ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

BY

ANIMUT BELAY ASRES

JUNE 2012

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

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

BY

ANIMUT BELAY ASRES

JUNE 2012

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

BY

ANIMUT BELAY ASRES

Name and signature of Members of the examining board:

Name	**Title**	**signature**	**Date**
_____	Chairperson	_____	_____
Solomon Teferra (PhD)	Advisor	_____	_____
_____	Examiner	_____	_____

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university.

_____

Date

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

_____

Advisor

Acknowledgement

Above all, I would like to thank God for his invaluable helps in my entire life span.

Next, I would like to express my sincere appreciation to Dr. Solomon Tefera for advising this thesis. I appreciate the continuous support and timely advice he has given me. His encouragement helped shape the direction of my work. He has continuously encouraged me and bigheartedly guided me on semi-supervised machine learning approach.

I am deeply indebted to Dr. Million Meshesha since he has continuously generously guided and supported me on the fundamentals of semi-supervised machine learning.

I would like to express my gratitude to Ato Erimias. He supported and gave me most of the Amharic news corpus that I used for this research.

I would like to express my gratitude to my best friend Ato Getahun wassie for his valuable suggestions and helpful comments.

I wish to thank my colleagues in Information Science department; especially Ato Tigabu Akal, Girma Debele and Ato Amare Mekonnen from Electrical and computer engineering department supported me and prevent memory problems that I faced by giving their computer.

Finally, I wish to thank my parents specially my mother W/o Tachawt Tadele and my father Ato Belay Asres for their continuous encouragement and support.

List of Acronyms and Abbreviations

SVM	Support Vector Machine
NB	NaiveBayes
K-NN	K-Nearest Neighbor
KDT	Knowledge Discovery in Text
ENA	Ethiopian News Agency
EM	Expectation Maximization
GMM	Gaussian Mixture Model
SSL	Semi-Supervised Learning
RBFN	Radial Basis Function Network
DF	Document Frequency
IDF	Inverse Document Frequency
SMO	Sequential Minimal Optimization
MI	Mutual Information Gain
TS	Term Strength
IG	Information Gain

Table of Contents

List of Tables

List of Figures

List of appendixes

Abstract

Text classification is getting more attention and there is an increasing need for text classification technique that provides automatic, fast, and accurate classification with the least human interaction with such systems. Many techniques of supervised learning and unsupervised learning do exist in the literature for data classification. Semi-supervised learning is halfway between the supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information but not necessarily for all example data.

The paper explored the semi-supervised text classification which is applied to different types of vectors that are generated from the Amharic text documents. 3,154 news articles were used to do this research. To come up with good results document preparation and preprocessing was done. Weka package is used for the classification of the preprocessed data. Machine learning techniques, Expectation maximization clustering algorithm with Naïve Bayes, Hyperpipe, and RBF Network classification algorithm were used to categorize the Amharic news items.

The accuracy of the classifiers was better when the number of classes is less. The best result was obtained by the Naïve Bayes , Hyperpipe and RBF Networks classifiers with four classes (83.44 %, 82.8 and 82.4%) and the least performance is shown on the 10 categories (55.42%,57.26% and 51.9%) respectively. This research indicated that Naïve Bayes is more applicable to semi-supervised categorization of Amharic news items.

Keywords: Text categorization, semi-supervised machine Learning, Naïve Bayes, Hyperpipe and RBF Networks

CHAPTER ONE

INTRODUCTION

1.1. Background

In today's world, communicating with others via internet has become an integral part of life. It is hard to find a college student, professional, or any educated person for that matter, who does not use internet and send or receive e-mails. Also, it is an established fact that a lot of the communication that occurs within companies and organizations is nowadays done by e-mails, rather than memos or common bulletin boards. Modern Information Technologies and Web-based services are faced with the problem of selecting, filtering and managing growing amounts of textual information to which access is usually critical. Information Retrieval (IR) is seen as a suitable methodology for automated management of information/knowledge as it includes several techniques that support an accurate retrieval of information and the consequent user satisfaction. Among others, the classification of electronic documents in general categories (e.g., Sport, Politic, Economy...) is an interesting means to improve the performances of IR systems (Hirotoshi, 2002). It helps users to more easily browse the set of documents of their own interests; sophisticated IR models can also take advantages of the categorized data. Automatic organization of documents has become an important research issue since the explosion of digital and online text information.

