to determine the correct categories for each test documents. This approach is known as a supervised learning approach[59] because a model is trained based on predefined documents and their corresponding categories.

4.3.3 Test Phase

Based on previous training a new document is supposed to be categorized and the model should help to predict correctly. As the result correct category label should be assigned for each new uncategorized document.

Since there are frequent 1-itemsets, 2-itemsets, 3-itemsets etc. a weight factor must be determined for the prediction purpose. This will give the possibility to measure in which category a given document lay. wf is defined to distinguish between singles, pairs, triplets, quadruplets, etc of an itemset i.e. 1-itemsets are defined by wf1, pairs by wf2, triplets by wf3, quadruplets by wf4 etc. When the candidate increases also the weight factor increases[11]. For prediction purpose a model developed associates a new document to the correct category based on the following formula:

$$W_{\Pi j} = \sum_{i=1}^{|c_j|} Wf_{|\Pi i|} \quad \text{(Eq 4.1)}$$

Where ($P_i \in C_j$) Λ (P_i Í D), for all j = 1, 2, 3, 4, 5, 6, 7 categories. D is the set of all significant terms obtained from the new test document. Wf_{Pi} is the weight factor of frequent itemsets.

According to [11] categorization weight can be determined by the sum of weight factors for all itemsets of a given category. Test document is associated with one category which has maximum weight factor. However if two or more categories have got the same maximum weight factor the document will be associated with those having the maximum weight factor. Total number of test documents used in each category in this research work is tabulated in Table 4.6.

The cleaning process for test documents is done in the same way as the cleaning process for training documents which has been done through the process of parsing, tokenization, stop words removal and stemming. Significant terms are generated for each test document. For demonstration, selected significant terms of one of the test document (TD5_1) from a Religion category is shown in Table 4.7.

Table 4.6 Testing set collection

No	Category	Number of Test set
1	Sport	45
2	Politics	109
3	Religion	38
4	Meteorology	45
5	Tourism	33
6	Education	41
7	Health	37

TD5_1={**ቤተ ክርስቲያን ሲባል የተለያዩ ትርጉሞች አሉት። በዚህም መሠረት ቤተ ክርስቲያን ሲባል ክርስቲያን ምእመናን፣ የቤተ ክርስቲያን መሪዎች፣ የቤተ ክርስቲያን አስተዳደርና የቤተ ክርስቲያን ሕንጻ ሲሆን ይችላል።** }

Table 4.7 List of significant terms of one of the test document with their corresponding frequency

Significant terms	Frequency
ቤተ	**71**
ክርስቲያን	**71**
ትርጉሞች	**17**
ምእመናን	**27**
መሪዎች	**9**
አስተዳደርና	**13**
ሕንጻ	**9**

Maximum $W_{\Pi j}$ value is calculated for each itemset and finally which category each itemset is mapped is determined and shown below (Table 4.8).

Table 4.8 Frequent 1 itemsets with their corresponding category.

Frequent 1-termsets	C1	C2	C3	C4	C5	C6	C7
{ቤተ}					X		
{ክርስቲያን}					X		
{ትርጉሞች}		X					
{ምእመናን}					X		
{መሪዎች}		X					
{አስተዳደርና}		X					
{ሕንጻ}				X			

For each selected significant term generated, the occurrence of the term in each category is determined. If a selected significant term generated occurs in a category then it increments wf value of that particular category. If it is a 1-itemset then wf equals 1, if 2-itemset wf equals 2 etc. In this way weights' of all selected significant terms for each category are determined and whichever is having highest value the document is linked with that category. The weight factors for TD5_1 are determined for each category and shown in Table 4.9.

Table 4.9 Weight factors of TD5_1.

No	**Category**	**wf of** TD5_1
1	Sport	0
2	Politics	2
3	Religion	7
4	Meteorology	0
5	Tourism	1
6	Education	0
7	Health	0

The given test document TD5_1 is mapped to category Religion because the wf of Religion is greater than the rest. If sum of weight factors are equal for any two categories, the document d belongs to both the categories.

For Religin category wf is 7. This is because

wf1 result of

Three frequent 1-itemsets **{ቤተ}, {ክርስቲያን}, and {ምእመናን},**

wf2 result of

Three frequent 2-itemsets **{ቤተ, ክርስቲያን}, {ክርስቲያን, ምእመናን}, and {ቤተ, ምእመናን},** and

wf3 result of

One frequent 3-itemsets **{ቤተ, ክርስቲያን, ምእመናን}**

wf = wf1 + wf2 + wf3

wf = 3 + 3 + 1

wf = 7

By applying the same method the testing documents' categories are predicted.

CHAPTER FIVE

EXPERIMENT

5.1 Introduction

Chapter five presents the experiment results found in the previous chapter (chapter 4) and the results are evaluated. The chapter illustrates the reasons behind documents which are wrongly predicted by the Apriori algorithm driven model.

5.2 Datasets

A total of 985 documents collection are mapped to these seven categories. These documents are further divided into two sets. They are

1. Training set with 637 documents and

2. Test set with 348 documents.

Training set collection and testing set collection are shown in Table 5.1.

Table 5.1 Training and testing documents collection with their source

No	Category	Total No of Documents	No of Training Documents	No of Test Documents	Document Source*									
					1	2	3	4	5	6	7	8	9	10
1	Sport	129	84	45	30	68			31					
2	Politics	312	203	109	81	149			44	38				
3	Tourism	96	63	33		26			17				32	21
4	Meteorology	113	68	45			113							
5	Religion	109	71	38				109						
6	Health	107	70	37							107			
7	Education	119	78	41								119		

Document sources* labeled from 1 to 10 are interpreted as follows

1 = ERTA
2 = Reporter Online
3 = NMA
4 = Mahibere Kidusan
5 = Deutsche Welle
6 =VoA
7 = MoH
8 = MoE
9 = ECTM
10 = AACTB

All texts from each document are represented in Unicode format. All the training documents are assigned to only one category.

