ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SCIENCE

AUTOMATIC CATEGORIZATION OF AMHARIC NEWS TEXT: A MACHINE LEARNING APPROACH

By

SURAFEL TEKLU WELDESELLASSIE

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

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SCIENCE

AUTOMATIC CATEGORIZATION OF AMHARIC NEWS TEXT: A MACHINE LEARNING APPROACH

BY
SURAFEL TEKLU WELDESELLASSIE

Signature of the Board of Examiners for Approval

Ato Getachew Jameneh, Chairman, Examining Board _____

W/ro Rahel Bekele, Advisor _____

W/ro Woinshet Abdele, Advisor _____

Ato Workshet Lamnew, Advisor _____

Prof. K. N. Murthy, External Examiner _____

ACKNOWLEDGEMENT

I would like to thank my advisors W/o Rahel Bekle, W/o Woinshet Abdela and Ato Workshet Lamenew for their constructive comments and encouragement on my work. I am also grateful to all staff of the Faculty of Informatics for their support during my stay at the faculty.

This research couldn't come into being without the help of ENA (Ethiopian News Agency) and my thanks goes to the management and staff of ENA for giving me access to the Amharic news database.

I am grateful to Cappuccin Francisco Institute of Philosophy and Theology (CFIPT) and Admass College for their motivation and payment of the tuition fee for the program.

I am also grateful to my classmates for their unreserved help and my special thanks goes to Henock Luelseged for his strong technical support during my research.

Finally, I thank all my families and friends for their continuous motivation and encouragement during my stay in the university.

TABLE OF CONTENTS

List of Tables

List of Figure

List of Appendices

ABSTRACT

Currently newspaper companies and news agencies in Ethiopia are implementing a manual categorization system to categorize Amharic news articles in their day-to-day activities (although they are using computer system to store and dispatch information).

The objective of this 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 this research. To come up with good results 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 this 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. This research indicated that Naïve Bayes is more applicable to automatic categorization of Amharic news items.

The result of this research is promising. Nevertheless, additional works are recommended in order to come up with good result.

Keywords: Text categorization, machine Learning, naïve Bayes, K Nearest Neigbor

CHAPTER ONE

INTRODUCTION

1.1. Background

Humans use classification techniques to organize things in various activities of their life. People make their own judgment to classify things in their every day life – they classify things based on similarities or likeliness of color, size, concept, ideas, and subject and so on. For instance, in information centers, books, pamphlets, journals and, etc are classified based on their subject content; supermarkets put similar products together. As also stated by Zelalem (2001) quoting Kumar (1999), human progress would be impossible without classification.

The need to classify information resources has become an important issue as the production of such resources increase dramatically from time to time. More specifically, for the last 6 decades, there is a great increase in the production of information. Manuscripts, newspapers, journals, magazines, thesis and dissertations are available in different formats such as text, audio, video, and graphics.

Text is the main form of communicating knowledge. Technically speaking, text is any string of language, usually one that is more than one sentence long (Russel & Norvig, 1995). Text is composed of symbols from a finite alphabet. Text has been created everywhere, in many forms (paper and electronic) and languages. We use the term document to denote a single unit of information, typically text in a digital form, but it can also include other media. A document can be a complete logical unit, like a research article, a book or a manual. It can also be part of a larger text, such as a paragraph or a sequence of paragraphs (also called a passage of text), an entry in a dictionary, a judge's opinion on a case, the description of an automobile part, etc.

Furthermore, a document can be any physical unit, for example a file, an email, or a World Wide Web page (Baeza-Yates & Ribeiro-Neto, 1999).

There are numerous text documents available in electronic form. Academic publication and journals are becoming available in electronic form. More are becoming available constantly. The web itself contains over a billion documents. Millions of people send e-mail every day. These collections and many others represent a massive amount of information (Rennie, 1999; Han et al., 1999).

However, seeking information of one's need in this huge collection requires organization. Especially in a system where there is large collection of documents, retrieval of a given document or set of documents is possible if the collection is organized systematically. Many web sites offer a hierarchically organized view of the Web. E-mail clients offer a system for filtering e-mail. Academic communities often have a Web site that allows searching on papers and shows an organization of papers. Nowadays, news items are produced every day in digital devices and organized in some order (Rennie, 1999). However, most of the time text classification process is done manually which brings about enormous costs both time and money wise. In other words, organizing documents by hand or creating rules for filtering is painstaking and labor-intensive. Therefore, automatic classification systems are very desirable since they minimize such problems (Neumann & Schmeier, n.d; Rennie, 1999).

The term automatic text classification is sometimes used in the literature to mean two views. The first view is the automatic identification of a set of categories and the grouping of documents under them. This is referred to as text clustering. The second view refers to the automatic

assignment of documents to a predefined set of categories, which is referred to as text classification (Sebastiani, 2002; Rasmussen, 1992). This research, however, is concerned with the second view.

For many decades classification techniques are applied in information retrieval system in order to facilitate access to, and use of the system (Van Uden, n.d).[1] Especially, in the last 10 years content-based document management tasks (collectively known as Information Retrieval) have gained a prominent status in the information systems field, due to the increased availability of documents in digital form and the ensuing need to access them in flexible ways. Text categorization (also known as text classification, or topic spotting), the activity of labeling natural language texts with thematic categories from a predefined set, is one such task (Sebastiani, 2002).[2]

As stated by Sebstiaini (2002), text classification dates back to the early 1960's, but only in the early 1990's did it become a major subfield of the information systems discipline, thanks to increased applicative interest and to the availability of more powerful hardware.

Until the late 1980's the most popular approach to text classification, at least in the "operational" (i.e., real-world applications) community, was knowledge engineering (KE): an expert system, consisted in manually defining a set of rules encoding expert knowledge on how to classify documents under the given categories. In the 1990's this approach has increasingly lost

1 Information retrieval (IR) system is a system that is capable of storage, retrieval and maintenance of information (Salton & McGill, 1983; Van Rijsbergen, 1979).

[2] In this research, text categorization and text classification are used interchangeably.

popularity (in the research community) in favor of the machine learning (ML) paradigm (Sebstiani, 2002).

Machine learning is a general inductive process that automatically builds an automatic text classifier by learning from a set of pre-classified documents, the characteristics of the categories of interest (Sebstiani, 2002). Machine learning techniques are relatively new learning techniques (Yang & Liu, 1999). In the areas of machine learning, extensive research has been done to test the possibility of automatic classification of documents. This approach is economically and qualitatively effective to that achieved by manual classification systems. Moreover, the advantages of machine learning over the knowledge engineering are a very good effectiveness, considerable savings in terms of expert labor power since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories (Sebstiani, 2002).

These days text categorization is a discipline at crossroads of ML and information (IR), and as such it shares a number of characteristics with other tasks such as information/knowledge extraction from texts and text mining. There is still considerable debate on where the exact border between these disciplines lies, and the terminology is still evolving (Ibid). The concern of this paper is, therefore, to make use of applying machine learning approach in text categorization.

Text categorization has been extensively studied by the ML community, as it is a classic example of a supervised learning problem. This involves computer “learning” - a categorization function from labeled training data provided by a supervisor. The function takes a document and returns a

set of categories to which it is likely to belong. It is hoped that the categorizer's approximation of the function will improve with the amount of training data supplied to it (Crimmins, 2001).

An important issue in text categorization is how documents are represented, and how features can be extracted from them, which can be used for categorization. A standard document representation is a vector of term occurrences, as used in the information retrieval field. Feature selection is used to extract a set of features, which will aid in categorizing a document. For example, the most significant terms in the document (defined using word frequencies) may be used as an input to the categorizer. Stop words (non-information bearing) are often removed. Further information may be provided by using phrases or term co-occurrence statistics (Crimmins, 2001).

Since 1960's text classification has been used for a number of different applications: automatic indexing for Boolean information retrieval systems, document organization and text filtering (e.g., news items classification, deciding to what folder an e-mail message should be directed and deciding to which news group a news article belongs), word sense disambiguation and hierarchical categorization of web pages (Sebstiani, 2002).

Other applications include speech categorization by means of a combination of speech recognition and Text classification, multimedia document categorization through the analysis of textual captions, author identification for literary texts of unknown or disputed authorship, language identification for texts of unknown language, automated identification of text genre, and automated essay grading, assigning individuals to credit status on the basis of financial and other

personal information, in preliminary diagnosis of a patient's disease in order to select immediate treatment while awaiting definitive test results (Michell, 1997; Sebstiani, 2002).

According to Crimmins (2001), there are many different machine learning techniques and algorithms, which have been used for text categorization. Examples include: Support vector machines (SVMs), Decision trees, Decision rules, neural networks, Rocchio relevance feedback, Nearest neighbor classifiers, regression models and Bayesian learning methods such as belief networks and Naïve Bayes.

Bayesian learning is a statistical classifier that provides a probabilistic approach to inference, based on the assumption that the quantities of interest are governed by probability distribution and the optimal decision; and that optimal decision can be made by reasoning about these probabilities together with observed data. All Bayesian learning methods are based on Bayes theorem (Mitchell, 1997).

Naïve Bayes, also called Naïve Bayes Classifier (NBC), is one of the known machine-learning techniques used for text classification. Many researches provide a detailed study comparing the naïve Bayes classifier to other learning algorithms, including decision tree and neural network algorithms. These researchers show that the naïve Bayes classifier is competitive with these other learning algorithms in many cases and that in some cases it outperforms these other methods. In particular, its application to the complex problem of learning to classify text documents such as electronic news articles, newsgroups and e-mail identification. For such learning tasks, the naïve Bayes classifier is among the most effective algorithms known. (Mitchell, 1997; Witten & Frank, 2000).

Keswani (2002) & Vaithyanathan et al. (2000) also described that Naïve Bayes is commonly used in practice and is a focus of research in text classification. Naive Bayes approaches form a very popular class of models used in machine learning applications (Lewis, 1998; McCallum & Nigam, 1998). Moreover, Witten & Frank (2000) added that naive Bayes provides a simple approach, with clear semantics, to representing, using, and learning probabilistic knowledge. Impressive results can be achieved using it. It assumes that all attributes of the examples are independent of each other (the parameters for each attribute can be learned separately) given the context of the class - that is why it is called "naive". This assumption greatly simplifies learning, especially when the number of attributes is large. While this assumption clearly is false in most real-world tasks, it often performs classification very well.

K nearest Neighbor (kNN) is an example of instance-based learning. It is a traditional statistical pattern recognition which has been studied extensively for text categorization. The kNN is quite simple, i.e., given a test document, the system finds the k nearest neighbors among the training documents, and uses the categories of the k neighbors to weight the category candidates. The calculation is done based on Euclidean distance. In many experiments it performs well than the other algorithms such as C4.5, RIPPER, Naïve-Bayes, etc. (Han et al., 1999; He, n.d; Yang & Liu, 1999).

This research, therefore, applies naïve Bayes and kNN classifiers for Amharic news items classification as they are well known in text categorization tasks.

1.2 Statement of the Problem and its justification

The public needs to be updated on important public events such as actions of governments, social trends, education, and international relationships, which are often referred to as hard news. Other

news types include gossip items about celebrities, off beat incidents, sensational crime cases, and sudden happenings, such as a jury verdict, or a parliamentary decision etc., which present public interest (ENA, 1993a).

News is a prompt, "bottom line" recounting of factual information about events, situations, and ideas (including opinions and interpretations) calculated to interest an audience and help people cope with themselves and their environment (ENA, 1993a). For this purpose any government or a private agency establishes an organization that facilitates the free expression of opinions and views among the people.

In order to make easy access and timely information, news items should be organized in systematic manner. The greater our ability to store information, the more attention must be paid to the problem of organizing and retrieving it. Traditionally, human experts engaged in classifying news items manually into their predefined classes. In the last few years, automatic text classification systems have proven to be just as accurate, correctly categorizing over 90% of the news stories. They are also far faster and more consistent, so there has been a switch from manual to automated systems (Russel & Norvig, 1995).

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

- Experts could classify news items manually; however, such classification task is tedious and may be full of errors. When two human experts decide whether to classify document d_j under category c_i, they may disagree, and this in fact happens with relatively high

frequency (Sebastiani, 2002). Therefore, developing an automatic news text classification system is an important step to assist experts in addressing such problems.

- The timeliness characteristic of news items is achieved by classifying news without the involvement of specialists.
- It would be possible to store old news for further retrieval.

Experiments have been conducted in automatic news text classification using machine learning techniques for English language (e.g. Joachims, n.d; Yang & Liu, 1999, etc), and other foreign languages, e.g., Chinese language by He et al. (n.d) and Indian language (Bandyopadhyay, 2001).

Although more than 80 languages are spoken in Ethiopia, Amharic is the working language of the Federal Government of Ethiopia. Amharic is the first language for more than 17 million and second language for over 5 million people and it is one of the widely used media in Ethiopia (ECSA, Ethiopian Central Statistics Authority, 1998). It is the most used languages for text/document storage and media purposes in the country. However, research work for automatic Amharic text categorization in general and news items categorization in particular is still in its infancy. To the researcher's knowledge, one research in the area of automatic text classification is conducted by Zelalem (2001). Zelalem conducted such a research in the case of the Ethiopian News Agency taking 1, 481 news items as training and test set where the news items were classified into three classes. He concluded that the result he got was promising – 90.5 % average accuracy. He also recommended that more research and development of Amharic automatic text

classification should be done. His research is based on the information retrieval techniques – vector space model.[1]

According to Chen (2003), information retrieval has attracted significant attention on the part of researchers in information and computer science over the past few decades. In the 1980s knowledge-based techniques also made an impressive contribution to "intelligent" information retrieval and indexing. More recently, researchers have turned to other newer artificial-intelligence based inductive learning techniques, i.e., machine learning. Text categorization, which is also found to be good in IR, is amenable to machine learning techniques where IR is not (Russel & Norvig, 1995).

Currently newspaper companies and news agencies in Ethiopia are implementing a manual classification system to classify news articles in their day-to-day activities (although they are using computer system to store and dispatch information). As mentioned in the previous paragraphs the manual classification system is time consuming and inconsistent.

With this in mind, the researcher is initiated to do a research on automatic Amharic news items classification using a machine learning technique taking more number of news instances and classes into consideration than the previous research done by Zelalem (2001). This variant of technique may obtain an optimal accuracy (Crimmins, 2001).

1.3. Objective of the Study

The general and specific objectives of the research are the following.

[1] Vector space model views documents and queries as vectors in an n-dimensional vector space and use distance as a measure of similarity (Salton, 1983).

1.3.1. General Objective

The general objective of this research is to investigate the application of machine learning techniques to automatic categorization of Amharic news items.

1.3.2. Specific Objectives

The specific objectives of this research are:

- To review literature on the concept of text classification in the area of machine learning
- To collect, prepare, and preprocess news items suitable for automatic classification.
- To build and train models using Naïve Bayes and kNN classifiers
- To test the performance of the models.

1.4. Methodology

The following methods are employed in conducting the research.

1.4.1 Literature Review

In order to get a good understanding of text classification and the Amharic language relevant published documents were reviewed: books, journal articles, previous related research works and electronic publication on the Internet were consulted. Moreover, discussions were conducted with the experts on the problem area.

1.4.2. Data Sources for the Experiment

Out of 11, 238 Amharic news items available in SQL server database 11, 024 of them were used as a source of data for training and testing. Such amounts of news items were obtained after some unnecessary recorded news items were deleted by the researcher (see section 5.3). The news items were manually classified by experts into 16 classes. These news items were collected from

Ethiopian News Agency (ENA). ENA is a government information center that dispatches news for broadcast over television and radio for local consumption. The news articles[1] were written in visual Ge'ez Amharic font.

1.4.3. Experimentation Method

The data which is stored in SQL Server database was imported into Ms-Access and converted to text form in a suitable manner for preprocessing and then the data was preprocessed through two phases. In the first phase, Delphi 5 programming language was used to remove extraneous characters from the collection. Moreover, so as to make the documents comfortable to the tool used in this research called Rainbow, this programming language was used to create each news item as a file and to group to its related category. Delphi 5 was used because it is familiar to the researcher. Because of its availability Rainbow and Red hat Linux were used for the experiment. Rainbow, which runs on UNIX or Linux, is a widely used software program that performs statistical text classification (Han et al., 1999; McCallum, 1998).

In the second phase, the preprocessing was done by the classification tool used; i.e., stop-words in Amharic language as well as in news items were removed from the collection as those words have high frequency throughout the collection. Moreover, those rare words which occur fewer than or equal to 3 times in the collection were removed.[2] From IR, it is assumed that the stop-words and rare words do not discriminate the documents from each other (Salton, 1983). The remaining words after preprocessing were indexed or modeled by the tool; Naïve Bayes and kNN classifiers were then compared for classification.