From the early 1990s a lot of work has been made in document classification tasks. The effectiveness of many studies has dramatically improved thanks to the introduction of Machine Learning methods into the Text Classification community.

Document classification can be defined as the process of assigning text documents to predefined classes. Text classification can be made manually or automatically. Each of them has advantage and disadvantage to the user (Gebrehiwot, 2011).Two of the most widely-used methods in machine learning for prediction and data analysis are classification and clustering (Duda, 1997).

Classification is a supervised task, where supervision is provided in the form of a set of labeled training data, each data point having a class label selected from a fixed set of classes (Mitchell,

1997). The goal in classification is to learn a function from the training data that gives the best prediction of the class label of unseen (test) data points.

Generative models for classification learn the joint distribution of the data and class variables by assuming a particular parametric form of the underlying distribution that generated the data points in each class, and then apply Bayes Rule to obtain class conditional probabilities that are used to predict the class labels for test points drawn from the same distribution, with unknown class labels (Ng, 2002). In the discriminative framework, the focus is on learning the discriminant function for the class boundaries or a posterior probability for the class labels directly without learning the underlying generative densities (Jaakkola, 1999). It can be shown that the discriminative model of classification has better generalization error than the generative model under certain assumptions (Vapnik, 1998), which has made discriminative classifiers, e.g., support vector machines (Joachims, 1999) and nearest neighbor classifiers (Devroye, 1996), very popular for the classification task.

Clustering is an unsupervised learning problem, which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular similarity metric (Jain, 1988). Here, the learning algorithm just observes a set of points without observing any corresponding class/category labels. Clustering problems can also be categorized as generative or discriminative. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the most general formulation, the number of clusters K is also considered to be an unknown parameter. Such a clustering formulation is called a "model selection" framework, since it has to choose the best value of K under which the clustering model fits the data. In the discriminative clustering setting (e.g., graph-theoretic clustering), the clustering algorithm tries to cluster the data so as to maximize within-cluster similarity and minimize between-cluster similarity based on a particular similarity metric, where it is not necessary to consider an underlying parametric data generation model. In both the generative and discriminative models, clustering algorithms are generally posed as optimization problems and solved by iterative methods like EM (Dempster, 1977), approximation algorithms like KMedian. As the number of clusters and documents increase, the clustering solutions produced by k-means and bisecting k-means become more internally cohesive and externally isolated. However, the

clustering results do not match better with the pre-defined classes and requires relatively high computational requirements (Lakechew, 2011).

Both of them have their own advantages and disadvantages. Supervised algorithms assume that the category structure or hierarchy of a text database is already known. They require a training set of labeled documents and return a function that maps documents to the pre-defined class labels. Knowing the category structure in advance and generation of correctly labeled training set are very challenging or even impossible in large and dynamic text databases.

In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters.

Recently, there has been a lot of interest in the continuum between completely supervised and unsupervised learning (Ghani, 2003). Many document classification tasks are most of all supervised learning problems, since the process learns from pre-classified labeled documents.

1.2. Statement of the Problem

In many practical learning domains (e.g. text processing, bioinformatics), there is a large supply of unlabeled Amharic data but limited labeled Amharic data, which can be expensive to generate. To prevent this problems numerous researches have been conducted such as Zelalem (2001), Surafel (2003), Yohannes (2007), Worku (2009), Alemu (2010), Zeleke (2010) and Lakechew(2011) were conducted. Except the last researcher, who did on unsupervised text categorization all researchers have done on supervised text categorization. Supervised text categorization has the following limitations

First, it can be ambiguous: objects might have non-unique labeling or the labeling themselves may be unreliable due to a disagreement among experts.