5.3 Results

5.3.1 Precision and Recall

To measure the performance of the categorization model standard precision, recall and F1 values [60] were evaluated. Let True Positive (TP) be number of test documents which both experts and the model developed agreed as belonging to the same category.

Let False Positive (FP) is the number of test documents that are wrongly categorized by the model as belonging to that category.

Precision is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

Let False Negative (FN) be the number of test documents which are not correctly categorized as belonging to the correct category but should have been.

Recall is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

The F1 measure (also called traditional F-measure or balanced F-score or **F_1 score**) is defined as the harmonic mean of precision and recall: [39,60]: The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and worst score at 0.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

In this research different support thresholds (θs) used to calculate precision, recall and F1 values. This is done for the purpose of evaluating the effect of variation of θ over performance of the model. The following table (Table 5.2) shows the result found when θ is 6% and this value is where the maximum performance is achieved in both precision and recall.

Table 5.2 Precision, recall and F1 values when θ = 6%

Category	Total Documents	TP	FP	FN	Precision (%)	Recall (%)	F1 (%)
Sport	45	43	3	2	94	96	95
Politics	109	102	6	7	94	95	94
Religion	38	38	0	0	100	100	100
Meteorology	45	45	0	0	100	100	100
Tourism	33	33	1	0	97	100	98
Education	41	37	2	4	95	90	92
Health	37	35	3	2	92	96	94

The average precision, recall and F1 values obtained are 96%, 97% and 96% respectively.

As shown in Table 5.2, when calculating precision, recall and F1 values from a total of 348 test documents a total of 15 false negative value is observed. This means the model developed has predicted wrong category labels for fifteen documents out of 348 documents. For instance, in the case of Politics category, 7 false negative value is observed. Those 7 documents which were wrongly categorized mapped to 4 different categories. The documents were wrongly categorized 2 in Sport, 1 in Tourism, 2 in education, and 2 in Health categories. This is because the maximum weight factors of those test documents were greater than that of the weight factors of the Politics category. Among those seven documents D168 was wrongly categorized as Education document this is because;

- The weight factor of D168 with Sport is 7
- The weight factor of D168 with Politics is 19
- The weight factor of D168 with Religion is 2
- The weight factor of D168 with Meteorology is 4
- The weight factor of D168 with Tourism is 5

- The weight factor of D168 with Education is 26
- The weight factor of D168 with Health is 9

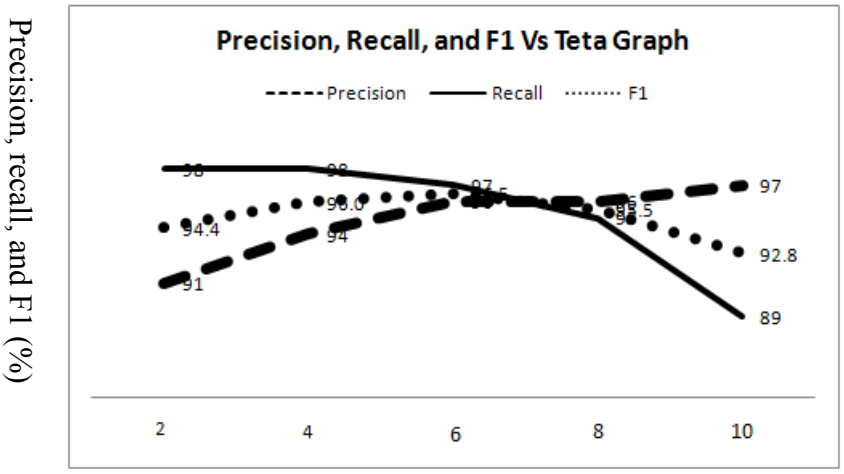
Among those weight factors it is clear that weight factor of D168 with Education has the maximum value and of course it is greater than that of Politics.

Multiple experiments were conducted to find the optimal performance of the model. In those experiments factors which can affect the performance are identified and numerical evidences are shown below.

5.3.2 Variation of q

Precision and Recall are determined for 5 different values of θ. Those θ values are taken near to 6%. This is because it is found that both Precision and Recall values arrived to their optimal value when the document frequency threshold is set to 38, i.e ≈6%. The following graph (Graph 5.1) shows Precision, Recall, and F1 Vs θ

Graph 5.1 Precision, Recall, and F1 Vs θ where the measure of the three parameters is in percentile



Where the above relation is observed in the experiment the all other factors were maintained as constant. The graph shows that the Precision increases while θ increases however Recall decreases while θ increases.

Experiments have been done for large values of θ (≥20) and results show that F1 values dropped drastically.

5.3.3 The effect of stemming the training documents

A simple experiment has been done to check the effect of stemming training documents before applying document categorization. Three values of θ were set and F1 for both stemmed and non-stemmed documents were determined. Table 5.3 shows the summary.

Table 5.3 F1 values for both stemmed and non-stemmed documents

q(%)	F1(%)	
	Stemmed Documents	**Non-stemmed Documents**
5	96	81
10	93	79
15	87	74

The above table shows that variation of θ affects F1 for both stemmed and non-stemmed documents. However in this experiment since the optimal value of F1 is not determined for non-stemmed documents it would not be possible to compare average F1 values.

Experiments showed that processing stemmed documents saves computational time. To train stemmed documents less time is required than that of training non-stemmed documents. Using the same computer (Dell OptiPlex 780, Core 2 Duo, 2.93 GHz, 4GB RAM) training stemmed documents took an average of 28 minutes but to train non-stemmed documents an average of 41 minutes required.