[1] In this research the terms news items, news texts, news articles, and documents are used interchangeably.
[2] Many authors used to remove words which occur fewer than or equal to 3 times in a collection either as the only form of dimensionality reduction or before applying another more sophisticated form (Sebastiani 2002).

In order to see the effect of the number of data and number of categories on the performance of the classifiers, experiments were done on three, four, seven, and on sixteen categories. These experiments used 33% of the data randomly as a test set and the remaining as training set. The models were trained until good results are obtained and the performance was tested based on the results of the test set preclassified by the experts. Then, the percentage of correct assignments by the system was taken to decide the system's effectiveness. The result is shown using confusion matrix provided by the system as seen in chapter four.

1.5 Significance of the Study

In addition to being an academic exercise to fulfill the requirement for the program, this research is believed to produce results that can indicate the application of a general Amharic automatic news items classifier based on machine learning techniques. The results of the research can be used as an input to the development of full-fledged automatic news items classification of Amharic language for ENA. The output of this thesis can also be used as a starting point for further investigations in the possibilities of automatic news items classification system development for the Amharic (or any other Ethiopian languages that make use of the same Ethiopic alphabets such as Tigrinya, Guragignya) language using machine learning approach. In addition to this, it will be useful as an input to any automatic Amharic text classification with little modification.

1.6 Scope and Limitation of the Study

Due to time constraint to train, test and analyze the results, only two of the algorithms (naïve Bayes and kNN) among the other algorithms available in the tool (Rainbow) are used to test the automatic categorization of Amharic news items.

1.7 Organization of the Thesis

This thesis is divided into six chapters. The first chapter is an introduction to the research environment. Moreover, this chapter also presents statement of the problem, objective of the study and methodology. In chapter two concepts in automatic text categorization and the application areas are discussed. Chapter three highlights machine learning concepts; text representation and techniques followed in this research, naïve Bayes and kNN, are reviewed.

Chapter four is a review of the Amharic language writing system. In this chapter problems in Amharic writing system are also reviewed. The experiments which are the output of this research are discussed and reported in chapter five. Finally, the conclusions drawn from the study and the recommendations are forwarded in Chapter six.

CHAPTER TWO

AUTOMATIC TEXT CATEGORIZATION

2.1 Introduction

With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Automatic categorization schemes can greatly facilitate the process of categorization (Joachims, n.d; Han et al., 1999). In this chapter the views of text categorization, the basic concepts and application of text categorization are discussed.

2.2 Automatic Text Categorization: Views on the definition

The term automatic text categorization is sometimes used in the literature to mean different things. The first concept refers to "the automatic identification of a set of categories and the grouping of documents under them, a task usually called text clustering."(Sebastiani, 2002). This technique simply classifies documents without having a pre-defined categories where the text to be classified. The classification process is then expected to create the classes or categories based on the similarity that exist among the documents. Text clustering is thus a typical example of unsupervised learning. As stated by Russell & Norvig (1995), unsupervised learning is learning where there is no hint at all about the correct outputs.

The second view for the definition of automatic text categorization refers to "the automatic assignment of documents to a predefined set of categories." (Sebastiani, 2002). This concept is also referred to as text categorization. Text categorization is a supervised learning problem (Rasmussen, 1992). Russell & Norvig (1995) described that supervised learning is any situation in which both the inputs and outputs of a component can be perceived.

2.3 Approaches to Text Categorization

We may have four approaches to text categorization: manual, IR, knowledge engineering (KE) and machine learning (ML) approaches (Sebastiani, 2002; Coster, 2002).

Manual text categorization is reading the text or document and assigning categories to it by domain experts (Coster, 2002). In spite of its accuracy, this approach is time consuming. For example, Yahoo, Looksmart, about.com, and Medline are currently using human experts to categorize documents. However, given the fast growth in online document data, this would become more difficult with time (Han et al., 1999).

Automatic text categorization dates back to the early 1960's in the information retrieval field. Since then classification techniques applied in this field in order to facilitate access to, and use of the system. The automatic document indexing for IR systems, which relies on a controlled dictionary is the most prominent example of Boolean systems. In such systems each document is assigned one or more key words or key phrases describing its content, where these key words and key phrases belong to a finite set called controlled dictionary, often consisting of a hierarchical thesaurus. If the entries in the controlled vocabulary are viewed as categories, text indexing is an instance of automatic text categorization (Sebastiani, 2002).

One of the important inventions of artificial intelligence (AI) research is the idea that formally intractable problems can be solved by extending the traditional scheme

Program = algorithm + data

to the more elaborate

Program = algorithm + data + domain knowledge

Applying domain knowledge, encoded in suitable data structure, is fundamental for solving problems of this kind, i.e., the use of knowledge shifts the bottleneck from the program to the knowledge engineer, who has to elicit it from an expert and encode it into the system (Kubat et al., 1996). This technique is called knowledge engineering.

In the 1980's, knowledge engineering technique became the most popular approach for the creation of automatic document classifiers consisting in the manual definition of a classifier by domain experts. It is an expert system capable of taking text classification (TC) decisions. Such an expert system would typically consist of a set of manually defined logical rules, one per category, of type

if (DNF formula) then (category).

A DNF ("disjunctive normal form") formula is a disjunction of conjunctive clauses; the document is classified under category if it satisfies the formula or at least one of the clauses. The most famous example of this approach is the CONSTRUE system (built by Carnegie Group for the Reuters news agency). CONSTRUE gave very good effectiveness: a 90 % "breakeven" result on a subset of Reuters test collection (Hayes et al. 1990; quoted by Sebastiani, 2002).

The drawback of this approach is that the rules must be manually pre-defined by a knowledge engineer with the aid of a domain expert, for instance, if the set of categories is updated, then these two professionals must intervene again, and if the classifier is ported to a completely different domain (i.e., set of categories), a different domain expert needs to intervene and the work has to be repeated from scratch (Sebastiani, 2002).

Moreover, the process of knowledge acquisition and encoding is far from real-world applications. For example, specialists in computer chess know that the brute-force approach[1] has led to more powerful programs than the Artificial methods because the knowledge needed to make the program beat a grandmaster is very difficult to formulate; grandmasters use their experience intuitively and are in most cases, unable to convey it to an artificial intelligence system in the form of production rules or other representational system. Such concepts usually lack precise definition and encoding them in a computer is very difficult. Thus, the alternative solution is to employ a learning system that will acquire such higher-level concepts and/or problem solving strategies through examples in a way analogical to human learning (Kubat et al., 1996) – machine learning.

According to Sebastiani (2002),

> *"The machine learning approach to text categorization has gained popularity in the 1990's and has eventually become the dominant one, at least in the research community. In this approach, a general inductive process (also called the learner) automatically builds a classifier for a category* c_i *by observing the characteristics of a set of documents manually classified under* c_i *or* $\bar{c}_i$ *by a domain expert; from these characteristics, the inductive process gleans the characteristics that a new unseen document should have in order to be classified under* c_i*. In ML terminology, the classification problem is an activity of supervised learning, since the learning process is "supervised" by the knowledge of the categories and of the training instances that belong to them.*

[1] Brute-force programs are written in a heavy-handed, tedious way, full of repetition and devoid of any elegance or useful abstraction (Hosting works, 2000).

The advantages of the ML approach over the KE approach are the engineering effort goes toward the construction not of a classifier, but of an automatic builder of classifiers (the learner). This means that if a learner is (as it often is) available off-the-shelf, all that is needed is the inductive, automatic construction of a classifier from a set of manually classified documents. The same happens if a classifier already exists and the original set of categories is updated, or if the classifier is ported to a completely different domain (Sebastiani, 2000). In general, The ML approach gives high-accuracy classifiers, and is significantly less expensive than manual construction because the algorithm automatically constructs the decision rule itself (Nigam, 2001).

Classifiers built by means of ML techniques nowadays achieve impressive levels of effectiveness, making automatic classification a qualitatively *(*and not only economically) viable alternative to manual classification (Sebastiani, 2000). This thesis follows the machine learning approach to text categorization as reviewed in chapter 3.

2.4 Automatic Text Categorization: Basic Concepts

As stated in chapter one, over the past decade, there has been an explosion in the availability of electronic information. As the availability information increases, the inability of people to assimilate and profitably utilize such large amounts of information becomes more and more evident. The most successful paradigm for organizing this mass of information, making it comprehensible to people, is by categorizing the different documents according to their topic (Koller & Sahami, 1997; Joachims, n.d).

Automated text categorization (TC) is defined as assigning pre-defined category labels to new documents based on the likelihood suggested by a training set of labeled documents (Yang and Liu 1999; Crimmins, 2001). It is the task of assigning a value to each pair (d_j, c_i) ε D x C where D is a domain of documents and C= { c_1,…,c|C|} is a set of predefined categories. The goal of text categorization is the classification of documents into a fixed number of predefined categories. Each document can be in multiple, exactly one, or no category at all. As mentioned earlier, using machine learning, the objective is to learn classifiers from which they perform the category assignments automatically. Since building text classifiers by hand is difficult and time-consuming, it is advantageous to learn classifiers from examples (Crimmins, 2001; Joachims, n.d).

The text categorization problem can be reduced to a set of binary classification problems one for each category – where for each one wishes to determine a method for predicting in-class versus out-of-class membership (Zhang & Oles 2000).

The practical reason for the need of automated categorization is that the sheer scale of resources available on electronic form and their ever-changing nature. It is simply not feasible to keep up with the pace of growth and change of electronic forms through manual categorization without expending immense time and effort (Pierre, 2000).

For instance, newspaper and news articles are being classified into the existing classes through the use of domain experts that consume the time of such experts. At legal publishing houses such as the West Group, human indexers read legal documents and index them manually, which is a bottleneck in publishing legal documents. Web browser users keep bookmarks to remember sites

they are interested in. Usually they categorize these sites according to their interests such as business, sports, travel, books, and movies. Therefore, it will be a great help to such problems if the automatic text categorization could classify all the documents based on the existing categories (Han et al., 1999).

2.4.1 Single-Label versus Multi-label Text Categorization

Different constraints may be enforced on the TC task, depending on the application. For instance we might need that, for a given integer *k*, exactly *k* (or $\leq k$, or $\geq k$) elements of *C* be assigned to each $d_j \in D$. The case in which exactly one category must be assigned to each $d_j \in D$ is often called the *single-label* (also known as *nonoverlapping categories)* case, while the case in which any number of categories from 0 to |*C*| may be assigned to the same $d_j \in D$ is dubbed the *multilabel* (also known as *overlapping categories*) case. A special case of single-label TC is *binary* TC, in which each $d_j \in D$ must be assigned either to category c_i or to its complement $\bar{c}_i$ (Sebastiani, 2002). This research employs single-label categorization as the selected data for the experiment allows such classification tasks.

2.4.2 Category-Pivoted Versus Document-Pivoted Text Categorization

There are two different ways of using a text classifier. Given $d_j \in D$, we might want to find all the $c_i \in C$ under which it should be filed (*document-pivoted categorization* —DPC); alternatively, given $c_i \in C$, we might want to find all the $d_j \in D$ that should be filed under it (*category-pivoted categorization*—CPC). This distinction is more pragmatic than conceptual, but

is important since the sets *C* and *D* might not be available in their entirety right from the start. It is also relevant to the choice of the classifier-building method, as some of these methods allow the construction of classifiers with a definite slant toward one or the other style. DPC is thus suitable when documents become available at different moments in time, e.g., in filtering e-mail. CPC is instead suitable when (i) a new category $c|C|_{+1}$ may be added to an existing set $C = \{ c_1, \ldots, c|C| \}$ after a number of documents have already been classified under *C*, *and* (ii) these documents need to be reconsidered for classification under $c|C|_{+1}$. DPC is more commonly used than CPC (Sebastiani, 2002; Coster, 2002) and this is the method used in this research since the selected data for the experiment needs to be classified into the existing fixed number of preclassified categories.

2.4.3 "Hard" Categorization versus Ranking Categorization

"Hard" categorization is when the algorithm decides a value for each document-category pair $(d_j, c_i) \in$ D x C. A complete automation of the TC task requires a decision for each pair $\{d_j, c_i\}$. Ranking is when the algorithm ranks all categories in C according to how well the document fit into each category, a partial automation (Coster, 2002).

Given $d_j \in D$ a system might simply *rank* the categories in $C = \{ c_1, \ldots, c|C| \}$ according to their estimated appropriateness to d_j, without taking any "hard" decision on any of them. Such a ranked list would be of great help to a human expert in charge of taking the final categorization decision, since it could thus restrict the choice to the category (or categories) at the top of the list, rather than having to examine the entire set. Alternatively, given $c_i \in C$ a system might simply

rank the documents in *D* according to their estimated appropriateness to c_i; symmetrically, for classification under c_i a human expert would just examine the top-ranked documents instead of the entire document set. These two modalities are sometimes called *category-ranking TC* and *document ranking TC*, respectively, and are the obvious counterparts of DPC and CPC (Sebastiani, 2002; quoted Yang, 1999).

Semiautomated, "interactive" classification systems are useful especially in critical applications in which the effectiveness of a fully automated system may be expected to be significantly lower than that of a human expert. This may be the case when the quality of the training data is low, or when the training documents of fully automated classifier cannot be trusted to be a representative sample of the unseen documents that are to come, which deteriorates the results and hence cannot be trusted (Sebastiani, 2002).

2.5. Applications of Text Categorization

As explained in chapter one TC has been used for a number of different applications: automatic indexing for Boolean information retrieval systems, document organization, text filtering, hierarchical categorization of web pages and word sense disambiguation - WSD (other examples of a variant of WSD are context-sensitive spelling correction, prepositional phrase attachment, part of speech tagging, and word choice selection in machine translation) (Sebstiani, 2002) and data analysis/text data mining (Lewis, n.d).

Referring Myers et al. (2000), Schapire and Singer (2000), Sable and Hatzivassiloglou (2000), Cavnar and Trenkle (199O) Larkey (1998), (Kessler et al. 1997), and (Forsyth 1999); Sebastiani (2002) stated that other applications areas of TC are speech categorization by means of a

combination of speech recognition and TC, multimedia document categorization through the analysis of textual captions, author identification for literary texts of unknown or disputed authorship, language identification for texts of unknown language, automated identification of text genre and automated essay grading. It is also useful in recommender systems (e.g. book) applications (Mooney & Roy, 1999). TC applications for document organization, text filtering and hierarchical categorization of web pages are discussed below.

2.5.1. Document Organization

Indexing with a controlled vocabulary is an instance of the general problem of document base organization. In general, many other issues pertaining to document organization and filing, be it for purposes of personal organization or structuring of a corporate document base, may be addressed by TC techniques. For instance, at the offices of a newspaper incoming "classified" ads must be, prior to publication, categorized under categories such as Personals, politics, economy, etc. Newspapers dealing with a high volume of classified ads would benefit from an automatic system that chooses the most suitable category for a given ad. Other possible applications are the organization of patents into categories for making their search easier, the automatic filing of newspaper or news stories under the appropriate sections (e.g., Politics, Home News, Lifestyles, etc.), or the automatic grouping of conference papers into sessions or case summaries may be put based on a sort of case classification (Zhang & Oles, 2000; Sebastiani, 2002). Topic spotting for newswire stories is one of the most commonly investigated applications domains of TC (Yang and Liu, 1999).

2.5.2. Text Filtering

Text filtering is the activity of classifying a stream of incoming documents dispatched in an asynchronous way by an information producer to an information consumer (Sebstiani, 2002; cited Belkin and Croft, 1992). A typical case is a newsfeed, where the producer is a news agency and the consumer is a newspaper (Sebastiani, 2002; cited Hayes et al., 1990). In this case, the filtering system should block the delivery of the documents the consumer is likely not interested in (e.g., all news not concerning sports, in the case of a sports newspaper). Filtering can be seen as a case of single-label TC, that is, the classification of incoming documents into two disjoint categories, the relevant and the irrelevant. Additionally, a filtering system may also further classify the documents deemed relevant to the consumer into thematic categories; in the example above, all articles about sports should be further classified according to which sport they deal with, so as to allow journalists specialized in individual sports to access only documents of prospective interest for them.

Similarly, an e-mail filter might be trained to discard "junk" mail and further classify nonjunk mail into topical categories of interest to the user. A filtering system may be installed at the producer end, in which case it must route the documents to the interested consumers only, or at the consumer end, in which case it must block the delivery of documents deemed uninteresting to the consumer. In the former case, the system builds and updates a profile for each consumer, while in the latter case, which is the more common, a single profile is needed. A profile may be initially specified by the user, thereby resembling a standing IR query, and is updated by the system by using feedback information provided (either implicitly or explicitly) by the user on the

relevance or nonrelevance of the delivered messages (Androutsopoulos et al., 2000; Drucker et al., 1999; Liddy et al., 1994; cited by Sebstiani, 2002).