Second, it uses limited vocabulary: Typical labeling setting involves selecting a label from a list of pre-specified labels which may not completely or precisely describe an object.

Third, supervised learning algorithms require a large, often excessive, number of labeled training documents for the accurate learning (Kohavi, 1996). Since the application area of automatic text categorization has diversified from articles and web pages to electronic mails and newsgroup postings, it is a difficult task to create training data for each application area (Nigam, 2000).

According to Nigam et al. (Nigam, 2000), in supervised text classification, obtaining training labels is expensive in huge volume of document collection. This is because in supervised text classification labeling of training data is done by a person manually and this is a time consuming, cumbersome and error prone process.

Fourth, Ozgur (Ozgur, 2004) concludes that unsupervised text classification techniques perform better in terms of time complexity and the quality of clusters produced as compared to supervised techniques. This shows that the overall similarities of the clustering solutions obtained by the unsupervised techniques are higher than the supervised ones.

Fifth, supervised text classification algorithms are expensive and time consuming to organize documents in to their categories. As Nigam et al. (Nigam, 2000) suggests, text clustering is a useful and inexpensive way to organize vast text repositories into meaningful topic categories. Furthermore, text clustering offers a low cost alternative to supervised classification, which relies on expensive and difficult handwork to label training data (Massey, 2004).

On the other hand unsupervised learning has the following limitations

First, unsupervised learning is more difficult problem than supervised learning due to the lack of a well-defined user-independent objective. Due to this reason, it is usually considered an ill-posed problem that is exploratory in nature; that is, the users are expected to validate the output of the unsupervised learning process. Devising a fully automatic unsupervised learning algorithm that is applicable in a variety of data settings is an extremely difficult problem, and possibly infeasible (Kang, 2003).

Second, unsupervised learning is less accurate than supervised text classifier. Since unsupervised learning is natural grouping because of nosy data different documents may be classified in the same group.

Third, unsupervised text classification algorithm is presumably the drive to create and apply explicit rules led to increase study and test phase response times when compared with the incidental conditions (Bradley, 2002).

Forth, interestingly, the subjects in the intentional conditions also performed worse in the filler task involving arithmetic problems, possibly indicating increased fatigue or the attempted rehearsal of study-phase items (Bradley, 2002).

Different from the unsupervised techniques, the supervised techniques use class label information in addition to the similarity information between documents. For this reason, it is

expected that the clusters (groups) obtained by the supervised techniques are of higher quality compared to the unsupervised techniques. However, the best performers of the unsupervised techniques k-means and bisecting k-means achieve generally better performance than NaiveBayes and not much worse performance than k-NN, which are supervised techniques, in terms of entropy, purity, overall similarity and F-measure. In the supervised document classification there may be bias or misclassification. Another observation is that, compared with the supervised techniques the unsupervised techniques generally achieve higher overall similarity performance. This is due to the fact that they make decisions depending only on the similarity information between documents. On the other hand the supervised techniques use a labeled training set. This observation has made us think that there may be some outliers in the labeled training set that leads to decrease in the overall similarity of the clusters obtained and unsupervised techniques can be used to enhance the task of pre-defining categories and labeling documents in the training set. Consequently, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. Unlabeled data is available in abundance, but it is difficult to learn the underlying structure of the data. Labeled data is scarce but is easier to learn from. Semi-supervised learning is designed to alleviate the problems of supervised and unsupervised learning problems, and has gained significant interest in the machine learning research community. The framework of semi-supervised approach is applicable to both classification and clustering. Semi-supervised classification algorithms train a classifier given both labeled and unlabeled data.