5.4 Comparison of the proposed system with related works

The performance achieved in this research was compared with other 4 research findings and the results are summarized in Table 5.4. The precision value achieved by this research is the highest one compared with other four experimental results.

Table 5.4 Comparison of precisions

No	Research Title and Method Used	No Test Documents	Precision (%)	No of Categories
1	Automatic Categorization of Amharic News Text Using Naïve Bayes Method [9]	2721	89.9	7
2	Automatic Amharic news Categorization[34]	321	90.5	3
3	Amharic documents categorization using Logic Model Tree (LMT) and Library of SVM (LibSVM) [10]	2,485	95.2	5
4	Artificial Neural Network to categorize Amharic documents [8]	-	69.5	-
5	Amharic Document Categorization Using Itemsets Method	348	96.0	7

5.5 Discussion

The performance gained using the proposed method shows the best performance when it is compared with other four findings. However the existence of variations of experimental setups in all those five different techniques evaluation of performance by comparing only the precision may not read to scientific conclusion. Therefore to compare the performance of each of the methods all the experiments should be conducted in the same environment. (e.g. by using the same corpus both for training and testing).

5.6 Implementation of the model to solve real world problem

After conducting those experiments to show how the findings of this research can be applied to solve a real problem, a problem is identified and solved as follows.

Problem: Creating a website which can automatically grab available Ethiopian videos from YouTube.

Method used: Itemsets Method

Procedure: Basic steps followed to retrieve a list of videos matching a user-specified search term.

1. Videos from youTube were searched using keywords "Ethiopia" and **"አማርኛ".**
   To do this google provides an API which can help to retrieve a list of videos matching a user-specified search term [61].
   For example the following API query can grab 50 videos' information which matches **"አማርኛ"** from youTube. https://gdata.youtube.com/feeds/api/videos?q=**አማርኛ**&start-index=1&max-results=50. YouTube allows grabbing a maximum of 50 videos at a time. In addition to that using an API query, it is not allowed to access videos those are ranked above 1000.
2. Among those videos grabbed by applying procedure 1, videos which are assumed to have Ethiopian contents are manually identified. Here Simple observations of Titles and Descriptions of videos have been done. None of the videos were watched. 400 videos were selected as training dataset. At the time of selecting those 400 videos, 200 of them were deliberately selected among those which have Amharic text in either of the Title or Description. Information about the 400 videos was stored in a database. (MySQL database is used and for coding PHP is used).
3. Using the information stored in the database generation of frequent itemsets has been done. Those frequent itemsets are now considered as keywords which can describe Ethiopian contents. Using those new keywords step 1 was re-executed. Here large number of keywords were generated.
4. Without manually filtering videos found in step 3, step 2 was applied.

Following those 4 procedures, above 50,000 Ethiopian content videos were automatically collected from youTube in a matter of 3 hours. And this application is now available in the Internet. It is hosted at http://hahutube.com.

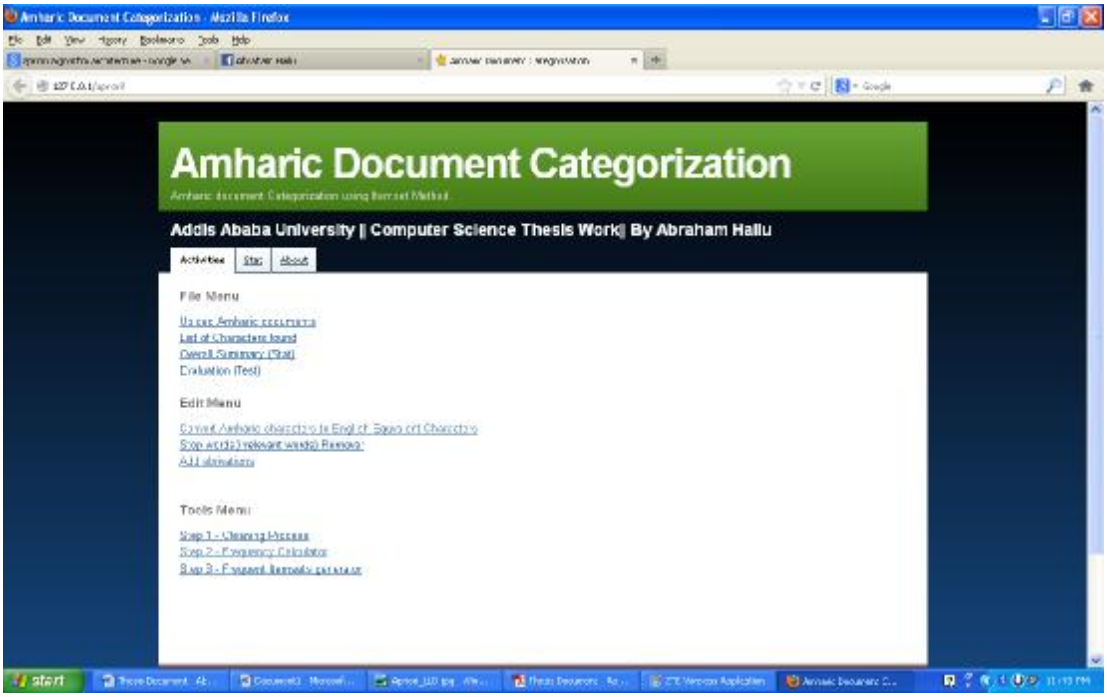
From this application, in addition to demonstrating how Apriori algorithm used to categorize documents, it is observed that the algorithm is convenient to work in mutilingual situation.

Fig 5.1 shows the screenshot of http://hahutube.com

Fig 5.1 hahutube.com homepage

Fig. 5.2 hahutube.com video collection page

Fig 5.3 demo for Amharic document categorization



CHAPTER SIX
CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

Existence of an efficient automatic Amharic document categorization system has much importance. This is because its existence can remove tedious document categorization tasks. Moreover the costs associated with handling and processing these documents will dramatically reduce. Using such systems to categorize documents can save time (e.g. by batch processing) and are more reliable than manual categorization of documents.