The explosion in the availability of digital information has boosted the importance of such systems, which are nowadays being used in contexts such as the creation of personalized Web newspapers, junk e-mail blocking, and Usenet news selection.

2.5.3 Hierarchical Categorization of Web Pages

Text categorization is crucial to find interesting information on the World Wide Web, and to guide a user's search through hypertext (Joachims, n.d). Hierarchical categorization of web pages is decomposing the classification problem into a number of smaller classification problems, each corresponding to a branching decision at an internal node. TC has recently aroused a lot of interest also for its possible application to automatically classifying Web pages, or sites, under the hierarchical catalogues hosted by popular Internet portals. When Web documents are catalogued in this way, rather than issuing a query to a general purpose Web search engine a searcher may find it easier to first navigate in the hierarchy of categories and then restrict the search to a particular category of interest. Classifying Web pages automatically has obvious advantages, since the manual categorization of a large enough subset of the Web is infeasible. Unlike the previous applications, it is typically the case that each category must be populated by a set of $k_1 \leq x \geq k_2$ documents. CPC should be chosen so as to allow new categories to be added and obsolete ones to be deleted (Sebastiani, 2002).

CHAPTER THREE

MACHINE LEARNING APPROACH TO TEXT CATEGORIZATION

3.1 Introduction

In this chapter the basic concepts of machine learning, the need for machine learning, text representation and feature selection in machine learning are discussed. Moreover, for the purpose of this research, two machine learning classifiers/algorithms, naïve Bayes and kNN are also reviewed.

3.2 Basic concepts

As discussed in chapter two, the dominant approach to text categorization problem is based on machine learning techniques. Machine learning develops computational methods that would implement various forms of learning, in particular mechanisms capable of inducing knowledge from examples or data (Kubat el al., 1996).

It is important to define both the concepts learning and machine learning. “Learning is constructing or modifying representations of what is being experienced.” (Ryszard Michalski, n.d.; as cited by Dyer, n.d). Things learn when they change their behavior in a way that makes them perform better in the future. Learning implies thinking and purpose (Witten & Frank, 2000). Machine Learning is the field of scientific study that concentrates on induction algorithms and on other algorithms that can be said to "learn." (Kohavi & Provost, 1998). Russel & Norvig (1995) defined machine learning as a subfield of artificial intelligence concerned with programs that learn from experience**.** Likewise Mitchell (1997) defined machine learning broadly as “any computer program that improves its performance at some task through experience.”

More specifically, a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. This means that machine learning considers a few learning tasks. For instance, a computer program that learns to play checkers might improve its performance as measured by its ability to win at the class of tasks involving playing checkers games, through experience obtained by playing games against itself. In general, to have a well-defined learning problem, we must identify these three features: the class of tasks, the measure of performance to be improved, and the source of experience (Mitchell, 1997).

For instance, a hand writing recognition learning problem can be expressed in terms of such tasks:

- Task T: recognizing and classifying handwritten words within images
- Performance measure P: percent of words correctly classified
- Training experience E: a database of handwritten words with given classifications

As software has become one of the main bottlenecks of today's computer technology, the idea of introducing knowledge into computers by examples seems particularly attractive and appealing to common sense. Such a form of knowledge induction is especially desirable in problems that lack algorithmic solutions, which are ill-defined, or only informally stated. Medical diagnosis, visual concept recognition, engineering design, material behavior, chess playing, and detection of large regularities in large data sets are examples of such problems (Kubat et al., 1996).

A successful understanding of how to make computers learn would open up many new uses of computers and new levels of competences and customization. A detailed understanding of

information processing algorithms for machine learning might lead to a better understanding of human learning abilities (and disabilities) as well (Mitchell, 1997).

According to Mitchell (1997),

> *"We do not yet know how to make computers learn nearly as well as people learn. Yet, algorithms have been invented that are effective for certain types of learning tasks, and a theoretical understanding of learning is beginning to emerge. Many practical computer programs have been developed to exhibit useful types of learning, and significant commercial applications have begun to appear. For problems such as speech recognition, algorithms based on machine learning outperform all other approaches that have been attempted to date. In the field known as data mining, machine learning algorithms are being used routinely to discover valuable knowledge from large commercial databases containing equipment maintenance records, loan applications, financial transactions, medical records, and the like."*

Recently, machine learning provides achievements in recognizing spoken words, predict recovery rates of pneumonia patients, detect fraudulent use of credit cards, drive autonomous vehicles on public highways, and play games at levels approaching the performance of human world champions (Mitchell, 1997).

The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. Machine learning theory attempts to answer questions such as "How does learning performance vary with the number of training examples presented?"; "Which learning algorithms are most appropriate for various types of

learning tasks?" and "How can the learner automatically alter its representation to improve its ability to represent and learn the target function? " (Ibid).

There have been important advances in the theory and algorithms that form the foundations of this field. Machine learning draws on concepts and results from many fields, including statistics, artificial intelligence, philosophy, information theory, biology, cognitive science, computational complexity, and control theory (Mitchell, 1997).

One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner. Many of the algorithms search a hypothesis space defined by some underlying representation (e.g., linear functions, logical descriptions, decision trees, artificial neural networks)

3.3 Benefits of Machine Learning?

According to Dyer (n.d) the benefits of machine learning can be summarized as:

- To understand and improve efficiency of human learning. For example, use to improve methods for teaching and tutoring people, as done in CAI – Computer-aided instruction.
- To discover new things or structure that is unknown to humans. Example: Data mining
- To fill in skeletal or incomplete specifications about a domain. Large, complex Artificial Intelligence system cannot be completely derived by hand and require dynamic updating to incorporate new information. Learning new characteristics expands the domain or expertise and lessens the "brittleness" of the system.

3.4 Induction learning

Induction is learning a function from examples of its inputs and outputs (Russel & Norvig, 1995), which uses specific examples to general conclusions (Dyer, n.d). Supervised learning wants to learn an unknown function f(x) = y, where x is an input example and y is the desired output, i.e., it implies that we are given a set of (x,y) pairs by a "teacher." Such kind of learning is called supervised concept learning by induction. Given a training set of positive and negative examples of a concept, construct a description that will accurately classify whether future examples are positive or negative. That is, learn some good estimate of function *f* given a training set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where each y_i is either + (positive)or – (negative) (Dyer, n.d).

"The Inductive learning hypothesis is any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples." (Mitchell, 1997). In other words, given set of examples, we can make accurate predictions about future examples (Dyer, n.d). Although the learning task is to determine a hypothesis *h* identical to the target concept *c* over the entire set of instances X, the only information available about *c* is its value over the training examples. Therefore, inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data. The fundamental assumption of inductive learning is that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data (Mitchell, 1997).

Inductive learning methods require a certain number of training examples to achieve a given level of generalization accuracy. Decision trees, k-nearest neighbors, neural networks and Bayesian

learning algorithms are examples of inductive learning algorithms that require large data (Mitchell, 1997).

3.5. Feature Selection and Representation

Integral parts of all algorithms/approaches to text categorization are tokenization, feature selection, and creating vector representation of documents. The first step, tokenization, is common to most methods of text categorization. Tokenization is the process of dividing the input into distinct tokens – words and punctuation marks (Zhang & Oles, 2000).

After tokenization, each document is represented by a list of word occurrences. i.e., transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. A document may have various components (title, body, sections, etc.) which are, for the most part, pieces of text. However, we can concentrate on the simplest case, where the document is a single piece of text (Lewis, n.d; Joachims, n.d). In order to allow content-based classification of documents we need to obtain a representation of their content (Schweighofer, 2001)**.**

Each distinct word w_j corresponds to a feature, with the number of times word w_j occurs in the document as its value. The preprocessing of the document produces a bag (multiset) of index terms which represent a text sample as an unordered set of all words appearing in the text. This representation is sometimes called the bag-of-words (Lewis, n.d, Bennett, 1998, Sebastiani, 2002) or set of words (Sebastiani, 2002).

A common method for inductive text classification is to use partial or limited features of documents to uncover the entire characteristics of documents, e.g. measure the associations between documents and categories being via the limited words, particularly in the "bag of words" model (Bi, et al., n.d).

It has been found that representations more sophisticated than this bag-of-words representation do not yield better effectiveness. For instance, some authors have used phrases, rather than the individual words, as indexing terms, but the experimental results have not been uniformly encouraging. The reason for the discouraging result is that, although indexing languages based on phrases have superior semantic qualities, they have inferior statistical qualities with respect to word-only indexing languages (Sebastiani, 2002).

Text categorization is essentially a classification problem. The words occurring in the document sets are considered as variables or features for the classification problem. A relatively moderate size of document sets could easily have a vocabulary of tens of thousands of distinct words. Theoretically, having more features should give us more discriminating power. However, the real-world provides us with many reasons why this is not generally the case. Many existing algorithms simply would not work with these many numbers of attributes (Han et al., 1999).

In inductive methods for text document classification, feature selection (reduction) has received considerable attention for improving the effectiveness and efficiency of performance of learning algorithms. The argument in favor of feature reduction is that the learning model can operate with the relatively small size of input features, and non-informative features can be removed. Thus, reduced dimensionality of features makes the learning algorithm to have relatively little

computational cost, and the resulting features are interpretable. We can also increase the accuracy of the resulting model. In other words, in domain with a large number of features, the distribution is very complex and of high dimension which makes it very difficult to obtain good estimates of the many probabilistic parameters. In light of this, a number of researchers have recently addressed the issue of feature subset selection in machine learning. Experiments on feature reduction show that the optimum number of features for document classification varies (Bi, et al., n.d; Koller and Sahami, 1996).

Some empirical evidence shows the amount of features can be eliminated up to 90% or more of the unique terms with either an improvement or no loss in classification (as measured by average precision). The removal of common words such as articles, prepositions, conjunctions, etc. is an important step in document or text representation. Words will be features only if they occur in the training data at least 3 times and if they are not "stop-words" (like "and", "or", etc.). This representation scheme may even lead to very high-dimensional feature spaces containing large dimensions. Many have noted that the need for feature selection is to make the use of conventional learning methods possible, to improve generalization accuracy, and to avoid "overfitting". Furthermore, from IR it is known that scaling the dimensions of the feature vector with their term frequency (tf) and inverse document frequency (IDF) improves performance (Joachims, n.d; Baeza_Yates & Ribeiro_Neto, 1999; Sebastiani, 2002).

Most of the times, the standard *tfidf* function is used, defined as

$$tfidf(t_k, d_j) = \#(t_k, d_j) . \log |Tr| / \#Tr(t_k), \qquad (1)$$

Where $\#(t_k, d_j)$ denotes the number of times t_k occurs in d_j, and $\#Tr(t_k)$ denotes the *document frequency* of term t_k, that is, the number of documents in *Tr* in which t_k occurs.

This function embodies the intuitions that (i) the more often a term occurs in a document, the more it is representative of its content, and (ii) the more documents a term occurs in, the less discriminating it is (Sebastiani, 2002).

Stemming (i.e., grouping words that share the same morphological root), is also one technique for feature reduction in IR. However, its suitability to TC is controversial. Stemming has sometimes been reported to hurt effectiveness, though the recent tendency is to adopt it, as it reduces both the dimensionality of the term space and the stochastic dependence between terms (Sebastiani, 2002). Many researchers in the machine learning to text categorization do not make use of stemming (e.g., McCallum & Nigam, 1998; Joachims, n.d; Lewis, n.d; etc).

The other more sophisticated information-theoretic term selection techniques play a role in feature selection (reduction). Such techniques are the chi-square, information gain, mutual information, odds ratio, and etc. The measures of such techniques are based on mathematical definitions. Furthermore, term clustering and Latent Semantic Indexing are dimensionality reduction techniques by term extraction (Sebastiani, 2002).[1]

Categorization of documents is challenging, as the number of discriminating words can be very large. In many document data sets, only smaller number of the total vocabulary may be useful in categorizing documents. However, if we become too aggressive in reducing the number of words, then we might lose critical information for categorization tasks. Normally, the number of words after feature selection could be still in thousands (Han et al., 1999; Bi et al., n.d).

[1] See Sebastiani (2002) for detail understanding of dimensionality reduction by term selection and term extraction.

According to Pierre (2000), for the purpose of automatic text classification text features should be relatively few in number, moderate in frequency of assignment, low in redundancy, low in noise, related in semantic scope to the classes to be assigned, and relatively unambiguous in meaning.

After feature selection, each document is represented by a vector of word occurrences for each category where each vector component corresponds to a word feature selected for the category in the previous step (Zhang & Oles, 2000). In the classic supervised learning task, we are given a training set of labeled fixed-length feature vectors, or instances, from which to induce a classification model. This model is then used to predict the class label for a set of previously unseen instances. Thus, the information about the class that is inherent in the features determines the accuracy of the model.

3.6 Evaluation of Machine Learning Text Classifiers

3.6.1 Selection of Text Classifiers

As mentioned in previous chapters there are many learning algorithms useful for text classification purposes. Such algorithms may be more appropriate for certain application domain and type or size of data than the other. In order to select or evaluate the best algorithm the following lists are helpful (Dyer, n.d; Witten & Frank, 2000; Han & Kamber, 2001):

- Predictive accuracy of classifier: This refers to the ability of the model/algorithm to correctly predict the class label of new or previously unseen data. This is the most common criterion.

- Speed of learner and classifier, which refers to the computation costs involved in building the classifier and classifying a new document respectively
- Robustness: the ability of the model to make correct predictions given noisy data or data with missing values
- Scalability, which refers to the ability to construct the model efficiently given large amounts of data.
- Interpretability: the level of understanding and insight that is provided by the model.

3.6.2 Training versus Test sets

The machine learning approach relies on the availability of an initial corpus of documents preclassified under categories. Therefore, prior to classifier construction the initial corpus is split in two training set and test set, not necessarily of equal size. Most researchers use 20% (McCalllum & Nigam, 1998), 30% (Koller & Sahami, 1997) or 33 % (Joachims, n.d) of data for test set and the remaining for training set respectively.

The training set is inductively built by observing the characteristics of the documents. In most research settings, once a classifier has been built it is desirable to evaluate its effectiveness. A test set is used for testing the effectiveness of the classifier. Each document from the test set is fed to the classifier, and the classifier decisions are compared with the expert decisions. The documents in test set cannot participate in any way in the inductive construction of the classifiers; otherwise, the experimental results obtained would likely be unrealistically good, and the evaluation is considered not scientific (Sebastiani, 2002).

3.6.3 Performance Measures

A measure of classification effectiveness is based on how often the classifier decisions values match the expert decisions. Classification effectiveness is usually measured in terms of the classical IR notions of precision and recall, adapted to the case of TC (Sebastiani, 2002). Recall (R) is the percentage of the documents for a given category that are classified correctly. Precision (P) is the percentage of the predicted documents for a given category that are classified correctly. These can be formalized as

$R = NCP/NC$

and $P = NCP/NP$ respectively, where NC is the number of testing documents for a given category c; NP is the number of documents that are predicted as category c by the classifier; and NCP is the number of documents that are classified correctly (He, et al., n.d).

Classification accuracy is also the other method of measure of performance represented by c/n where n is the total number of test instances and c is the number of test instances correctly classified by the system (Sebastiani, 2002). Accuracy (error rate) is the rate of correct (incorrect) predictions made by the model over a data set. The Average results of accuracy can be represented in confusion matrix form. A confusion matrix is a matrix showing the predicted and actual classifications. A confusion matrix is of size LxL, where L is the number of different label values (Kohavi & Provost, 1998).

3.7 Text Classifiers

As discussed in chapter one there are many algorithms/classifiers useful for practical text classification purposes. For the purpose of this research, however, naïve Bayes and kNN are discussed in the following sections.

3.7.1 Naïve Bayesian Classifier

Naïve Bayesian classifier is one of the Bayesian Learning methods. The basis for all Bayesian Learning Algorithms is the Bayesian theorem. The following discussion on Bayes theorem is based on Mitchell (1997).

Bayes theorem provides a direct method for calculating the best probabilities. More precisely, Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability which may reflect any background knowledge about the chance that *h* is a correct hypothesis before we have observed the training data, P(h); the probability of observing various data given the hypothesis, P(D/h); and the observed data itself given no knowledge about which hypothesis holds, P(D). If we have no prior knowledge, then we might simply assign the same prior probability to each candidate hypothesis, P(h).

The formula for Bayes Theorem is given:

$$P(h/D) = P(D/h)P(h) / P(D) \qquad (2)$$

P(h/D) increases with P(h) and with P(D/h). P(h/D) decreases as P(D) increases, because the more probable it is that D will be observed independent of h, the less evidence D provides in support of h.

In machine learning problems we are interested in the probability p(h/D) that h holds given the observed training data D, which is called the posterior probability of h, because it reflects our confidence that h holds after we have seen the training data D, in contrast to the prior probability p(h), which is independent of D.