Most semi-supervised learning algorithms developed to modify existing supervised or unsupervised algorithms, or devise new approaches. Semi-supervised classification has received significant amount of interest, as it provides a way to utilize the large amount of readily available unlabeled data for improving the classifier performance. Semi-supervised classification has been successfully applied to various applications in computer vision and machine learning, such as text classification, human computer interaction, content based image retrieval , object detection, person identification , relevance feedback, computational linguistics and protein categorization, to name a few. Similarly, side-information such as pairwise constraints has been utilized to improve the performance of clustering algorithms by aiding them in arriving at a clustering desired by the user. Semi-supervised learning continues to pose both theoretical and practical questions to researchers in the machine learning. There is also an increasing interest in the fields

of cognitive sciences and human psychology since there are demonstrated settings where humans performed semi-supervised learning (Kang, 2003).

In order to maximize the accuracy, similarity performance, and other things the researcher has explored application of semi-supervised approach for text classification for Amharic documents.

To this end, this study attempts to address the following research questions:

- ✓ Which classification algorithm is best for classifying Amharic text documents?
- ✓ Which clustering algorithm is more suitable for labeling unlabeled Amharic text documents?
- ✓ Which learning approach is more appropriate for creating classification model that helps in Amharic text categorization?
- ✓ To what extent the semi-supervised model is able to predict according to the experts judgment?

1.3. Objective of the study

1.3.1. General Objective

The main objective of this study is to explore automatic Amharic text classification, using both labeled and unlabeled data or supervised and unsupervised machine learning approaches.

1.3.2. Specific objective

The specific objectives are:

- To apply semi-supervised learning approach that improves the existing and established supervised and unsupervised learning algorithms without having to modify them. That is an algorithm that utilizes unlabeled data along with the labeled data while training classifiers.
- To review literature on the concepts, techniques and tools of text classification particularly in the area of semi-supervised learning.
- To select suitable techniques for classifying Amharic text news.
- To design Amharic news text classification architecture
- To design a prototype Amharic text news classifier.
- To evaluate the performances of the prototype Amharic text news classification system.
- To recommend research direction for future work(s) in the area of automatic Amharic text classification.

1.4. Methodology

The methodology that was used for this study includes knowledge discovery in text (KDT) approach that is recommended by Karanikas et al. (Arzucan, 2002). KDT is a multi-step process, which includes all the tasks from the gathering of documents to the visualization of the extracted information. In this research, the three phase KDT process used to achieve the above objectives and it is adapted from the previous researcher.

1.5. Literature review

To get more information about text categorization and the tools and techniques that the researchers used before the researcher referred to text books and also discussion with ENA workers was held.

1.6. Data source and data set preparation

The data sets collected from Ethiopian News Agency (ENA) because in this organization almost all works or classifications are made manually, which is tedious. The researcher used the data collected and used by the previous researchers who did their research in ENA and also collect additional data from ENA. The format of data sets converted in to text for pre processing. The preprocessing had two steps. The first step was removing the irrelevant characters from the corpus. The second step was performing the preprocessing. Both tasks were done using python.

1.7. Design procedures

Categorization of Amharic documents has been developed using three steps. These are preprocessing, clustering and classification. The first task is to preprocess, which includes tokenization, normalization, stop words removal and stemming. The second task is to cluster documents in a structured organization of semi labeled classes.

The idea is to perform a preliminary classification of documents using only the labels associated with the categories and the relationships between classes. This is the unsupervised classification problem. The third task is to classify documents in a structured organization of full labeled classes.

1.8. Tools and Techniques

As it is mentioned, the main objective of this study is to categorize Amharic documents. Text categorization includes clustering and classification of different Amharic documents. This task handled by using weka, because it is easy to use and has a graphical user interface.

1.9. Evaluation Techniques

We need evaluation methods to compare various text classifiers. Evaluation of a classifier can be conducted by measuring its efficiency and its effectiveness. Efficiency is typically measured by using the elapsed processor time and it refers to the ability of a classifier to run fast. Efficiency of a classifier can usually be measured on two dimensions: learning efficiency (i.e., the time a machine learning algorithm takes to generate a classifier from a set of training examples) and categorization efficiency (i.e., the time the classifier takes to assign appropriate categories to a new document). Because of the unstable nature of parameters on which the evaluation depends, efficiency is rarely used as the singular performance measure in text categorization. However, efficiency is important for the practical application of the system.