This thesis contributes to this goal by examining a useful method for automatically categorize Amharic documents. In the work described in this thesis, the possibility of carrying out Amharic document categorization using association rule has been investigated. The study provides evidence that association rule can be used in automatic Amharic document categorization efficiently and effectively.

6.2 Conclusion

Automatic categorization is concerned with the procedures and systems that can make comparison between terms found in documents so that to automatically categorize documents to their predicted categories. The automated prediction of documents categories should provide good results. The result of this research showed that the system developed to categorize Amharic document has good performance. Moreover the performance of the system is best when it is compared with other similar research finds' results [8, 9, 10, 34].

The research has tried to look into the techniques of automatic categorization using itemsets method and identified factors affecting automatic categorization.

Handling automatic Amharic document categorization begins with understanding the basics of the language. However experimental results obtained in this study demonstrate that, the new system introduced does not make use of advanced (detailed) language-specific characteristics or deep linguistic analysis. To produce Amharic text different kinds of Amharic software can be used. The variation of those software products (such as Power Ge'ez, Visual Unicode, Abinet, etc) used may results variation on the technique to process texts for categorization. To

solve this problem conversion of texts produced by those different software products into a common format is relevant. In this study, all the texts were converted to a Unicode system.

All texts appeared in a given document may not be relevant so filtering those irrelevant non-Amharic characters from the documents is important. In this research distinct 110 non-Amharic characters were removed from training as well as test documents. This was important for the performance of the entire system.

Typographic error sometimes may result non frequent terms and those terms could not have any contribution for categorization. In the preprocessing phase of the document categorization since there was no manual correction of such terms, by setting a minimum frequency threshold, they were removed. Variation of the threshold affects the performance of the system. To find the maximum performance repeated experiments were done by varying the threshold value starting from 0 to 20. While doing the experiments all other parameters were kept constant. Finally the maximum performance was observed when the minimum threshold was set to 9. This means that all items whose frequency was less than 9 was discarded and considered as rare terms. At the time of training the model, to get a good performance removing those non-frequent terms had positive impact.

Experimental results show that words having document frequency greater than 47 at-least in two different categories could not help to distinguish between two documents and hence removed. This shows that identification of important words (feature selection) play an important role for the performance of the system. To find the optimal value it needs repeated experiments by varying document frequency. Without performing exhaustive identification and removal of stop words, this technique easily selected those stop words. Since stop words occurrence vary from one category to the other, the above mentions approach can be taken as a better solution to have smaller feature size.

The process for reducing inflected words to their stem or root form improved the performance of the categorization system by reducing the feature size.

Frequency based term weight determination has shows interesting result to determine in which category a given word lays. To predict the category of a given document, determination of the

maximum weight factor results a good performance. Experimental result shows that the weight factor which was result of the count of frequent words also considered the frequent occurrence of words together so that this improved the performance of the system.

Experimental results obtained in this study demonstrate that, Apriori algorithm does not make use of advanced (detailed) language-specific characteristics or deep linguistic analysis.

Results demonstrate that relatively high categorization accuracy can be achieved by using Apriori algorithm. The precision and recall values obtained from the experiment shows that this algorithm is an efficient to perform automatic Amharic document categorization. Moreover it is found that a threshold value which is used to determine the frequent itemset has a very important role.

6.3 Recommendation

In this research the possibility of developing an automatic and efficient Amharic categorization system is proven. However since Amharic document categorization is an active research area more researches are required for the performance improvement and accuracy of the documents categorization. Different scholars [3, 5, 6, 11] present the existence of different enhanced implementation techniques of Apriori algorithm but in this research those variations were not considered and this needs further study.

The performance evaluation of the proposed algorithm has been done for different data sets and in comparison with existing technique like [8, 9, 10, 34] it is found that the proposed system has efficient and superior performance for categorizing Amharic documents. However the existence of variations of experimental setups in those examined Amharic document categorization techniques may not lead to scientific conclusion by comparing only the precision. Therefore to compare the performance of each of the methods all the experiments should be conducted in the same environment.

REFERENCES

[1] Jiawei Han, Micheline Kamber. “Data Mining Concepts and Techniques” A book 2001 Edition (pp. 279-330)

[2] Agrawal R, Imielinski T, Swami AN. "Mining Association Rules between Sets of Items in Large Databases." SIGMOD. June 1993, 22(2):207-16, pdf.

[3] Mannila H, Toivonen H, Verkamo AI. "Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases (SIGKDD). July 1994, Seattle, 181-92, ps.

[4] Anita Wasilewska, Apriori Algorithm, www.cs.sunysb.edu/~cse634/lecture_notes/07a Apriori.pdf, (Accessed on December 5, 2011).

[5] Lauri Lahti, Apriori algorithm, Seminar of Popular Algorithms in Data Mining and Machine Learning, TKK, Presentation on 12/3/2008

[6] Michael R. Wick and Paul J. Wagner , Department of Computer Science, University of Wisconsin-Eau Claire, Eau Claire, WI 54701. Using Market Basket Analysis to Integrate and Motivate Topics in Discrete Structures.

[7] Sowjanya Alaparthi, Market Basket Analysis, www.itk.ilstu.edu/.../Market%20Basket %20Analysis%20new.ppt, (Accessed on December 8, 2011).

[8] Samuel Eyassu and Björn Gambäck. Classifying Amharic News Text Using Self-organizing Maps. In Proc. of ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, Jun. 2005.