In many learning scenarios, the learner considers some set of candidate hypotheses H and is interested in finding the most probable hypothesis h $\in$ H given the observed data D (or at lest one of the maximally probable if there are several). Any such maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis. It is possible to determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis, i.e., h_{MAP} is a MAP hypothesis provided

$$h_{MAP} \equiv \arg\max_{h \in H} P(h/D)$$

$$\equiv \arg\max_{h \in H} P(D/h)P(h)/P(D)$$

$$\equiv \arg\max_{h \in H} P(D/h)P(h) \qquad (3)$$

We dropped the term P(D) because it is a constant independent of h. In some cases, it is assumed that every hypothesis in H is equally probable a priori (P(h_i and h_j in H). Therefore, we need only to consider the term P(D/h) to find the most probable hypothesis. P(D/h) is often called the likelihood of the data D given h, and any hypothesis that maximizes P(D/h) is called a maximum likelihood (ML) hypothesis,

$$h_{ML} \equiv \arg\max_{h \in H} P(D/h) \qquad (4)$$

In machine learning problems the data D is considered as training examples of some target function H as the space of candidate target functions.

Since Bayes theorem provides a principled way to calculate the posterior probability of each hypothesis given the training data, we can use it as the basis for a straightforward learning

algorithm that calculates the probability for each possible hypothesis, and then outputs the most probable.

Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents, long a favorite punching bag of new classification techniques. It has had a longer history as a simple, yet powerful classification technique. It has recently emerged as a focus of research itself in machine learning. Naïve Bayes probabilistic classifiers are commonly studied in machine learning. Machine learning researchers tend to be aware of the large pattern recognition literature on naive Bayes, but may be less aware of an equally large information retrieval literature dating back almost forty years. In fact, naive Bayes methods, along with prototype formation methods, accounted for most application of supervised learning to information retrieval until quite recently (Lewis, n.d).

Naïve Bayesian classifiers have exhibited high accuracy and speed when applied to large databases. It provides a very simple yet surprisingly accurate technique and is efficient and effective inductive learning algorithms for machine learning and data mining. The computational efficiency of this classifier has made it the benefactor of a number of research efforts; it is time efficient for its time complexity is only linear of the training data (Witten & Frank, 2000; Mitchell, 1997; Zheng, 1998; Zhang & et al., n.d; Keswani, 2002). Given *e* training examples over *f* attributes, the time required to learn a naive Bayesian classifier is O(ef) , i.e. linear. No learning algorithm that examines all its training data can be faster (Zhang, n.d).

One of the reasons behind naïve Bayes is its space efficiency; since, after discretization, it builds up a frequency table in size of the product of the number of attributes, number of class values,

and the number of values per attribute. It does not need to store the training data in memory when it builds the frequency table, but just scans the data once from the disk (Zhang, n.d).

In addition, naïve Bayesian classifier learning is robust to noise and irrelevant attributes. In general with Bayesian learning formulations and in particular with naïve Bayes classifier, there is no problem with missing values at all. The calculation would simply omit the missing attribute – it is not included in the frequency counts, and the probability ratios are based on the number of values that actually occur rather than on the total number of instances (Witten & Frank, 2000).

Naïve Bayes uses all attributes and allow them to make contributions to the decision that is equally important and independent of one another, given the class. It is simply counting how many times each attribute-valued pair occurs with each value. To classify a new coming instance, we count the coming attribute values that how many times are each attribute occurring on the given dataset. The attribute values and the category values contribute equally to the calculation. Then, the one with a higher outcome will be selected as more probable result to classify. This simple and intuitive method is bases on Bayer's rule of conditional probability (Witten & Frank, 2000) discussed above.

As mentioned repeatedly, Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called *class conditional independence,* i.e., the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. It is made to simplify the computations involved and, in this sense, is considered "*naïve*." (Witten & Frank 2000; Yang & Liu, 1999; Mitchell, 1997).

The good performance of Naïve Bayes is surprising because it makes a realistic assumption that is almost always violated in real-world applications: given class values, all attributes are independent. The basic idea in naïve Bayes approach is to use the joint probabilities of words and categories to estimates the probabilities of categories given a document. This assumption makes the computation of the naïve classifiers far more efficient than the exponential complexity of non-naïve Bayes approaches and has the minimum error rate in comparison to other classifiers because it does not use word combinations as predictors. Bias in estimating probabilities often may not make a difference in practice – it is the order of the probabilities, not their exact values that determine the classifications (Sahami, n.d; Zheng, 1998; Yang & Liu, 1999; Mitchell, 1997; Witten & Frank, 2000).

Though this is a fairly strong assumption, it is often not applicable which may lead to poor predictive generalization or inaccuracies (Zheng, 1998). The other limitations of Naïve Bayes classifier is that it decreases in accuracy if the numbers of training dataset used are small (Keswani, 2002).

3.7.1.1 How Does Naïve Bayes Classifier work?[1]

Assume that a set of training examples of the target function is provided. Each data sample is represented by *n* dimensional vector $x=(x_1, x_2, x_3, \ldots, x_n)$ depicting *n* measurements on the sample from *n* attributes $A_1, A_2, A_3, \ldots, A_n$. Suppose that there are m classes $C_1, C_2, C_3, \ldots, C_m$. A new instance is presented described by the set of tuple of attribute values (having no class) as $x =< a_1, a_2, a_3, \ldots, a_n >$. In order to classify the new instance, the Bayesian approach assigns

[1] [1] Most of the concepts of this section and section 3.7.1.2 are adapted from Mitchell (1997).

the most probable target value V_{MAP}, given the attribute values $< a_1, a_2, a_3, \ldots, a_n$ that describes the instance. The V_{MAP} is given by

$$V_{MAP} = \underset{v \in V}{\operatorname{argmax}} P(v_j / < a_1, a_2, a_3, \ldots, a_n >.)$$

By Bayes theorem, this can be rewritten as:

$$V_{MAP} = \underset{v \in V}{\operatorname{argmax}} (P(<a_1, a_2, a_3, \ldots, a_n > / v_j) * P(v_j)) / P(<a_1, a_2, a_3, \ldots, a_n >)$$

The denominator is constant for all classes. Hence, we can drop it., i. e.,

$P(<a1\ a_1, a_2, a_3, \ldots, a_n > / v_j) * p(v_j)$ (5) needs to be maximized.

Now it is easy to estimate each of $P(v_j)$ by counting the frequency with which each target value v_j occurs in the training set. Class prior probabilities may be estimated by $P(v_j) = s_i/S$ where s is the number of training samples of class v_j, and S is the total number of training samples. If the class prior probabilities are not known, then it is assumed that the classes are equally likely, i.e., $P(v_1) = P(v_2) = P(v_3) = P_n$ and therefore we would only maximize $p(< a_1, a_2, a_3, \ldots, a_n > / v_j)$.

However, Calculating $P(<a_1, a_2, a_3, \ldots, a_n > / v_j))$ is expensive. To reduce computational cost of $P(<a_1, a_2, a_3, \ldots, a_n > / v_j)$, class conditional independence is made to its equivalent formula:

$$P(a_i / V_j), i = 1, 2, \ldots, n = \prod_i P(a_i / V_j) \quad (6)$$

Hence, Naïve Bayes classifier: $VNB = \underset{v \in V}{\operatorname{argmax}} P(V_j) * \prod_i P(a_i / V_j), i= 1,2,\ldots,n$ (7)

where VNB denotes the target value output by the naïve classifier

3.7.1.2 Learning to Classify Text

The naïve Bayes classifier is effective and practical to classify such texts as electronic news articles, newsgroup articles, e-mail texts etc. It considers an instance space X consisting of all possible text documents (i.e., all possible strings of words and punctuation of all possible lengths). The algorithm is given training examples of some unknown target function f(x), which can take on any value from some finite set V. The task is to learn from these training examples to predict the target value for subsequent text documents.

The two main design issues involved in applying the naïve Bayes classifier to such text classification problems are first to decide how to estimate the probabilities required by the naïve Bayes classifier. The approach to representing arbitrary text documents is to define an attribute for each word position in the document and define the value of that attribute to be a word found in that position. Obviously, the long text documents will require a larger number of attributes than short documents; however, this will not cause any trouble. Given this representation for text documents, we can now apply the naïve Bayes classifier. Here, the algorithm calculates the probabilities of each class based on formula (7).

To calculate VNB using (7), we require estimates for the probability terms P(v_j) and P(a_i= w_k / V_j), w_k indicates the $_k$th word in the language vocabulary. However, estimating the class conditional probabilities is more problematic because we must estimate one such probability term for each combination of text position, word, and target value: i.e., number of positions in the

current example multiplied by the possible target values and the result is again multiplied by the number of vocabulary in the language.

Therefore, we can make an additional reasonable assumption that reduces the number of estimated probabilities by assuming the attributes are independent and identically distributed, given the target classification; that is, $P(a_i = w_k / V_j) = P(a_m = w_k / V_j)$ for all *i*, *j*, *k*, *m*. Then, we estimate the entire set of probabilities $P(a_1 = w_k / V_j)$, $P(a_2 = w_k / V_j)$… by the single position-independent probability $P(w_k / V_j)$, which we will use regardless of the word position. We now require only the number of target values multiplied by the number of vocabulary in the language. This is still a large number, but manageable.

Up to now we have estimated probabilities by the fraction of times the event is observed to occur over the total number of opportunities, i.e., n_c / n where *n* is the total number of training example for which V_j is w_k and n_c is the number of these for which $a_i = w_k$.

While this observed fraction provides a good estimate of the probability in many cases, it provides poor estimates when n_c is very small and the most probable value for n_c is 0. First, n_c / c produces a biased underestimate of the probability. Second, when this probability estimates is zero, this probability term will dominate the Bayes classifier because the quantity calculated requires multiplying all the other probability terms by this zero value.

To avoid this difficulty it is important to adopt a Bayesian approach to estimating the probability, using the smoothing methods like Laplace estimator and m-estimate. Laplace estimator is a strategy that gives uniform priors by priming each estimate with a count of one instead of 0. The reason of such a smoothing is to avoid zero probabilities of a term event. It is a standard technique (Witten & Frank, 2000; Soe, n.d). The other smoothing method is the m-estimate defined as:

$(n_c + m_p)/ (n+m)$,

where *p* is our prior estimate of the probability we wish to determine, and *m* is a constant called equivalent sample size, which determines how heavily to weight *p* relative to the observed data. A typical method for choosing *p* in the absence of other information is to assume uniform priors, if an attribute has *k* possible values we set $p = 1/k$. If *m* is zero, the m-estimate is equivalent to the simple fraction n_c/n. If both *n* and *m* are nonzero, then the observed fraction n_c/n and prior *p* will be combined according to the weight *m*.

To complete the design of the learning algorithm, we must still choose a method for estimating the probability terms. We can adopt the m-estimate with uniform priors and with *m* equal to the size of the word vocabulary. Thus, the estimate for $P(w_k / v_j)$ will be

$(n_k + 1)/ (n+|\text{Vocabulary}|$

where *n* is the total number of word positions in all training examples whose target value is v_j, n_k is the number of times word w_k is found among these *n* word positions, and |vocabulary| is the total number of distinct words (and other tokens) found within the training data.

Hence, the final algorithm is summarized below:

LEARN_NAIVE_TEXT(Examples, V)

Examples are a set of text documents along with their target values. V is the set of all possible target values. This function learns the probability terms P(w_k / v_j), describing the probability that a randomly drawn word from a document in class v_j will be the word w_k. It also learns the class prior probabilities P(v_j).

1. Collect all words, punctuation, and other tokens that occur in Examples

- Vocabulary $\leftarrow$ the set of all distinct words and other tokens occurring in any text document from Examples

2. Calculate the required P(v_j) and P(w_k / v_j) probability terms

- For each target value v_j in V do
    - Docs $_j$ $\leftarrow$ the subset of documents from Examples for which the target value is v_j
    - P(v_j) $\leftarrow$ |dos $_j$ | / |Examples|
    - Text $_j$ $\leftarrow$ a single document created by concatenating all members of docs $_j$
    - N $\leftarrow$ total number of distinct word positions in Text $_j$
    - For each word w $_k$ in Vocabulary
        - N $_k$ $\leftarrow$ number of times word w $_k$ occurs in Text $_j$
        - P(w_k / v_j) $\leftarrow$ (n_k + 1)/ (n+|Vocabulary|

CLASSIFY_NAIVE_BYES_TEXT(Doc)

Return the estimated target value for the document Doc.a $_i$ denotes the word founding the ith position within Doc.

- Positions ← all word positions in Doc that contain tokens found in Vocabulary
- Return VNB, where

$$V_{NB} = \text{argmax } P(v_j) \prod_{i \in positions} (a_i / v_j)$$

3.7.1.3 Event Models of Naïve Bayes Assumption

According to McCallum and Nigam (1998) there are two recent different probabilistic approaches to text classification, Multi-variate Bernoulli model and multinomial model, both of which make the naïve Bayes assumption.

A multi-variate Bernoulli model works with no dependencies between words and binary word features. In other words, this model specifies that a document is represented by a vector of binary attributes indicating which words occur and do not occur in the document; i.e., the number of times a word occurs in a document is not captured. When calculating the probability of a document, one multiplies the probability of all the attribute values, including the probability of non-occurrence for words that do not occur in the document. Here we can understand that the document to be the 'event," and the absence or presence of words to be attributes of the event. This approach is more traditional in the field of Bayesian networks, and is appropriate for tasks that have a fixed number of attributes (McCallum and Nigam, 1998).

A multinomial model is a uni-gram language model with integer word counts, which specifies that a document is represented by the set of word occurrences from the document. Alike the multi-variate Bernoulli model, the order of the words is lost; however, the number of occurrences of each word in the document is captured. When calculating the probability of a document, one

multiplies the probability of the words that occur. This means that the individual word occurrences to be the "events" and the document to the collection of word events. This approach is more traditional in statistical language modeling for speech recognition (Ibid).

Although both models has been used for text classification tasks the multinomial model is applied in this research.

3.7.2 K Nearest Neighbor

K Nearest Neighbor (kNN) is the most basic instance-based learning.[1] KNN is a well-known traditional statistical approach which has been intensively studied in pattern recognition for over four decades. KNN algorithm is one of the most thoroughly analyzed algorithms in machine learning, due to its age and simplicity. It has been applied to text categorization since the early stages of the research (Han et al., 1999; He et al., n.d; Mitchel, 1997; Witten & Frank, 2000).

The KNN algorithm is quite simple. Given test document as an input, the system finds the k nearest neighbors among the training documents, and uses the categories of the k neighbors to weight the category candidates. The similarity score of each neighbor document to the test document is used as the weight of the categories of the neighbor document. If several of the k nearest neighbors share a category, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of candidate categories, and ranked list is

[1] Instance-based learning is also called exemplar-based methods, and they are sometimes called "lazy" learners because they delay processing until a new instance must be classified. A key advantage of the delayed learning is that instead of estimating the target function once for the entire instance space, these methods can estimate it locally and differently for each new instance to be classified.

obtained for the test document. By thresholding on these scores, binary category assignments are obtained (Yang, 1999; Han et al., 1999; He et al., n.d).

The classes of these neighbors are weighted using the similarity of each neighbor to the test document, where similarity is measured by Euclidean distance. The distance metric that measures the similarity between two normalized patterns **a** and **b** can be defined by

$$D(a,b) = \sqrt{\sum_i (a_i - b_i)^2} \qquad (8)$$

The class assignment to the test pattern is based on the class assignment of the closest *k* patterns. A simple method is to label the test pattern with the class that has the most instances among the *k* nearest neighbors. Specifically, the class index y(x) *assigned* to a test pattern **x** is given by

$$Y(x) = \arg\max_i \{n(d_j, c_{i)} \mid d_j \in kNN\}$$

where $n(d_j, c_i)$ is the number of training pattern d_j in the *k* nearest neighbor set that are associated with class c_i. The similarity measurement is the distance between the word-frequency of vectors.

The value for *k* is pre-selected as a threshold value. In practical applications, typically, k is in units or tens rather than in hundreds or thousands. Using relatively larger *k* may include some not so similar pixels and on the other hand, using very smaller *k* may exclude some potential candidate pixels. In both cases the classification accuracy will decrease. However, the optimal value of k depends on the size and nature of the data (Khan et al., n.d). Of course, the computing time goes up as k goes up, but the advantage is that higher values of k provide smoothing that reduces vulnerability to noise in the training data. Larkey and Croft (1996) used k= 20, while

Yang (1999) has found $30 \leq k \leq 45$ to yield the best effectiveness. Anyhow, various experiments have shown that increasing the value of k does not significantly degrade the performance (Sebastiani, 2002).