[9] Surafel Teklu Weldesellassie, Automatic categorization of Amharic news text: A machine Learning approach, A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Information Science, July 2003

[10] Yohannes Afework Worku, Automatic Amharic Text Categorization, Addis Ababa University

[11] Jiri Hynek, Karel Jezek, Ondrej Rohlik, Short Document Categorization - Itemsets Method

[12] Amharic Language, http://en.wikipedia.org/wiki/Amharic_language, (Accessed on December 16, 2011).

[13] Jiri Hynek and Karel jezek, Use of Text Mining Methods in a Digital Library, elpub2002

[14] Frédéric Flouvat · Fabien De Marchi · Jean-Marc Petit, A new categorization of datasets for frequent itemsets, J Intell Inf Syst (2010) 34:1–19

[15] Aurangzeb Khan , Baharum B. Bahurdin, Khairullah Khan, "An Overview of E-Documents Classification" 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) IACSIT Press, Singapore

[16] Ethiopian Radio and Television Agency website: http://www.erta.gov.et

[17] Reporter Online Amharic Magazine: http://www.ethiopianreporter.com/

[18] National Meteorology Agency: http://www.ethiomet.gov.et/

[19] Ethiopian Orthodox Tewahedo Church Sunday Schools Department Mahibere Kidusan USA Center: http://www.mahiberekidusan.org/

[20] Deutsche Welle: http://www.dw.de/dw/0,,11646,00.html

[21] Voice of America: http://www.voanews.com/amharic/news/

[22] Ministry of Health: http://www.moh.gov.et/Amharic/Pages/index.aspx

[23] Ministry of Education: http://www.moe.gov.et/AMHARIC/Pages/index.aspx

[24] Ethiopian Culture and Tourism Ministry: http://www.tourismethiopia.gov.et/Amharic/Pages/Home.aspx

[25] Addis Ababa Culture and Tourism Bureau: http://www.addisculturetourism.gov.et/am/addis-ababa-ethiopia.html

[26] Md.Ahsan-ul Morshed, Towards the automatic categorization of documents in user-generated categorizations, Technical Report # DIT-06-001

[27] Google search engine: http:// www.google.com

[28] Yahoo search engine: http:// www.yahoo.com

[29] Sebastiani, F.: Text Categorization. In Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press, Southampton, UK, 2005, pp109-129.

[30] Document classification; http://en.wikipedia.org/wiki/Document_categorization, (Accessed on May 2, 2012)

[31] Shailja, Thesis submitted in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science & Engineering, Thapar University, Patiala, ying Web Services With and Without Association Rules

[32] Amharic language, http://en.wikipedia.org/wiki/Amharic_language, (Accessed on May 2, 2012)

[33] Akash Saxena, Differential Virtual Support Algorithm for Association Rule Mining, Dr B R Ambedkar National Institute of Technology

[34] Zelalem Sintayehu, Automatic Amharic news Categorization, A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Information Science

[35] Ge'ez script , http://www.omniglot.com/writing/ethiopic.htm, (Accessed on March 12, 2012)

[36] The Ethiopic Writing System: a Profile, http://www.spellingsociety.org/journals/ j19/ethiopic.php (Accessed on March 12, 2012)

[37] Atelach Alemu Argaw , Lars Asker: An Amharic Stemmer : Reducing Words to their Citation Forms, Department of Computer and Systems Sciences Stockholm University/KTH, Sweden

[38] About the Unicode Standard, http://www.unicode.org/standard/standard.html, (Accessed on May2, 2012)

[39] Madaadi, Pratima, Text Categorization Based on Apriori Algorithm, University of Nevada, (2009), USA

[40] CLIFTON Phua, Vincent Lee, Kate Smith & Ross Gayler, A Comprehensive Survey of Data Mining-based Fraud Detection Research

[41] Taiwo Oladipupo Ayodele, Types of Machine Learning Algorithms, University of Portsmouth United Kingdom

[42] Daniel Josiah-Akintonde, Expert Systems: Final (Research Paper) Project, Transforming Quantitative Transactional Databases into Binary Tables for Association Rule Mining Using The Apriori Algorithm

[43] Inverted indexes: Types and techniques, Ajit Kumar Mahapatra, Sitanath Biswas, Information Technology, ITER, Siksha ‘O’ Anusandhan University

[44] Fabrizio Sebastiani,' Text Categorization', University of Padova, Italy, 2005.http://nmis.isti.cnr.it/sebastiani/Publications/TM05.pdf. (accessed on May 17, 2012)

[45] Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1–47, 2002.

[46] Supervised Learning for Automatic categorization of Documents using Self-. Organizing Maps. Dina Goren-Bar, Tsvi Kuflik, Dror Lev.

[47] Christoph Goller, Joachim Löning, Thilo Will, Werner Wolff, Automatic Document Classification: A thorough Evaluation of various Methods

[48] The Relationship between Recall and Precision, Michael Buckland* and Fredric Gey School of Library and Information Studies, University of California, Berkeley, Berkeley, CA 94720

[49] Ethiopian Video grabbing website http://hahutube.com

[50] Dimension Reduction in Text categorization with Support Vector Machines, Hyunsoo Kim, Peg Howland, Haesun Park, Department of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E., 4-192 EE/CS Building, Minneapolis MN 55455, USA

[51] An Efficient Algorithm for Closed Itemset Mining by Mohammed J. Zaki and Ching-Jui Hsiao

[52] McNicholas, T.B. Murphy M. O'Regan a Department of Statistics, Trinity College Dublin, Ireland, Standardising the Lift of an Association Rule

[53] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, Prabhakar Raghavan; IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA; Scalable feature selection, categorization and signature generation for organizing large text databases into hierarchical topic taxonomies

[54] Cowell, J. Hussain, F. Department of Computer Science, De Montfort Univ., Leicester, UK, Amharic character recognition using a fast signnature based algorithm

[55] Abera Molla, Ethiopian Computers and Software, http://www.ethiopic.com/ (Accessed on August 21, 2012)

[56] Girma Awgichew Demeke, Addis Ababa University, LINCOM Studies in Afroasiatic Linguistics Vol. 28, 29; The Origin of Amharic.

[57] Python Text Processing with NLTK 2.0 Cookbook.pdf , http://python.6.n6.nabble.com/ (accessed on August 24, 2012)

[58] Kenneth W. Church, William A. Gale; AT&T Bell Laboratories, Murray Hill, NJ, USA 07974; Inverse Document Frequency (IDF): A Measure of Deviations from Poisson

[59] S. B. Kotsiantis, Department of Computer Science and Technology, University of Peloponnese, Greece, End of Karaiskaki, 22100 , Tripolis GR., Supervised Machine Learning: A Review of categorization Techniques

[60] POWERS, D.M.W. Journal of Machine Learning Technologies ISSN: 2229-3981 & ISSN: 2229-399X, Volume 2, Issue 1, 2011, pp-37-63; Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation

[61] https://developers.google.com/youtube/2.0/developers_guide_protocol_api_query_parame ters#Searching_for_Videos (Accessed on February 21, 2013)

[62] Lior Rokach, Department of Industrial Engineering, Tel-Aviv University; Decision Trees

[63] M.Sathish, S.Vydehi, Department of Computer Science, Dr. Sns Rajalakshmi College of Arts and Science; A Literature Survey on association rule using apriori algorithm in various fields.

Annex 1. List of basic Amharic Characters

1st	2nd	3rd	4th	5th	6th	7th
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
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
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ

ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ፘ	ፙ	ፚ	፛	፜	፝	፞
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

Annex 2. List of 400 characters found in the training dataset.

!
#
$
&
>
"
'
(
)
*
,
-
.
/
:
;
=
?
a
•
•
‧
‧‧
‧
‧

‧
‧-
‧‧
•
•
Š
á
š
£
¤
¥
¨
é
«
®
°
±
²
³
÷
»
½
ž
µ

Œ
Â
€
™
•
,
,,
,,
†
•
…
‰
‹
›
“
‘
–
—
’
B
c
d
e
~

ˆ
F
g
H
i
Θ
≅
™
♣
∃
)
J
k
l
M
n
o
P
Q
r
s
t
u

V
W
x
z
[
]
|
f
‧
‧
0
1
2
3
4
5
6
7
8
9
‧
‧
‧
‧

ነ
ኝ
አ
ኡ
ኣ
ኤ
እ
ኦ
ኧ
ከ
ኩ
ኪ
ካ
ኬ
ክ
ኮ
ኰ
ኲ
ኳ
ኸ
ኹ
ኺ
ኻ
ኼ
ኽ
ኾ
ዀ
ዄ
ዅ

ኵ
ዎ
ዏ
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
ዬ
ይ
ዮ
ዯ
ደ
ዱ
ዲ
ዳ
ዴ
ድ
ዶ
ዷ
ዸ
ዹ
ዺ
ዻ
ዼ
ዽ
ዾ
ዿ
ጀ
ጁ
ጂ
ጃ
ጄ
ጅ
ጆ
ጇ
ገ
ጉ
ጊ
ጋ
ጌ
ግ
ጎ
ጏ
ጐ
጑
ጒ
ጓ
ጔ
ጕ
ጘ
ጙ
ጚ
ጛ
ጜ
ጝ
ጞ
ጟ
ጠ
ጡ
ጢ
ጣ
ጤ
ጥ
ጦ
ጧ
ጨ
ጩ
ጪ
ጫ
ጬ
ጭ
ጮ
ጯ
ጰ
ጱ
ጲ
ጳ
ጴ
ጵ
ጶ
ጷ
ጸ
ጹ
ጺ
ጻ
ጼ
ጽ
ጾ
ጿ
ፀ
ፁ
ፂ
ፃ
ፄ
ፅ
ፆ
ፇ
ፈ
ፉ
ፊ
ፋ
ፌ
ፍ
ፎ
ፏ
ፐ
ፑ
ፒ
ፓ
ፔ
ፕ
ፖ
ፗ
ፘ
ፙ
ፚ
፛
፜
፝
፞
፟
፠
፡
።
፣
፤
፥
፦
፧
፨
፩
፪
፫
፬
፭
፮
፯
፰
፱
፲
፳
፴
፵
፶
፷
፸
፹
፺
፻
፼
፽
፾
፿

ሦ
ረ
ሩ
ሪ
ራ
ሬ
ር
ሮ
ሯ
ሰ
ሱ
ሲ
ሳ
ሴ
ስ
ሷ
ሸ
ሹ
ሺ
ሻ
ሼ
ሽ
ሾ
ሿ
ሌ
ሜ
ሃ

ለ
መ
ሉ
ሐ
ሑ
ሒ
ሓ
ሔ
ሕ
ሖ
ሗ
መ
ሙ
ሚ
ማ
ሜ
ም
ሞ
ሟ
ሠ
ሡ
ሢ
ሣ
ሤ
ሥ
ሦ
ሧ
ረ
ሩ
ሪ
ራ
ሬ
ር
ሮ
ሯ
ሰ
ሱ
ሲ
ሳ
ሴ
ስ
ሷ
ሸ
ሹ
ሺ
ሻ
ሼ
ሽ
ሾ
ሿ
ቀ
ቁ
ቂ
ቃ
ቄ
ቅ
ቆ
ቇ
ቈ
ቊ
ቋ
ቌ
ቍ
ቐ
ቑ
ቒ
ቓ
ቔ
ቕ
ቖ
ቘ
ቚ
ቛ
ቜ
ቝ
በ
ቡ
ቢ
ባ
ቤ
ብ
ቦ