As stated by Khan et al. (n.d), the algorithm for kNN is:

1. For each instance in the test set (the set to be classified), locate the k closest members (the k nearest neighbors) of the training dataset. A Euclidean Distance measure is used to calculate how close each member of the training set is to the test set that is being examined.
2. Examine the k nearest neighbors - which classification (category) do most of them belong to? Assign this category to the test instance being examined.
3. Repeat this procedure for the remaining instances in the test set.

KNN algorithm is very simple, robust and highly effective for many practical problems. A number of different experiments have shown kNN to be quite effective (Sebastiani, 2002; Atramentov & Atramentov, n.d).

However, kNN has its own drawbacks. One of its drawbacks is the difficulty in deciding an optimal *k* value. Typically it has to be determined through conducting a series of experiments using different *k* values (He et al., n.d).

The second major draw back of kNN is that it's not efficient in classification time and in terms of memory used. This is due to the fact that nearly all computation takes place at classification time rather than when the training examples are first encountered. In the KNN classification method,

no classifier model is built in advance. KNN refers back to the raw training data in the classification of each new sample, i.e., it uses all the features while computing the similarity between a test document and training documents. Therefore, one can say that the entire training set is the classifier. In other words, the training stage consists of simply memorizing the training instances and classifying new instances can be expensive (Atramentov & Atramentov, n.d; Mitchell, 1997; Witten & Frank, 2000; Sebastiani, 2002; Khan et al., n.d).

As explained in section 3.5, in many text data sets, using all the features may affect performance. Therefore, relatively small number of features (or words) maybe useful in categorizing documents.

The third problem of kNN is that it performs badly with noisy data, because the class of a test instance is determined by its single nearest neighbor without any averaging to help eliminate noise. Moreover, it is also affected and dominated by large number of irrelevant attributes, because all attributes contribute equally to the distance formula (Witten & Frank, 2000).

3.8 Related Literature

According to Sebastiani (2002), an increasing number of evaluations of naïve Bayes, kNN and other algorithms/methods on Reuter's collection have been published by Koller & Sahami (1997), Lewis & Ringuette (1994), and McCallum & Nigam (1998) and etc. [1] Experiments on the 20 newsgroups collection which contains 20, 000 articles[2] are made by Joachims (1997), Baker and McCallum (1998), McCallum & Nigam (1998), McCallum et al. (1998), Nigam et al. (2000) and Schapire & Singer (2000), etc.

[1] The Reuters collection, consisting of a set of newswire stories classified under categories, accounts for most of the experimental work in TC so far.
[2] The documents are messages posted to Usenet newsgroups, and the categories are the newsgroups themselves.

Joachims (1996) did an experiment on the 20 electronic newsgroups. That is 1,000 articles were collected from each newsgroup, forming a data set of 20,000 documents. The naïve Bayes algorithm was then applied using two-thirds of these documents as training examples, and performance was measured over the remaining third. The accuracy achieved by the program was 89%. In the experiment, 100 most frequent words were removed (these include words such as "the" and "of"), and any word occurring fewer than three times was also removed. The resulting vocabulary contained approximately 35, 500 words (Mitchell, 1997).

McCallum and Nigam (1998) did a research comparing the event models for naïve Bayes classifier. They use Yahoo! Science hierarchy (containing 13, 589 Yahoo web pages), the industry Sector hierarchy (consists of 6, 440 a company web pages classified in a hierarchy of industry sectors), the 20 newsgroups, and the WebKB[1] data sets. In all data sets except for the WebKB data set, they remove words that occur only once after the removal of the stop-words from each data set, but they didn't use stemming. They found that the multinomial performs better at larger vocabulary sizes over the multi-variate Bernoulli model at any vocabulary size.

Many empirical comparisons between Naïve Bayes and modern decision tree algorithms such as C4.5, Neural network and instance-based classifiers/algorithms showed that Naïve Bayes predicts equally well and in some cases it outperforms these other methods on many datasets (Zhang & et al., n.d; Elkan, 1997; Domingos & Pazzani, 1997; Mitchell, 1997).

[1]The WebKB data set contains 4,199 web pages gathered form university computer science departments

K-nearest neighbor classification has shown to be very effective for a variety of problem domains and has been shown to produce better results when compared against other machine learning algorithms such us C4.5 and RIPPER (Han, et al., 1999; Jin & Hauptmann, n.d).

Yang & Liu (1999) conducted an experiment on topic spotting of newswire stores on the Reuters-21578 corpus which consisting of 7, 769 training documents and 3, 019 test set documents. They reported that KNN is one of the top-performing methods. Generally, their result indicated that SVM, kNN and LLSF (Linear Least Squares Fit) significantly outperform the neural network and Naïve Bayes when the number of positive training instances per category is small (less that ten), and that all the methods perform comparably when the categories are sufficiently common (over 300 instances).

He et al. (n.d) conducted an experiment on Chinese text categorization based on a re-constructed People's Daily corpus, and evaluated three machine learning methods, namely KNN algorithm, SVM, and Adaptive Resonance Associative Map (ARAM), in terms of their capabilities in mining categorization knowledge from high dimensional, sparse, and relatively noisy document features vectors. The Experiments reveal that all three produce satisfactory performance on the test corpus. In their experiment, specifically in the kNN case they used the plain Euclidean distance equation as the similarity measure. On each pattern set containing a varying number of documents, different values of k running from 1 to 29 are tested and the best results are reported. In their experiment, kNN exhibits a relatively satisfactory performance on small training set.

Han et al. (1999) also did an experiment of weighted kNN that provides better classification accuracies over other classifiers. Their experiment shows that weighted kNN perform well

against several classification algorithms, such as C4.5, RIPPER, Naïve-Bayesian, PEBLS and VSM (Vector Space Model). However, they reported that it had a high computational cost.

In general, the result and performances of the algorithms may be different depending on the application area, the size of the data set and the preprocessing techniques used during the experiment.

3.9 Summary

A machine learning approach to text categorization passes some steps as discussed in the previous sections of this chapter. Given a pre-classified documents, the words in a document are assembled into a dictionary and represented in a vector of terms. After feature reduction from the vocabulary the documents are trained by the learning algorithm. Then a new instance is to be classified based on the contents of the training data (Su, 2002). The following is a summary of ML approach to TC.

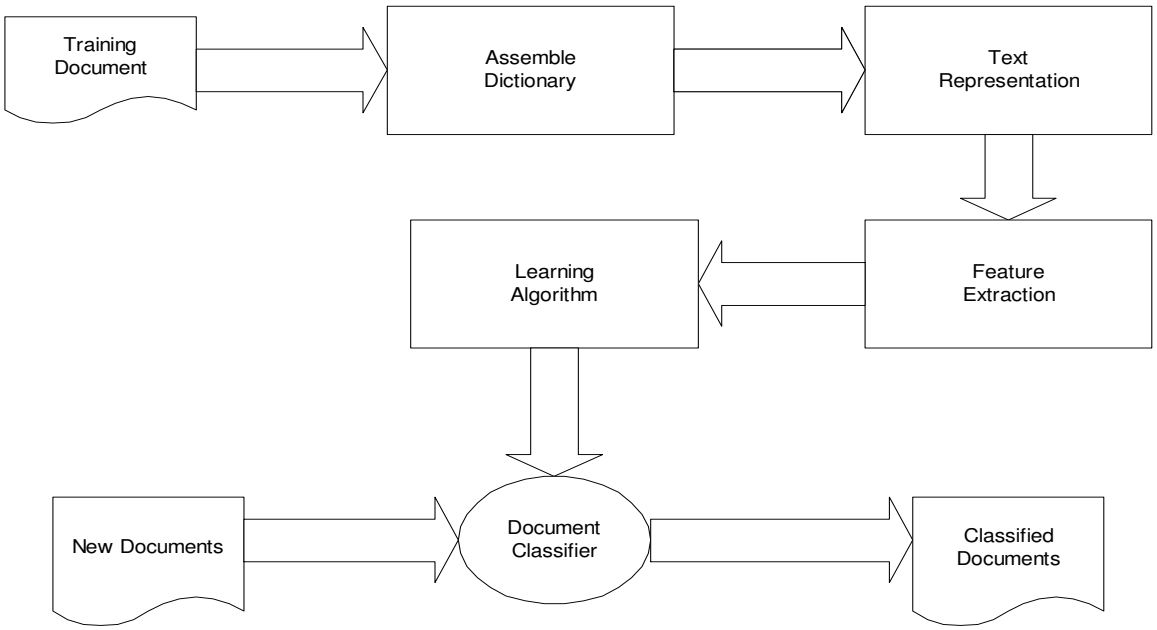

Fig 1. Text Classification Architecture

CHAPTER FOUR

AMHARIC WRITING SYSTEM

4.1. Introduction

As the automatic classification is implemented in Amharic news items, this chapter briefly discusses the origins of Amharic writing system; the Amharic alphabets, numerals and punctuation marks being used; Amharic writing problems and the fonts used in Amharic.

4.2. The Origins

The Ethiopic script had developed from the script of Ethiopia's classical language, Ge'ez, which was derived from the Sabaen script. The script used to write Ge'ez has been in use since the 4th century AD (Bender et al., 1976; Ethiopic script).

Amharic is the national language of Ethiopia and it is one of the Semitic languages. Together with a number of the lesser languages of Ethiopia, Amharic constitutes the Ethiopic branch of this family. All the Ethiopic languages (e.g. Amharic, Tigrinya) are descended from Ge'ez, the ancient literary and ecclesiastic language of Ethiopia (Introduction to Amharic Language, n.d).

Through adaptation and use over a long period of time, the script has undergone changes in shape and the number of symbols has changed, some being dropped and other added (Bender et al., 1976).

The Sabaean alphabet is said to have had twenty-nine symbols. Ge'ez took over only twenty-four of the twenty-nine Sabean symbols. The five which were rejected, represented sounds not used in

Ge'ez. In addition, two new symbols were invented to represent sounds of Greek and Latin loan words not found in Ge'ez. These are ጰ/p'/ and ፐ/p/. When Ge'ez was abandoned as the spoken language and other languages like Tigrinya and Amharic came into being, additional symbols were added to the script. The new symbols added in Amharic script are, ሸ(š), ዠ(ž), ቸ(c), ጀ(j), ኘ(n), ጨ(c′), ቨ(v), and ኸ(hɛ) (Bender et al., 1976).

4.3 The Alphabets

The Ethiopic writing system, which the Amharic language uses, consists of a core of thirty-three characters (ፊደል, fidel) each of which occurs in one basic form and in six other forms all known as orders. The seven orders (the first basic order and the other six orders) of the Ethiopic script represent the different sounds of a consonant-vowel combination (a characterization known as syllabic). The non-basic forms are derived from the basic forms by more-or-less regular modifications (Bender and et al., 1976; Hudson, 2001). A list of the Amharic alphabets (called fidel, ፊደል) with its orders is shown in appendix 1. The Amharic alphabet does not have capital and lower case distinctions.

4.4 Punctuation Marks

Words in Amharic language are separated by two dots (፡ ሁለት ነጥብ), though this is not exercised in type-written texts, blank spaces are generally used instead. The end of the sentence is marked by a square-formed four dots (። አራት ነጥብ), and the symbols ፣ (ነጠላ ሰረዝ) and ፤ (ድርብ ሰረዝ) represent a comma and semicolon respectively. Moreover, the language borrows some punctuation marks form foreign languages such as (?, ! , " , ", ', /, \, etc.). According to

Beletu (1982) there are about 17 punctuation marks used in Amharic language. However, the existing Amharic software do not make use some of them.

4.5. Numbers

Numbers in Amharic consist of single characters for one to ten, for multiples of ten (twenty to ninety), hundred, and thousand (see Appendix 2 for the list). According to Bender et al. (1976), these characters are derived from Greek letters, and some were modified to look like Amharic fidel. Each of the symbols has a horizontal stroke above and below. There is no symbol for zero in the Amharic script. Thus, arithmetical computations using the symbols are very difficult, if ever done. As a result, people generally use the Hindu-Arabic numerals. Ethiopic numbers are used mostly in writing dates and page numbers in text (Bender et al., 1976).

4.6 Problems in the Amharic Writing System

There are many problems regarding the writing system of Amharic language, as summarized below.

4.6.1 Consonants with Different Form

One of the problems in Amharic writing is the redundancy of symbols used with the same pronunciation. Although in the Ge'ez language, these different symbols give each word different meanings, in the Amharic language they have been used interchangeably. For instance, the use of all of the three different symbols **ሀ, ሐ,** and **ኀ** became unnecessary, since all these are now given the same pronunciation; h. Likewise, both **ሠ** and **ሰ** have the same pronunciation (/s/), both **ፀ** and **ጸ** (/s'/), and both **ዐ** and **አ** are pronounced as (/a/) (Bender et al., 1976; Getachew, 1967).

One may write **ሀይማኖት**, **ሃይማኖት**, **ሐይማኖት**, **ኀይማኖት** which means ‘religion’. The words **ንጉስ** and **ንጉሥ** (to mean ’king’) have the same sound and do not change the meaning of the word. Similarly, the words **ፀደይ** and **ጸደይ** (to mean ’spring’) have the same sound. Therefore, such different writing systems should be solved in the automatic text processing of the language as they may decrease the performance of the learner.

4.6.2 Formation of Compound Nouns

Bender and et al. (1976) stated that compound nouns are sometimes written as two separate words. For example, **ቤተ-መንግስት** which means “palace” may be written as **ቤተ መንግስት** or **ቤተመንግስት** and; **ቤተ-ሙከራ** as **ቤተ ሙከራ** which means “laboratory”. This happened to be inconsistent in Amharic texts and should be considered in automatic classification.

4.6.3 Transliteration Problems

Transliteration from foreign words to Amharic words is also another problem. The problems resulted from use of loan words that are borrowed from other languages and that do not possess their own translation in Amharic. The word “Oxford” may be transliterated as **ኦክስፈርድ** (oxferd) or **ኦክስፎርድ** (oxford) (Bender et al., 1976). Other examples of such loan words are transliterated differently as in the following:

Sponsor	**ስፖንሰር**	**እስፖንሰር**
Stadium	**ስታድዮም**	**እስታድየም**
Sport	**ስፖርት**	**እስፖርት**
Encyclopedia	**እንሳይክሎፕድያ**	**እንሳይክሎፔድያ**

4.6.4 Abbreviation Problems

In Amharic language, as in the English writing system, concepts can also be abbreviated or written in full spelling forms. This inconsistency creates problems for automatic classification. For instance, the phrase AD can be written in the text as ዓ.ም or ዓመተ ምህረት. However, in text classification tasks such words should come into one common form.

4.7 Amharic Fonts

Amharic alphabets do not have a representation in the ASCII (American Standard Code for Information Interchange) code table. As a result font developers have tried to develop their own keyboard driver programs that make use of the existing English Keyboard (ASCII codes) for writing Amharic. The English keyboard buttons are used in various combinations to produce Amharic characters (Zelalem, 2001).

Different Amharic fonts have been produced over the years (e.g. Alpas, Brana I, Brana II, Power Ge'ez, Geez, Agafari, Alxethiopian, Visual Ge'ez, etc.) but they all use the existing symbol sets differently. The Amharic text used for this research is written in the Visual Ge'ez font (VG2 Main). The symbol representation of the visual Ge'ez Amharic alphabets is attached in Appendix 3. In the Visual Ge'ez font, as in most Amharic fonts, some Amharic characters (those with diacritic markings) extend to more than one byte in their internal representation. The base character is one byte and the diacritic marking, another type. E.g. ቤ which is a composition of ቨ and ↄ, which is internally represented as the two symbols 'b@'.

CHAPTER FIVE

EXPERIMENT

5.1 Introduction

This chapter shows the results of the experiment. The preprocessing and the experiments are performed based on the concepts discussed in the previous chapters. More specifically, the efforts carried out to collect, prepare, and preprocess the data for the experiment are explained. Moreover, indexing/modeling, training and testing made are also reported.

5.2 Data Source

The data which is used for the experiment is collected from Ethiopian News Agency (ENA). The purpose of the Agency is to gather and distribute balanced and accurate news and news materials, concerning Ethiopia and the rest of the world in accordance with media policies, laws and directives (ENA, 2000).

Currently, ENA is utilizing the computer technology to provide the public and its clients with important and timely news. ENA developed a software called ENASoft, a bi-lingual-Amharic and English software used to create, edit, manage and archive stories. For the purpose of this research only Amharic news items were used.

5.3 Data Preparation

For ease of use and retrieval the Agency has its own classification system. ENA uses a manual classification system (the subject classification is performed by the journalists) and its classification system is divided into 16 main categories as listed in table 1 and each category also

contains a number of subclasses.[1] The data set of the Amharic news items was stored in MS-SQL Server database. The database contained 11, 238 records. For ease of processing, the database was imported to MS-Access and necessary fields for the study were queried from it. However, it was impossible to use the data contained in the databases as it is because it will create some problems for the experiment. Hence, manual data cleaning was done by the researcher. For instance, characters wrongly separated words (this happened at the time of data entry) were joined together because such problems can decrease the performance of the classifiers. Moreover, the researcher found that there were records which do not give meaning in Amharic language. This is because the data entry clerks or the journalist sometimes enter records for testing purposes, i.e., they simply type any character for checking purposes. Therefore, such records were deleted from the database (see appendix 4).