ቧ
ቨ
ቩ
ቪ
ቫ
ቬ
ቭ
ተ
ቱ
ቲ
ታ
ቴ
ቶ
ቷ
ቸ
ቹ
ቺ
ቻ
ቼ
ች
ቾ
ቿ
ኀ
ኁ
ኂ
ኃ
ኄ
ኅ
ኆ
ኇ
ኈ
ኊ
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

ቓ
ቔ
ቕ
ቖ
ቘ
ቚ
ቛ
ቜ
ቝ
ቐ
ቑ
ቒ
ቓ
ቔ
ቕ
ቖ
ቘ
ቚ
ቛ
ቜ
ቝ
በ
ቡ
ቢ
ባ
ቤ
ብ
ቦ
ቧ
ቨ
ቩ
ቪ
ቫ
ቬ
ቭ
ተ
ቱ
ቲ
ታ
ቴ
ቶ
ቷ
ቸ
ቹ
ቺ
ቻ
ቼ
ች
ቾ
ቿ
ኀ
ኁ
ኂ
ኃ
ኄ
ኅ
ኆ
ኇ
ኈ
ኊ
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

ዮ
ደ
ዱ
ዲ
ዳ
ዴ
ድ
ዶ
ዷ
ዸ
ዹ
ዺ
ዻ
ዼ
ዽ
ዾ
ዿ
ጀ
ጁ
ጂ
ጃ
ጄ
ጅ
ጆ
ጇ
ገ
ጉ
ጊ
ጋ
ጌ
ግ
ጎ
ጏ
ጐ
ጒ
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
ጷ
ጸ
ጹ
ጺ
ጻ
ጼ
ጽ
ጾ
ጿ
ፀ
ፁ
ፂ
ፃ
ፄ
ፅ
ፆ
ፇ
ፈ
ፉ
ፊ
ፋ
ፌ
ፍ
ፎ
ፏ
ፐ
ፑ
ፒ
ፓ
ፔ
ፕ
ፖ
ፗ
ፘ
ፙ
ፚ
፛
፜
፝
፞
፟
፠
፡
።
፣
፤
፥
፦
፧
፨
፩
፪
፫
፬
፭
፮
፯
፰
፱
፲
፳
፴
፵
፶
፷
፸
፹
፺
፻
፼
፽
፾
፿

Sample Amharic Documents Categorization (ADC) Program Code

```
        // A function used to check input values to protect sql injection
function checkValues($value)
    {
        // Use this function on all those values where you want to check for both sql injection
    and cross site scripting
        //Trim the value
        $value = trim($value);

       // Stripslashes
       if (get_magic_quotes_gpc()) {
                $value = stripslashes($value);
       }

        // Convert all &lt;, &gt; etc. to normal html and then strip these
        $value = strtr($value,array_flip(get_html_translation_table(HTML_ENTITIES)));

        // Strip HTML Tags
        $value = strip_tags($value);

       // Quote the value
       $value = mysql_real_escape_string($value);
       $value = htmlspecialchars ($value);
       return $value;

    }
//decoding string
function _utf8_decode($string)
{
  $tmp = $string;
```

```
  $count = 0;
  while (mb_detect_encoding($tmp)=="UTF-8")
  {
    $tmp = utf8_decode($tmp);
    $count++;
  }

  for ($i = 0; $i < $count-1 ; $i++)
  {
    $string = utf8_decode($string);

  }
  return $string;

}
```

//str_split — Convert a string to an array

```
function str_split_unicode($str, $l = 0) {
    if ($l > 0) {
        $ret = array();
        $len = mb_strlen($str, "UTF-8");
        for ($i = 0; $i < $len; $i += $l) {
            $ret[] = mb_substr($str, $i, $l, "UTF-8");
        }
        return $ret;
    }
    return preg_split("//u", $str, -1, PREG_SPLIT_NO_EMPTY);
}
```

//check the presence of a character in the database

```
function charexist($char)
{
```

```
$sql3="SELECT * FROM `tb_char` WHERE `charList` LIKE '".$char."' LIMIT 0 , 1";
$rsd3 = mysql_query($sql3);
$num3=mysql_numrows($rsd3);
return ($num3);
}
//check the presence of a word in the database
function wordexist($str2)
{
$sql3="SELECT *
FROM `tb_words`
WHERE `word` LIKE '".$str2."'
LIMIT 0 , 1";
$rsd3 = mysql_query($sql3);
$num3=mysql_numrows($rsd3);
return ($num3);
}
```

//check white sapace

```
function is_whitespace($string){
    //    Return FALSE if passed an empty string.
    if($string == "") return FALSE;

    $char    =    ord($string);

    //    Control Characters
    if($char < 33)                    return TRUE;
    if($char > 8191 && $char < 8208)    return TRUE;
    if($char > 8231 && $char < 8240)    return TRUE;

    //    Additional Characters
    switch($char){
```

```
            case 160:    // Non-Breaking Space
            case 8287:    // Medium Mathematical Space
                return TRUE;
                break;
        }
        return FALSE;
    }
//check  the sound of a word and return string key
function MakeSoundEx($stringtomakesoundexof)
{
  $temp_Name = strtoupper($stringtomakesoundexof);
  $SoundKey = array(1=>"BPFV", "CSKGJQXZ", "DT", "L", "MN", "R", "AEHIOUWY");
  $temp_Last = "";
  $temp_Soundex = substr($temp_Name, 0, 1);

  for ($x = 1; $x <= sizeof($SoundKey); $x++)
    for ($i = 0; $i < strlen($SoundKey[$x]); $i++)
      if ($temp_Soundex == substr($SoundKey[$x], $i - 1, 1))
        $temp_Last = (string)($x==7?"":$x);

  for ($n = 1; $n < strlen($temp_Name); $n++)
    if (strlen($temp_Soundex) < 4)
    {
      for ($x = 1; $x <= sizeof($SoundKey); $x++)
        for ($i = 0; $i < strlen($SoundKey[$x]); $i++)
          if (substr($temp_Name, $n-1, 1)==substr($SoundKey[$x], $i-1, 1))
          {
            if($x<7 && $temp_Last!=(string)$x)
              $temp_Soundex = $temp_Soundex.$x;
            $temp_Last = (string)($x);
```

```
        }
    }

  return $temp_Soundex . str_repeat("0", 4-strlen($temp_Soundex));
}
```
//convert amharic word in to english equivalent
```
function amhaWordEngConvert ($text)
{
$str[]="";
$engWord="";
$str=str_split_unicode($text, 1);
for($i=0;$i<count($str);$i++)
{
//$str[$i]
$sql="SELECT `sound` FROM `tb_char` WHERE `charList` LIKE '".$str[$i]."' LIMIT 0 , 1";
$res = mysql_query($sql);
if($rs = mysql_fetch_array($res)) {
$eng = $rs['sound'];
$engWord=$engWord.$eng;
}
}
return ($engWord);
}
```
//prefix remover
```
function prefixRemover($text)
{
$sub3=substr($text, 0, 4);
$sub4=substr($text, 4);
if(strlen($text)>4 &&($sub3=="ENDE" ||$sub3=="ENDI" ||$sub3=="WEDE" )){
$text=($sub4);
```

```
}
$sub5dd=substr($text, 0, 4);
$sub6dd=substr($text, 3);
if(strlen($text)>4 &&($sub5dd=="SIYA" )){
$text=($sub6dd);
}
$sub5=substr($text, 0, 3);
$sub6=substr($text, 3);
if(strlen($text)>4 &&($sub5=="EYE" )){
$text=($sub6);
}

$sub=substr($text, 0, 2);
$sub2=substr($text, 2);

if(strlen($text)>4 &&($sub=="YE"||$sub=="KE"||$sub=="LE"||$sub=="BE")){
$text=($sub2);
}
return($text);
}
```
//postfix remover
```
function postfixRemover($text)
{
$sub30c=substr($text, 0,strlen($text)-4);
$sub30b=substr($text, strlen($text)-4);
$sub8a=substr($text, 0,strlen($text)-6);
$sub8=substr($text, strlen($text)-5);
if(strlen($text)>5&&($sub30b=="OCHU"||$sub8=="GNAW"))
{
$text= ($sub30c);
```

```
}

$sub5=substr($text, strlen($text)-2,1);
$sub6=substr($text, strlen($text)-1);
$sub7=substr($text, 0,strlen($text)-1);
if(strlen($text)>4&&($sub5!="A"&&$sub5!="I"&&$sub5!="U"&&$sub5!="E"&&$sub5!="O
    ")&&($sub6=="N"||$sub6=="M")&&$sub5!="G")
{
$text= ($sub7);
}
$sub=substr($text, strlen($text)-3);
$sub2=substr($text, 0,strlen($text)-3);

if(strlen($text)>4
    &&($sub=="NNA"||$sub=="OCH"||$sub=="GNA"||$sub=="GNA"||$sub=="WAN")){
$text=($sub2);
}
$sub3=substr($text, strlen($text)-2);
$sub4=substr($text, 0,strlen($text)-2);
if(strlen($text)>4
    &&($sub3=="UN"||$sub3=="NA"||$sub3=="UM"||$sub3=="NU"||$sub3=="UN"||$sub3=
    ="LN")){
$text=($sub4);
}

$sub8a=substr($text, 0,strlen($text)-6);
$sub8=substr($text, strlen($text)-5);

if(strlen($text)>6&& $sub8=="ACHEW")
{
```

```
$text= ($sub8a);
}
$sub30=substr($text, 0,strlen($text)-2);
$sub30a=substr($text, strlen($text)-3);

if(strlen($text)>4&&$sub30a=="ECH")
{
$text= ($sub30);
}

$sub30e=substr($text, 0,strlen($text)-3);
$sub30d=substr($text, strlen($text)-3);

if(strlen($text)>4&&$sub30d=="OCH")
{
$text= ($sub30e);
}

$sub31=substr($text, strlen($text)-1);
$sub31a=substr($text, 0,strlen($text)-1);
if(strlen($text)>3&&$sub31=="W")
{
$text= ($sub31a);
}
return($text);
}
//string position checker
function stringrpos($haystack,$needle,$offset=NULL)
{
   return strlen($haystack)
```

```
        - strpos( strrev($haystack) , strrev($needle) , $offset)
        - strlen($needle);
}
//return last news/document Id
function lastnewsId()
{
$sql="SELECT id
FROM  `tb_news_store`
ORDER BY  `tb_news_store`.`id` DESC
LIMIT 0 , 1";
$rsd = mysql_query($sql);
if($rs = mysql_fetch_array($rsd)) {
$id = 0+$rs['id'];
return ($id);
}
}
function lastnewsId2()
{
$sql="SELECT id
FROM  `tb_news_test`
ORDER BY  `tb_news_test`.`id` DESC
LIMIT 0 , 1";
$rsd = mysql_query($sql);
if($rs = mysql_fetch_array($rsd)) {
$id = 0+$rs['id'];
return ($id);
}
}
//stopwords counts
function stopWordSync($str2)
```

```
{
//$val=0;
if(wordexist($str2))
{
$sql="SELECT id FROM  `stopword` WHERE    `word` LIKE  '$str2'";
$rsd = mysql_query($sql);
$num=mysql_numrows($rsd);
//$status = $rs['status'];
return ($num);
}
}
```
//check valid characteres
```
function isValidChar($txt)
{
$sql="SELECT `id` FROM `tb_char` WHERE `charList` LIKE '".$txt."' AND `validity` =1
      LIMIT 0 , 1";
$rsd = mysql_query($sql);
$num=mysql_numrows($rsd);
return ($num);
}
```