Then after only 11, 024 available Amharic news documents (only the headline, slug and keywords) were queried and copied from the database and made available in Rich text format for further processing. – i.e., Rainbow accepts the text data in Rich or plain text files. A list of category code given by the Agency, the category names, the folder name given by the researcher and the number of news items of each class is shown in table 1 below.

[1] This research considers only the main classification system of the Agency.

Table 1 Main Categories of Amharic Classification System

No.	Category Code	Category name and description	Folder name	No. of news items
1	ጋደአ	አደጋዎች (Accidents)	'adega'	554
2	ዓአጉ	ዓለም አቀፍ ጉዳዮች (International relations)	'alemakef'	1483
3	ባሕጉ	ባሕል ጉዳዮች (Cultural affairs)	'bahl'	22
4	ኢኮኖ	ኢኮኖሚ (Economy)	'economy'	491
5	ሕናፍ	ሕግና ፍትሕ (Law and justice)	'feteh'	917
6	ግብጉ	ግብርና ጉዳዮች (Agriculture)	'gebrena'	633
7	ሌፈዓ	ሌሎች የፈርጅ ዓይነቶች (Other classifications)	'leloch'	144
8	ማኅበ	ማኅበራዊ (Social)	'mahberawi'	1980
9	መናጸ	መከላከያና ጸጥታ (Defense and security)	'mekelakeya'	72
10	ብፓለ	ብሔራዊ ፖለቲካ (Politics)	'poletica'	497
11	ሣናተ	ሣይንስና ቴክኖሎጂ (Science & Technology)	'science'	157
12	ስፖር	ስፖርት (Sport)	'sport'	1510
13	ትምህ	ትምህርት (Education)	'temhert'	1993
14	ጤናጥ	ጤና ጥበቃቀ (Health)	'tena'	489
15	ማቆጥ	ጥቆማ (List of Events)	'Tkoma'	12
16	የአፀ	የአየር ጸባይ (Weather condition)	'yayertsebay'	73

The Agency has its own news items structure which contains four parts: a header, headline, lead paragraph, and body. The header incorporates classification code; slug (a general identification of subject in the form of a generic master slug), author's name, and dateline which gives date and place of story's origin and agency's acronym. The headline gives the content of a story in a few crisp words to catch the reader's interest. It doesn't exceed one line. The lead is the opening paragraph which captures the essence of a situation event clearly, and if possible, dramatically. Finally, the body elaborates on the lead and provides any necessary details (Zelalem, 2001).

Text representation can be analyzing the whole document or only part of the document. Using the whole available features may not lead to successful automatic text classification but rather using only a "good" subset of those (Jaochims, 1996). Zelalem (2001) also stated that rather than processing the whole parts of the news item, it would be easier to process the headline, lead, or a

combination of both so that the number of features will be reduced. In other words, the processing and training time complexity of the algorithms is reduced. Furthermore, the researcher found that there are few news items without having the body but every news item includes a headline, a keyword and slug. In most cases, the body of the news items contain details of the news such as the name of a place, person, or office; the time or date when the event took place; and numeric amount of an event or a thing and obviously the words in the headline. However, such words except the words in the headline may appear in almost all news items and they may not discriminate the classes. In this research; therefore, the headline, the slug and the keyword parts of the news structure are taken to perform the experiment assuming that the features in such parts represent each news items (see samples of such news records in appendix 5).

5.4 Text Preprocessing

In order to get good results from the experiment, language dependent text preprocessing should be performed before automatic classification is implemented. Text or document preprocessing is the step by which the text is made comfortable to the learning algorithm. The preprocessing step is simply a removal of non-informative words or characters from the text. In this research two phases of preprocessing were performed. The researcher removed non-informative characters such as punctuation marks, control characters (line feed, carriage return, tab) from the text before the data was given to the tool for processing. Since the tool has also facilities to remove non-informative words from the text, stop-words and words that occur fewer than or equal to 3 times in the collection were removed from the collection.

However, the problems of Amharic writing system mentioned in section 4.6.2 and 4.6.3 were not considered in the preprocessing task of this research due to time constraint. The problems

mentioned in section 4.6.4 is not significant to this research as such problems will in most cases appear in the body of the news article; i.e., abbreviations rarely appear in the headline, slug and keyword parts of a news article.

Preprocessing may also consider changing of cases (upper to lower or vice versa), removal of numbers, etc. In the Visual Ge'ez font (as in other Amharic fonts) upper case and lower case of the same alphabet represent two different symbols (orders) (Amharic fidel). For example, 'H' is the character used to represent ሀ, "h" (sixth order) whereas "h" is used to represent ሐ, "He" (first order). Therefore, no case conversion was done as part of text processing.

5.4.1. Removal of Extraneous Characters

The numbers, punctuation marks and control characters in the text of each file were not considered for classification as they do not have contribution to discriminate the classes. Words containing numbers like (2nd i.e. 2ኛ or ኢዜአ103862) were excluded at the first phases of preprocessing. Moreover, the standard control character; Amharic punctuation marks ። ፡ ፡፡ ፣ ፤; and symbols borrowed from English language (?, ! , " , ", ', /, \, etc.) were ignored. The following algorithm adapted from Zelalem (2001) was used to eliminate such extraneous characters from the news items.

1. Initialize the variable to hold the word
2. Read a character from the document
3. Check if the character is Amharic word delimiter
4. If not, concatenate the character to the variable

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

5.4.2. Changing Characters to their Common Form

In section 4.6.1 of the previous chapter, a discussion of the different symbols in the Amharic writing system with the same sound was made. These different symbols must be considered as equivalent because they do not cause changes in meaning. As a result, in this research, all different symbols of the same sound were converted to one common form. In order to exploit this equivalence Zelalem's (2001) algorithm was used. Thus, for example, if the character was one of ሐ ኀ ኃ ሐ ኸ or ሃ (all of them with a similar sound , h) then it was converted to ሀ. By the same token, all orders of ሠ (with the sound s) were changed to their equivalent respective orders of ሰ, all orders of ዐ (with the sound a) were changed to their equivalent respective orders of አ, all orders of ፀ (with the sound tse) were changed to their equivalent respective orders of ጸ. For those orders that use diacritic markings, the base characters are changed and the diacritic markings are attached.

The algorithm is as follows (for each of the seven orders);

1. read the character
2. if the character is any of

   ሐ ኀ ኃ ሐ ኸ ሃ or any other order thereof then

   change it to ሀ

   else if it is ሠ or any other order thereof

   change it to ሰ

   else if it is ፀ or any other order thereof

change it to ጸ

3. if the character that follows is a diacritic marking, attach it to the changed base character.

5.4.3 Creating Files and Folders

To make the news text comfortable for classification to the tool sixteen folders for each class were created and each news item was automatically created as a file (the NewsId field value is used as a file name for each news item) and grouped to its appropriate class.[1] News articles which are not assigned a class are ignored. The algorithm for this grouping is as follows:

1. Read the first word (which is the Newsid) from the file
2. Create a file with the name of the Newsid
3. Read the remaining characters of the line
4. Write the read characters in the created file except the Newsid
5. If the end of the line's word is among the classification code listed in Table 1

   Save the file in the appropriate folder (category)

   Else reject the file
6. Repeat until the end of the file.

5.4.4 Stop-word Removal

Stop-word removal is a necessity for text classification purposes. The assumption is that words which occur frequently in almost all documents are non-informative. Hence, 710 Amharic stop-words were identified. The stop words are of two kinds: those which are common to Amharic language text and Amharic news items.

[1] Rainbow accepts each instance as a file and the files should be grouped to a folder where a folder represents each class or category.

Like the English language, some words in Amharic are used very frequently in the normal usage of the language such as ነው (is), ሆኖም ግን (however), etc. Common words of this kind were identified. Moreover, it is usual that news is full of some common words that occur frequently in almost all news items. For instance, the words ተካሄደ to mean 'took place', ተጠየቀ to mean 'it was requested', etc., frequently occur in most Amharic news texts. Such words are verbs which usually found at the end of a sentence.[1] Hence, news specific common words of this type were used as a stopword list. Both types of common words were used by Zelalem (2001) for his experiment and adapted for this research.

Such stop words were saved as a file, and the file name was provided to the tool as the tool is capable of reading the file and removes the stopwords from each document during the indexing process.

5.5 Indexing/Modeling

The main requirement of the classification scheme is to provide sufficient background information on any topic. As mentioned in chapter one the tool used to automatically classify the news texts in this research is Rainbow. Rainbow is a command line program that performs statistical text classification.[2] Rainbow runs on UNIX or Linux. Because of its availability Red hat Linux was used for this experiment. Rainbow supports several different classification methods, (and the code makes it easy to add more). The default is Naïve Bayes, but kNN, TFIDF, SVM and probabilistic indexing are all available. Rainbow does not directly support

[1] Unlike the sentence structure of English (Subject-Verb-Object), the Amharic language structure is Subject-Object-Verb (SOV) where verbs usually come at the end of a sentence.
[2] It is based on the Bow library available at http:/www.cs.cmu.edu/~mccallum/bow.

classification tasks in which individual documents have multiple class levels. The general pattern of rainbow usage is a two steps process (McCallum, 1998; Ragone, n.d):

1. It reads the training data and writes/indexes to disk or archives a "model" containing their statistics. The test data may also be read as part of the model, or it can be left out and read later. The model simply stores various statistics on each document such as its corresponding category and the number of times words appear in a document in a matrix format.
2. Using the model, rainbow performs classification or diagnosis. Once indexing is performed and a model has been archived to disk, Rainbow can perform document classification. Statistics from a set of training documents will determine the parameters of the classifier; classification of a set of testing documents will be output. Rainbow does not look at the actual documents, but the tokenized model of the document.

Accordingly, the news items were indexed/modeled by ignoring the words in the stop-word list and words that occur fewer or equal to three times in the whole collection. After removal of such words, the vocabulary size of all categories reduced to 5,248 unique words. Consequently, the indexed documents were trained and tested afterwards as discussed in section 5.6 below.

5.6 Testing

Testing is important part of the experiment. Empirically evaluating the accuracy of hypothesizes is fundamental to machine learning. Estimating the accuracy of a hypothesis is relatively straightforward when data is plentiful. Therefore, to obtain an unbiased estimate of future accuracy, it is important to test the hypothesis on the test examples chosen independently of the

training examples and the hypothesis. When evaluating a learned hypothesis estimating the accuracy with which it will classify future instances is the critical one (Mitchell, 1997).

To the researcher's knowledge, there is no standard established text corpus for Amharic text classification testing. Hence, Amharic news items from ENA were selected by the researcher for the experiment. After the documents are indexed and modeled training and testing was performed.

In order to examine the applicability of a machine learning algorithm to Amharic news items naïve Bayes and kNN were compared on the same data and a set of categories. Furthermore, to see the effect, the researcher performed the experiment with different number of categories. That is three ('mahberawi, 'sport' and 'timhirt'), four ('alemakef', 'mahberawi', 'sport' and 'temhert'), seven ('adega','alemakef', 'feteh', 'gebrena', 'mahberawi, 'sport' and 'timhirt') categories were tested independently and finally the sixteen categories were tested together.

In all the experiments 33% of the news items were randomly taken as a test set and the rest as training set. Changing the numbers of the training/test split didn't bring a good result in this experiment. Hence, throughout the experiment only one training/test split is reported. Moreover, using the sophisticated feature reduction techniques such as information gain and odds ratio didn't bring good results in this experiment and are not reported. [1]

The testing results for naïve Bayes and kNN are discussed in the following sections.

[1] McCallum & Nigam (1998) found that the best multinomial classification accuracy of the 20 newsgroups data set was achieved using the entire vocabulary. In contrast, they found that the best multinomial performance on the 'interest" category of the Reuters data set was achieved using about 50 words.

5.6.1 Naïve Bayes Test

As explained in section 3.7.1 naïve Bayes is one of the simple algorithms of machine learning. In this experiment, Laplace estimator was used as smoothing method. The test results for the naïve Bayes classifier is depicted in the following sections.

5.6.1.1 Experiment on Three categories

Three classes 'mahberawi', 'sport', and 'temhert' that contain relatively equal number of news items were selected; where 1,809 news items were considered as a test set and 3,674 as training set. The number of test and training set selected from each category are shown in table 2 below:

Table 2: Naïve Bayes experiment on three categories

Category	Training data	Test set	total
mahberawi	1326	654	1980
sport	1012	498	1510
temhert	1336	657	1993

The classification accuracy for this test can be shown using confusion matrix. A confusion matrix contains a row and column where the row is actual categories and column is predicted number of documents classified to the corresponding class. For instance, the confusion matrix details tested for table 2 are:[1]

classname	0	1	2	:total	
0 mahberawi	610	16	28	:654	93.27%
1 sport	8	470	20	:498	94.38%
2 temhert	4	.	653	:657	99.24%

[1] The dot symbol (.) in the confusion matrix indicates that no document from the category specified in the row is incorrectly classified to other category or categories.

The first row indicates that 610 documents are classified correctly as the category ‘mahberawi; 16 documents from this category are predicted to be classified to the category ‘sport’ incorrectly; and 28 documents of the category ‘mahberawi’ are also misclassified as the category ‘temhert’ wrongly. The second row indicates 8 documents from the category ‘sport’ are classified incorrectly to the category ‘mahberawi’; 470 documents classified correctly; and 20 documents classified incorrectly to the category ‘temhert’. Likewise, for the third row, category ‘temhert’, 4 documents classified incorrectly as a category ‘mahberawi’, no document is classified incorrectly to the category ‘sport’ and 653 documents classified correctly in the category. The total test set and percent accuracy for each row is also given there. Hence, correctly classified news items are 1,733 out of 1,809 test set and the average accuracy is 95.78 percent.

5.6.1.2 Experiment on Four categories

The second experiment was performed on four categories: ‘alemakef’, ‘mahberawi’, ‘sport’ and ‘temhert’. The number of test and training set selected from each category are shown in table 3 below:

Table 3: Naïve Bayes experiment on four categories

Category	Training data	Test set	total
alemakef	993	490	1483
mahberawi	1327	653	1980
Sport	1012	498	1510
temhert	1336	657	1993

Total of 2,298 documents were randomly selected for the test set and the remaining 4,668 documents were placed in train set.

The confusion details of this experiment are:

	classname	0	1	2	3	:total	
0	alemakef	465	7	5	13	:490	94.90%
1	mahberawi	27	585	17	24	:653	89.59%
2	sport	19	4	463	12	:498	92.97%
3	temhert	6	5	2	644	:657	98.02%

As shown from this confusion details 2,157 news items out of 2,298 are correctly classified and the average percent accuracy is 93.86.

5.6.1.3 Experiment on Seven Categories

The third experiment was performed on seven categories: , 'adega', 'alemakef', 'feteh', 'gebrena', 'mahberawi', 'sport', and 'temhert', and. The number of test and training set selected from each category are shown in table 4 below:

Table 4: Naïve Bayes experiment on seven categories

Class	Training data	Test set	total
Adega	387	167	554
Alemakef	1008	445	1483
Feteh	641	276	917
Gebrena	444	189	633
Mahberawi	1396	594	1980
Sport	1057	453	1510
Temhert	1396	597	1993

Here total of 2,721 documents are randomly selected for the test set and remaining 6,329 documents for the training set.

The confusion details of this experiment are:

	classname	0	1	2	3	4	5	6	:total	
0	adega	147	6	1	3	6	2	2	:167	88.02%
1	alemakef	3	414	1	4	6	4	13	:445	93.03%
2	feteh	1	23	201	4	13	8	26	:276	72.83%
3	gebrena	1	10	1	169	5	.	3	:189	89.42%
4	mahberawi	3	19	13	6	505	11	37	:594	85.02%
5	sport	.	12	2	.	3	427	9	:453	94.26%
6	temhert	.	5	2	2	1	3	584	:597	97.82%

As depicted from the confusion details 2,447 out of 2,721 news items classified correctly. Hence, the average accuracy is 89.93 percent.

5.6.1.4 Experiment on sixteen Categories

The fourth experiment was performed on the sixteen categories as a whole listed in table 1. The number of test and training set selected from each category are shown in table 4 below:

Table 5: Naïve Bayes experiment on sixteen categories

Class	Training data	Test set	Total
Adega	371	183	554
Alemakef	993	490	1483
bahl	14	8	22
economy	329	162	491
Feteh	614	303	917
Gebrena	424	209	633
leloch	96	48	144
Mahberawi	1326	654	1980
mekelakeya	48	24	72
poletica	334	163	497
science	106	51	157
Sport	1022	498	1510
Temhert	1336	657	1993
tena	328	161	489
Tkoma	9	3	12
yayertsebay	49	24	73

Total of 3,638 documents were randomly selected for the test set and remaining 7,389 documents for the training set.

The confusion details of this experiment are:

	classname	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	:total	
0	Tkoma	.	.	1	.	.	.	.	.	2	.	.	.	.	.	.	.	3	0.00%
1	adega	.	163	5	.	.	3	.	.	9	.	.	.	2	1	.	.	183	89.07%
2	alemakef	.	2	457	.	.	3	5	.	7	.	.	.	3	9	4	.	490	93.27%
3	bahl	.	1	2	.	1	1	.	.	3	.	.	.	.	.	.	.	8	0.00%
4	economy	.	.	11	.	60	2	11	.	65	.	.	.	7	5	1	.	162	37.04%
5	feteh	.	3	26	.	.	213	3	.	20	.	3	.	10	23	2	.	303	70.30%
6	gebrena	.	1	10	.	.	.	184	.	4	.	1	.	1	7	1	.	209	88.04%
7	leloch	.	2	13	.	.	3	1	.	7	.	.	.	8	13	1	.	48	0.00%
8	mahberawi	.	3	15	.	1	8	12	.	549	.	7	.	19	27	13	.	654	83.94%
9	mekelakeya	.	1	6	.	.	6	.	.	2	.	3	.	3	2	1	.	24	0.00%
10	poletica	.	.	23	.	.	5	2	.	53	.	63	.	5	12	.	.	163	38.65%
11	science	.	1	17	.	.	5	1	.	7	.	.	4	2	7	7	.	51	7.84%
12	sport	.	1	20	.	.	4	1	.	4	.	.	.	461	7	.	.	498	92.57%
13	temhert	.	.	9	.	.	.	1	.	4	.	.	.	1	641	1	.	657	97.56%
14	tena	.	1	15	.	.	4	2	.	54	.	.	.	11	15	59	.	161	36.65%
15	yayertsebay	.	.	7	.	.	2	7	.	2	.	.	.	4	1	.	1	24	4.17%

The correctly classified news items for this test are 2,855 out of 3638 which gives 78.48 percent average accuracy. As we can see from this confusion details, all categories which contain less than 144 new items have 0.00% accuracy.

5.6.2 KNN Test

It was explained in section 3.7.2 that kNN algorithm is also a simple algorithm which classifies documents based on the Euclidean distance. The experiments of kNN are tested on the Amharic news items which are already tested on naïve Bayes algorithm. For each experiment, the researcher tested the kNN by fixing the value of k to 5, 10, 20 and 30 randomly. For curiosity, in some experiments k=2 is used. To save space, the confusion matrix for each experiment is not shown in the next sections.

5.6.2.1 Experiment on Three Categories

The first kNN experiment was done using three categories listed in Table 2 by fixing the k value to 5. Correctly classified news items are 1, 621 out of 1, 809 which gives 89.61 percent accuracy.

However, when k = 10, 20, and 30 the average accuracy is 89.55%, 88.56%, and 86.01% respectively. We can see that the good performance of this experiment is when k is 5 (89.61%).

5.6.2.2 Experiment on Four Categories

The second kNN experiment was performed using the four classes listed in Table 3; but k=2 in this case. Hence, 1,808 out of 2,298 documents are classified correctly and the average accuracy is 78.68 percent.

When k is adjusted to 5 the average accuracy increased to 84.25%. However, when k = 10, 20, and 30 the average accuracy resulted in 83.25, 84.51, and 83.72 percent respectively. Here, kNN performs best when k is 20 (84.51%).

5.6.2.3 Experiment on Seven Classes

The next experimental testing was done on the seven categories listed in table 5 and while k=5, 10, 20 and 30 the average accuracy is 72.55, 73.94, 75.27, 74.60 percent respectively. As indicated from this experiment kNN performances well when k=20 (75.27%).

5.6.2.4 Experiment on Sixteen Categories

Finally, the experiment was done on the sixteen categories as a whole. The result obtained was 59.18%, 62.56%, 64.40%, and 64.18 % average accuracy when k is 5, 10, 20 and 30 respectively where the best performance is obtained when k is 20.

The experiments of kNN tested above showed that k=20 performs best in all except the first experiment.

5.6.3 Comparison

The test accuracy results for naïve Bayes and kNN can be summarized as shown in table 6 below.

Table 6. Comparison between Naïve Bayes and kNN (in percent).

No. of categories considered	Naïve Bayes	kNN
3	95.80	89.61
4	93.86	84.51
7	89.93	75.27
16	78.48	64.50

As indicated from table 6 the naïve Bayes and kNN performs well with 6.19% average difference when three categories were considered for the test, i.e., naïve Bayes (95.80%) performances better than kNN (89.61%). However, the accuracy of both algorithms decreases as the number of categories increases (kNN dramatically decreases compared to naïve Bayes).

5.7 Automatic classification

As mentioned in section 5.5 once the data is indexed/modeled Rainbow classifies a new instance based on the contents of the training model. The following is a sample of full text sport news item (called a new instance) which is not modeled into the system (or not trained); the text supplied to the system is represented as:

"DÊ805
DÊdê¿HÄR 23 qN 1995¼x!z@x¼ (
bDÊdê S¬Ä!yM T§NT btµ/@dW yB/@‰êE l!G XGR µ*S WDDR DÊdê =RÜq=RQÂ
çRÑD ÃlMNM GB tlÃ†ÝÝ

bxSR s›T §Y ytjmrW yh#lt$ b#DñC Åw¬ kXrFT bðT bÈM qzÜ²Â x_Ub! yGB Ñk‰ãC ÃL¬†bT s!çN kXRFT mLS bh#lt$M wgN GB ÆYgŸ XNµ*N ytšl Ñk‰ãC y¬ybT nbRÝÝ

bXlt$ Åw¬ çRÑD kXrFt bðT mrUUT ÃL¬ybT bmçn# br¾W tÅêÓCN l¥rUUT µ*S b¥zGyT b#Dn#N l¥rUUT yÈr s!çN y=RÜq=RQ x_qEãC dGä çRÑD KB KLL s!dRs# µ*S lçrÑD tk§k×C bmS-T x_Ub! yGB Ñk‰ x§drg#MÝÝ

kXrFT mLS h#lt$M bDñC ytšl bmNqúqS tmLµCN ÃSdst$ b!çNM =RÜq=RQ µdrUcW _qET xSdNU+ yGB Ñk‰ãC bqR µ*SÂ mrB úYgÂß# yXlt$ Åw¬ xBQaLÝÝ

h#lt$M b#DñC kDÊdê yw-# XNdmçn# bXlt$ Wd »Ä ygÆW yµ*s xFÜq¶ kmc&WM g!z@ q$_„ Ãns nbR

Åw¬WN ym„T Ød‰L Ä¾ \§Ñ bql knrÄèÒcW _„ ydß# s!çN lçRÑD 2 tÅêÓC l=RÜq=RQ xND tÅêC b_Í¬cW b!Å µRD xúytêLÝÝ

¼DÊdê QRNÅF¼23(3(1995¼¸X¼mX¼

The above text in Visual Ge'ez font is:

ኝድሬ805
ድሬደዋ፤ህዳር 23 **ቀን** 1995/**ኢዜአ**/
በድሬዳዋ ስታዲየም ትላንት በተካሔደው የብሔራዊ ሊግ እግር ኳስ ውድድር ድሬዳዋ ጨርቃጨርቅና ሆርሙድ ያለምንም ግብ ተለያዩ::

በአስር ሰዓት ላይ የተጀመረው የሁለቱ ቡድኖች ጫወታ ከእረፍት በፊት በጣም ቀዝቃዛና አጥጋቢ የግብ ሙከራዎች ያልታዩበት ሲሆን ከእርፍት መልስ በሁለቱም ወገን ግብ ባይገም እንኳን የተሻለ ሙከራዎች የታየበት ነበር::

በእለቱ ጫወታ ሆርሙድ ከእረፍተ በፊት መረጋጋት ያልታየበት በመሆኑ በረኛው ተጫዋቾችን ለማረጋጋት ኳስ በማዘግየት ቡድኑን ለማረጋጋት የጣረ ሲሆን የጨርቃቃጨርቅ አጥቂዎች ደግሞ ሆርሙድ ክብ ክልል ሲደርሱ ኳስ ለሆረሙድ ተከላከዮች በመስጠት አጥጋቢ የግብ ሙከራ አላደረጉም::

ከእረፍት መልስ ሁለቱም ቡድኖች የተሻለ በመንቀሳቀስ ተመልካችን ያስደሰቱ ቢሆንም ጨርቃቀጨርቅ ካደረጋቸው ጥቂት አስደንጋጭ የግብ ሙከራዎች በቀር ኳስና መረብ ሳይገናኙ የእለቱ ጫወታ አብቅቷል::

ሁለቱም ቡድኖች ከድሬዳዋ የወጡ እንደመሆኑ በእለቱ ወደ ሜዳ የገባው የኳስ አፍቃቀሪ ከመቼውም ጊዜ ቁጥሩ ያነሰ ነበር

ጫወታውን የመሩት ፌደራል ዳኛ ሠላሙ በቀለ ከነረዳቶቻቸው ጥሩ የዳኙ ሲሆን ለሆርሙድ 2 **ተጫዋቾች ለጨርቃቀጨርቅ አንድ ተጫዋች በጥፋታቸው ቢጫ ካርድ አሳይተዋል::**

/ድሬደዋ ቅርንጫፍ/23-3-1995/**ሚእ/መእ/**

This text was classified as a ‘sport’ category by the experts. This text was given to naïve Bayes and kNN classifiers for categorization purpose. It was observed that naïve Bayes predicts it correctly as a category ‘sport’ with the probability value of 0.9999999976. The probability value for each category is also indicated by the system at the right side of each category as indicated below:

sport 0.9999999976
economy 1.716767401e-09
leloch 6.353381949e-10
poletica 8.846635274e-11
feteh 6.002569133e-12
alemakef 3.566731631e-13
gebrena 2.225106941e-13
science 2.2723673e-14
mekelakeya 7.621667823e-17
mahberawi 1.317192802e-17
tena 4.640018701e-18
adega 6.060071463e-19
bahl 1.7362255e-25
yayertsebay 3.955577202e-26
temhert 2.072197735e-30
Tkoma 1.35422903e-30

As discussed in the literature kNN needs the k value. Rainbow provides k=30 by default and this was tested for the text written above; the distance measured by the system is:

temhert 161
sport 144
economy 109
mahberawi 88
alemakef 61
gebrena 18
tena 18
poletica 17
adega 0
bahl 0
feteh 0
leloch 0
mekelakeya 0
science 0
Tkoma 0

yayertsebay 0

However, kNN wrongly predicted the document to the category ‘temhert’ with highest Euclidean value. KNN also predicted it incorrectly when k is 5, 10 and 20. Whereas when k is readjusted to 45, kNN predicted the document correctly to the category ‘sport’ as indicated below:

sport 210
temhert 178
economy 109
mahberawi 104
alemakef 93
science 49
gebrena 34
poletica 33
tena 18
adega 16
feteh 16
bahl 0
leloch 0
mekelakeya 0
Tkoma 0
yayertsebay 0

5.8 Discussion

From the experiments it can be seen that when categories which have almost equivalent number of documents are trained or tested together resulted in better classification accuracy than when the categories with small number of documents are added. In other words, the classification accuracy decreases as the categories contain relatively very few documents. This happened for both naïve Bayes and kNN.

This experiment produces similar results with Rennie (1993). Rennie found that Naïve Bayes performs poorly when one class has relatively few examples. This is because the performance of naïve Bayes classifier can easily be dictated by the class with the smallest number of examples,

i.e., if one class has little training data, its variance may be much greater than other classes and that variance will dominate the variance of the overall classification outputs. In other words, insufficient training examples in one class can negatively affect overall performance. The benefit that Naïve Bayes receives from additional training data is marginal if the data is not distributed evenly across the categories. In general it is widely believed that additional training data improves classification; for every word, the variance contribution for that word diminishes with additional training data and at the same time the overall variance in the classification output also decreases (Rennie, 1993; Vaithyanathan et al, 2000).

From the experiment it is indicated that the performance of kNN dramatically diminishes compared to naïve Bayes. As explained in the literature this may be due to the fact that kNN provides relatively satisfactory performance on small training set than large dataset (He et al., n,d). One major problem of kNN is in choosing k value. In the experiment it is shown that k=20 performs good in most of the experiments.

Generally, we can see that as the data for each class is sparse (the number of observations is very low compared to the dimensionality of the data), the average percent accuracy is reduced. The results from the above experiment confirmed that it is better if the categories contain almost equal number of documents. Moreover, naïve Bayes tends to be the better classifier for Amharic News items.

CHAPTER SIX

CONCLUSTIONS AND RECOMMENDATION

6.1 Conclusions

This research has presented an automatic news items categorization for Amharic news articles using machine learning techniques: naïve Bayes and kNN.

Research studies which have been done in the area of machine learning in text categorization indicate good results. Accordingly, the objective of the research was to test the applicability of machine learning techniques to Amharic news text categorization; and this research showed promising results.

For the purpose of this research 11, 024 (originally the data were 11, 238) news articles written in Visual Ge'ez font were used. Only the headline, the slug and keywords were considered to build the models assuming that they contain features which represent the document and the processing and learning time of the algorithms also be reduced. The database which was stored in MS-SQL server was imported to Ms-Access database. Then after the Amharic data was processed (as discussed in section 5.3, only 214 records inconvenient to the research were excluded) and saved in plain text format. Text preprocessing is a pre-request for automatic text categorization. Therefore, during the research removal of extraneous characters, numeric characters and stop-words, changing different characters with the same pronunciation to one common character was done. In order to make the text comfortable to the tool used in this research each document was created as independent file and grouped to its appropriate category automatically.

Finally, the model was trained and tested. 33% of the data was considered as a test set and the remaining as a training set. 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 data (78.48% vs. 64.50%) respectively. This research indicated that naïve Bayes classifier is more applicable to Amharic news articles than the kNN classifier. Moreover, it is learnt that considering categories with equal number of news items increases the performance of the classifiers. For instance, the three categories contain relatively equal number of documents; however, the 16 categories contain unevenly distributed documents over the categories. From the experiment we observed that uneven distribution of data over the categories decreases the performance of the classifiers (K nearest Neighbor dramatically decreases than naïve Bayes). In other words, insufficient examples in one class can affect the classifier as a whole. The reason why kNN classifier dramatically decreases its performance may be due to the fact that it exhibits good performance on small data than a large data set.

It was also observed that the classification of Amharic news articles is possible without using the sophisticated feature reduction techniques such as information gain and odds ratio.

6.2 Recommendations

The result of this research indicated that machine learning techniques (naïve Bayes and kNN) are applicable for automatic Amharic news text classification. However, continuous researches have to be conducted to get better results. Hence, the following points are recommended.

- The availability of standard stop-word list would possibly facilitate researches in the areas of automatic classification. Nevertheless, there is no standard stop word list for use in the Amharic language. Therefore; a standard Amharic stop-word list should be developed.

- The availability of standardized text corpus promotes text classification researches. Nevertheless, there is no established text corpus for Amharic text classification purposes. Hence, the need to develop text corpus is recommended.
- The researcher tried to correct some spelling errors manually which is not exhaustive for the purpose of this research. Spelling errors may deter the text processing in general and text automatic classification tasks in particular. This shows the need to develop Amharic spell checker.
- This research showed promising if not best result. Incorporating the Amharic stemmer may also provide good performance and should be tried by other researchers.
- This research tested Naïve Bayes and kNN algorithms to automatically classify Amharic news articles. There are many other algorithms available for automatic text classification. Hence, different machine learning algorithms have to be tested for the task.
- This research considers the single-label classification which assigns a document only to one category. There may be a need one document may be assigned to more than one category (multi-label classification). Accordingly, research has to be done on multi-label classification for Amharic texts.
- This research was employed document-pivoted categorization where a document is automatically assigned to already existing predefined classes. Whereas category-pivoted categorization is considering documents for reclassification to newly added categories; therefore, a research may be conducted for such method of text categorization.
- Finally, this research didn't consider hierarchical categorization system, i.e., the main categories may have a number of subcategories (as in the case of ENA). Therefore, research of this kind should be done.

BIBLIOGRAPHY

Atramentov, Anna, and Oleksiy Atramentov. RRT-compression for knn Classification Algorithm. http://www.cs.iastate.edu/~anjuta/cs573/project.htm, n.d.

Baeza-Yates, Ricardo and Berthier Ribeiro-Neto. Modern Information Retrieval. Harlow, England: Addison Wesley, 1999.

Bandyopadhyay, Sivaji. "An Example Based MT System in News Items Domain from English to Indian Languages." In Machine Translation Review. No. 12, pp. 7-10, 2001. http:// www.bcs.org.uk/siggroup/nalatran/mtreview/mtr-12/7.htm

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

Bender, M. L et al. Language of Ethiopia. London: Oxford University Press, 1976.

Bennettt, Paul N. Text Categorization through Probabilistic Learning. Applications to Recommender Systems. http://citeseer.nj.nec.com/bennett98text.html, 1998.

Bethlehem Mengistu. N-Gram-based Automatic Indexing for Amharic Text. Masters Thesis. Addis Ababa University: Addis Ababa, 2002.

Bi, Y. et al. Text Passage Classification Using Supervised Learning. http://ir.dcs.gla.ac.uk/lumis99/papers/bi.pdf, n.d.

Chen, Hsinchum. Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning and Genetic Algorithms. http://ai.bpa.arizona.edu/papers/mlir93/mlir93.html, 2003.

Coster, Rickard. Introduction to Text Categorization. http://www.dsv.su.se/~rick, 2002.

Crimmins, Francis. Classification. http://dev.panopficearch.com/classification.html,

2001.

Domingos, P. and M. Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss." In Machine Learning, No.29, pp.103–130, 1997. http://citeseer.nj.nec.com/domingos97optimality.html

Dyer, C. R. Machine Learning. Lecture Notes in Computer Science. Madison: University of Wisconsin. http://www.cs.Wisc.edu/%7Edyer/cs540/notes/learning/html, n.d.

Elkan, Charles. Naive Bayesian Learning. Adapted from Technical Report No. CS97-557, University of California, San Diego. http://citeseer.nj.nec.com/30545.html, 1997.

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

Ethiopian News Agency. 2000.

Ethiopia: Central Statistical Authority (ECSA). The 1994 Population and Housing Census of Ethiopia: Results at Country Level. Vol. 1 Statistical Report 44. Addis Ababa, 1998.

Ethiopic script. http://www.omniglot.com/writing/ethiopic.htm

Getachew Haile. The Problems of Amharic Writiing System. Unpublished.

Han, Jiawei, and Mitcheline Kamber. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers, 2001.

Han, Eui-Hong et al. Text Categorization using Weight K-Nearest Neighbor classification. Minnesota: University of Minnesota. http://www.cs umn.edu/~han, 1999.

He, Ji et al. A Comparative Study on Chinese Text Categorization Methods. http://citeseer.nj.nec.com/he00comparative.html, n.d.

Hosting Works. http://hostingworks.com/support/dict.phtml?foldoc=brute+force, 2000.

Hudson, G. Aspects of the History of Ethiopic Writing. IES Bulletin, 25, pp. 1-10, 2000.

Introduction to Amharic Language.

http://216.239.37.100/search?q=cache:pZ9j8aKjjr8C:www.worldlanguage.com/Languages/Amharic.htm+introduction+to+amharic+language&hl=en&ie=UTF-8, n.d.

Jin, Rong, and Alexander G. Hauptmann. Automatic Title Generation for Spoken Broadcast News. Pittsburgh: Carnegie Mellon University. http://zero.inf.cs.cmu.edu/alex/HLT_2001.pdf, n.d.

Joachims, Thorsten. Text Categorizatin with Support Vector Machines: Learning with Many Relevant Features. Dortmund, Germany: Universitat Dortmund. http://citeseer.nj.nec.com/cache/papers/cs/902/http:zSzzwww-ai.cs.uni-dortmund.dezSzDOKUMENTEzSzJoachims_98a.pdf/joachims98text.pdf, n.d.

Keswani, Girish. NBC: Niave BayesClassifierAlgoirthms. http://modern.csee.edu/~keswani/thesis/node14.html, 2002.

Khan, Maleq, et al. k-Nearest Neighbor Classification on Spatial Data Streams Using P-Trees1, 2 Fargo: North Dakota State University. http://cs.hbg.psu.edu/~ding/publications/PAKDD02_KNN.pdf, n.d.

Kohavi, Ron, ed. and Foster Provost, ed. “Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process.” In Machine Learning, 30, pp. 271-274, 1998. http://robotics.stanford.edu/%7Eronnyk/glossary.html.

Koller, Daphne and, Mehran Sahami. “Hierarchically Classifying Documents Using Very Few Words.” In Proceedings of ICML-97, 14th International Conference on Machine Learning, pp.170-178, 1997. http://citeseer.nj.nec.com/cache/papers/cs/8574/http:zSzzSzwww-diglib.stanford.eduzSzdiglibzSzWPzSzPUBLICzSzDOC130.pdf/koller97hierarchically.pdf

Koller, Daphne and, Mehran Sahami. “Toward Optimal Feature Selection.” Proceedings of the 13th International Conference on Machine Learning (ML), pp.284-292, 1996. http://robotics.stanford.edu/~koller/papers/ml96.html

Kubat, et al. A Review of Machine Learning Methods. http://www.google.com/search?hl=en&lr=&ie=ISO-8859-1&q=kubat+a+review+of+machine+learning+methods, 1996.

Lewis, David D. Machine Learning for Text Classification. Lecture Notes. n.d.

Lewis, David D. Naïve (Bayes) at Forty: “The Independence Assumption in information Retrieval. In ECML, 1998.” http://www.research.all.com/~lewis.

McCallum, Andrew & Kamal Nigram. "A Comparison of Event Models for Naive Bayes Text Classification." In AAAI/ICML-98 Workshop on Learning for Text Categorization. pp. 41-48, 1998. http://citeseer.nj.nec.com/mccallum98comparison.html

McCallum, Andrew. Rainbow. http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/, 1998.

Mitchell, Tom M. Machine Learning. New York: The McGraw-Hill Companies, Inc., 1997.

Mooney, Raymond J. and Loriene Roy. “Content-Based Book Recommending Using Learning for Text Categorization.” In Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, CA, 1999. http://citeseer.nj.nec.com/cache/papers/cs/14198/ftp:zSzzSzftp.cs.utexas.eduzSzpubzSzmooneyzSzpaperszSzlibra-dl-00.pdf/mooney00contentbased.pdf

Neumann, Günter and Sven Schmeier. Combining Shallow Text Processing and Machine Learning in Real World Applications. http://www.dfki.de/~neumann/publications/new-ps/ijcai99-ws.pdf, n.d.

Nigam, Kamal Paul. Using Unlabeled Data to Improve Text Classification. PhD. Thesis. http://citeseer.nj.nec.com/cache/papers/cs/22803/http:zSzzSzwww.cs.cmu.eduzSz~knigamzSzpaperszSzthesis-nigam.pdf/nigam01using.pdf, 2001.

Pierre, John M. Practical Issues for Automated Categorization of Web Sites. http://www.ics.forth.gr/isl/SemWeb/Proceeedings/session33/html_version/semanticweb/html, 2000.

Ragone, Andrew. Machine Learning. http://plan.mcs.drexel.edu/courses/ml/homework/homework1.pdf, n.d.

Rasmussen, E. Clustering Algorithms. In W. B. Frakes and R. Baeza-Yates (eds.) Information Retrieval Data Structures and Algorithms. Prentice Hall: N. J., 1992.

Rennie, Jason D. M. Improving Multi-Class Text Classification with Naive Bayes. Masters Thesis. Massachusetts Institute of Technology, 2001. http://citeseer.nj.nec.com/460958.html

Review of K-Nearest Neighbor Text Categorization Method http://wwwcsif.cs.ucdavis.edu/~liaoy/research/text_ss02_html/node4.html, n.d.

Rijsbergen, V. Information Retrieval. London: Butterworths, 1979.

Russel, Stuart and Peter Norvig. Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall, 1995.

Sahami, Mehran. Learning Limited Dependence Bayesian Classifiers. Stanford : Stanford: StanfordUniversity. http://citeseer.nj.nec.com/cache/papers/cs/346/http:zSzzSzwalrus.stanford.eduzSzdiglibzSzpubzSzreportszSzkdd96-learn-bn.pdf/sahami96learning.pdf, n.d.

Salton, G. Introduction to Modern Information Retrieval. New York: McGraw-Hill, Inc., 1983.

Schweighofer, Erich. Automatic Text Representation, Classification and Labeling in European Law**.** http://www.ifs.tuwien.ac.at/ifs/research/pub_html/sch_icail01/ , 2001.

Sebastiani, Fabrizo "Machine Learning in Automated text classification." In ACM Computing Surveys. Vol.34 NO.1, pp1-47, 2002. http://faure.iei.pi.cnr.it/~fabrizio/

Seo, Young-Woo. Multinomial General Model. http://www2.cs.cmu.edu/afs/cs.cmu.edu/user/ywseo/www/textminerdocs/doc/textminer/text/MultinomialGenerativeModel.html, n.d.

Su, Xiaomeng. Text Categorization. Lecture Notes. http://www.idi.ntnu.no/emner/sif8047/presentations/sif8047_TxtCat.pdf, 2002.

Vaithyanathan, Shivakumar, Jianchang Mao and Byron Dom. "Hierarchical Bayes for Text Classification." In Npricai 2000 Workshop on Text and Web Mining, pp. 36-43, 2000. http://citeseer.nj.nec.com/57872.html.

Van Uden, Mark. Rocchio: Relevance Feedback in Learning Classification Algorithms. http://citeseer.nj.nec.com/cache/papers/cs/5681/http:zSzzSzwww.cs.kun.nlzSznsccszSzartikelenzSzmarkuden.pdf/rocchio-relevance-feedback-in.pdf, n.d.

Witten, Ian H. and Eibe Frank. Data mining: Practical Machine Learning Tools and Techniques with Java implementations. San Francisco: Morgan Kaufmann publishers, 2000.

Yang, Yiming. An Evaluation of Statistical Approaches to Text Categorization. Netherlands: kluwer Academic Publishers, 1999. http://citeseer.nj.nec.com/yang97evaluation.html

Yang, Yiming, and Xin Liu. A re-examination of Text Categorization Methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval. pp 42-49, 1999.

http://citeseer.nj.nec.com/cache/papers/cs/26885/http:zSzzSzranger.uta.eduzSz~alpzSixzszreadingszSzYangSigir99CategorizationBenchmark.pdf/yang99reexamination.pdf

Zelalem Sintayehu. Automatic classification of Amharic news items: the Case of Ethiopian News Agency. Master Thesis at SISA. Addis Ababa University. Addis Ababa. 2001.

Zhang et al. The Learnability of Naïve Bayes. Lecture Notes in Computer Science. http://citeseer.nj.nec.com/434974.html, n.d.

Zhang, Tong and Frank J. Oles. Text categorization based on regularized linear classification methods. New York, 2000. http://citeseer.nj.nec.com/cache/papers/cs/26805/http:zSzzSzwww.research.ibm.comzSzdssgrpzSzPaperszSzlincat_final.pdf/zhang00text.pdf.

Zheng, Zijian. "Naïve Bayesian Classifier Committees." In Proceedings of ECML'98, Berlin: Springer Verlag, 196-207, 1998.

APPENDICES:

Appendix 1. The Amharic character set (Bender *et al*., 1976).

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

ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ

Appendix 2: Amharic numbers

፲	፳	፴	፵	፶	፷	፸	፹	፺	፻
10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**

፩	፪	፫	፬	፭	፮	፯	፰	፱
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Appendix 3: List showing the symbols used in the Visual Ge'ez font for the Amharic fidel

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
h	h#	£	¦	ÿ	H	ç
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
l	l#	l!	§	l@	L	lÖ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
/	/#	/!	ˆ	/@	?	‡
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
m	Ñ	¸	¥	»	**M**	ä
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
\	\#	œ!	œ	œ@	|	*f*
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
r	„	¶	‰	Ê	**R**	é
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
s	s#	s!	ú	s@	S	î
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
¹	¹#	¹!	š	¹@	>	ë
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
q	q$	qE	”	³	Q	ö
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
b	b#	b!	Æ	b@	B	ï
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
t	t$	tE	¬	t&	T	è
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
c	c$	cE	Ò	c&	C	Ó
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
x	**x#**	**x!**	¨	x@	X	å
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
n	n#	n!	Â	n@	**N**	**ñ**
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
ß	ß#	ß!	¾	ß@	"	®
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
k	k#	k!	µ	k@	K	÷
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
,	,#	,!	–	,@	<	—
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
w	ý	ê!	ê	ê&	W	ã
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
;	;#	›!	›	›@	:	â
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
z	z#	z!	²	z@	Z	ø
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
¢	¢$	¢E	Ï	¢&	™	Î

y	†	‘	Ã	ü	Y	×
d	Ç	d!	Ä	Á	D	ì
j	°	©!	©	Ë	J	í
g	g#	g!	U	g@	G	¯
-	-#	-!	È	-@	_	õ
=	=#	À	Å	~	+	ô
’	’#	’!	Ô	’@	ù	Õ
]	]#	É!	É	É@	}	ò
[	[	[!	Ú	[@	A	Û
f	û	ð	Í	Ø	F	æ
p	p$	pE	-	p&	P	±
v	v#	v!	Š	v@	V	‹
%	Đ	–	—	,	“	a
¡	•	…	Þ	à	O	
þ	ª	Ÿ	Ì	Ù	`	Æ*

The diacritic markings,

!	@	#	$	&	*	E~

Appendix 4: Sample of News records wrongly entered into the database

NewsId	Headline	Slug	Keyword	Classification code
ኢዜአ24292	ጀከለ	ጭፀጀ	ጭፀነ	ፖለ
ኢዜአ102650	?	ደደ	ደደ	ሕናፍ
ኢዜአ105142	ደረጀ	ደረጀ	ደረጀ	ሕናፍ
ኢዜአ112448	ታ	ታ	ታ	ሕናፍ
ኢዜአ87049	በበ	ለለ	ለለ	ሕናፍ
ኢዜአ108422	ዛሬ	ነገ	ትናንት	ጤናጥ
ኢዜአ114217	ሰ	ቸ		
ኢዜአ26515	ፈደደገሀገ	ተጀጀሀሀ	ኩጀከጀጀ	
ኢዜአ31139	መ·ከራ·	መ·ከራ·	መ·ከራ·	ጤናጥ
ኢዜአ44423	2	3	4	ጤናጥ
ኢዜአ75460	መ·ከራ·	ከከ	ከከ	ጤናጥ
ኢዜአ101742	ጸጸጸጸጸ	ጸጸጸጸ	ጸጸጸጸ	ዓአጉ

Appendix 5: Sample of News records taken for the experiment from the database

NewsID	HeadLine	Keywords	Slug	Classification code
ኢዜአ107352	ቤተ ክርስቲያኗ በአዋሳ ከተማ ኮሌጅ ከፈተች ።	ኮሌጅ ተከፈተ	ኮሌጅ	ትምህ
ኢዜአ107391	ትምህርት ቤት በንፋስ በመፈራረሱ ትምህርት ተቋረጠ	ትምህርት ቤት	ትምህርት ቤት	ትምህ
ኢዜአ105934	በከተሞች የስፖርት ውድድር ተካሔደ	ስፖርት	ስፖርት	ስፖር
ኢዜአ87366	በምስራቅ ወለጋ ከአንድ ሚሊየን በላይ የቡና ችግኝ ተዘጋጀ	ቡና	ቡና	ግብጉ
ኢዜአ103361	ኢትዮጵያ የአፍሪካ ህብረትም ፅህፈት ቤት መቀመጫ እንድትሆን ለዘለቄታዉ ተመረጠች።	ኢትዮጵያ	ለአፍሪካ ህብረት ጽህፈት ቤት	ዓአጉ
ኢዜአ103556	ስምንት ሀሰተኛ የብር ኖቶች ተያዙ	ሀሰተኛ ብር	ሀሰተኛ ብር	ሕናፍ
ኢዜአ95619	ለተቀናሽ የሰራዊት አባላት ከአንድ ነጥብ ስድስት ሚሊየን ብር በላይ ተሰጠ	ተቀናሽ ሰራዊት	ተቀናሽ ሰራዊት	መናጸ
ኢዜአ11320	ፓርቲዎችና የግል እጩ ተወዳዳሪዎች ለወረዳ 17 ሴቶች መርሀ ግብራቸውን አስተዋወቁ።	እጩ ተወዳዳሪዎች	እጩ ተወዳዳሪዎች	ፖለ
ኢዜአ12419	ጠቅላይ ሚኒስትር መለስ በከፍተኛ ድምፅ ለተወካዮች ምክርቤት አባልነት አለፉ	ጠቅላይ ሚኒስትር መለስ	ጠቅላይ ሚኒስትር መለስ	ፖለ

DECLARATION

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

Surafel Teklu Weldeselassie

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

W/ro Rahel Bekle

W/ro Woinshet Abdela

Ato Werkshet Lamenew