A much more common evaluation method for text categorization systems is effectiveness: this refers to the ability to take the right decisions on the categorization of new incoming documents. There are several commonly used performance measures of effectiveness. However, there is no agreement on one single measure for use in all applications. Indeed, the type of measure that is preferable depends on the characteristics of the test data set and on the user's interests. The absence of one optimal measure of effectiveness makes it very difficult to compare the relative effectiveness of classifiers. The system has been evaluated based on its efficiency and effectiveness.

1.10. Scope of the study

The scope of this study is limited to investigate the feasibility of designing Amharic news text classification system using semi-supervised approach. In this study, different classification algorithms, such as, Naive Bayes, HyperPipe, RBF Networks and the EM clustering algorithms were used. The study is limited only to classification of text news items from ENA. HTML documents, image documents and others were not considered in this study.

1.11. Significance of the study

Semi-supervised Learning (SSL) takes advantage of a large amount of unlabeled data to enhance classification accuracy. Its application to text categorization is stimulated by the easy availability of an overwhelming number of unannotated documents, in contrast to the limited number of annotated ones. Intuitively, corpora with different topics may not be content wise related, however, word usage exhibits consistent patterns within a language. The main purpose of this work is to address Amharic news classification task. This can be explored in a process that exploits a taxonomic structure, approaching both the unsupervised and the supervised problem. The first goal can be conceived as the activity of finding a hypothesis of the right location of a document taking into account only the structured organization of classes. Then locate each cluster in different categories. This enables the user to find and locate Amharic documents in simple and easy way.

1.12. Thesis Organization

This thesis is organized into six chapters: Chapter 1 - Introduction; Chapter 2- Literature Review; Chapter 3 – The Amharic Language and its Writing system; Chapter 4 - Methodology Chapter-5 Experiment and Performance Evaluation and Chapter 6 – Conclusion and Recommendations.

Chapter one includes background, statement of the problem, objectives of the study, methodology, scope and applications of the study. Chapter 2 discuses different text classification approaches, document preprocessing and representation, overview of the different classification and clustering algorithms and evaluation techniques. Chapter 3 gives highlight about Amharic writing system. Chapter 4 discusses details of methodology adopted and chapter 5 presents the experimental results and findings of the study. In chapter 6 summarizes findings of this study and recommendations are given for further research.

CHAPTER TWO

LITERATURE REVIEW

Introduction

With the ever-increasing volume of text data from various online sources, it is an important task to categorize or classify these text documents into categories that are manageable and easy to understand. Text classification is the task of assigning previously unseen documents to appropriate predefined categories. The task is commonly described as follows: Given a set of labeled training documents of n classes, the system uses this training set to build a classifier, which is then used to classify new documents into the n classes. The problem has been studied extensively in information retrieval, machine learning and natural language processing. The supervised machine learning approach makes this automatic, by learning classifiers from a set of training examples. For most supervised learning algorithms, building accurate classifiers needs a large volume of manually labeled examples. This manual labeling process is time-consuming, expensive, and will have some level of inconsistency (kang, 2003). This problem motivates the researcher work towards a text categorization system that can achieve a satisfactory level of performance with fewer training examples.

Many machine learning algorithms have been developed and applied to the construction of classifiers. They can usually be grouped into rule-based, probability based, and similarity-based learning algorithms. This thesis focuses on the similarity based approach. This builds upon the large volume of previous work in the area of text categorization that has adopted this approach. The similarity-based approach offers the possibility of exploring statistical information that may capture the target concepts hidden in documents (kang, 2003).

2.1. Text Categorization

In this section, the paper gives an overview of text categorization. First defines the text categorization task and discusses ambiguities in most natural languages on which classifiers should be built. Then, discusses two general approaches, “knowledge engineering” and “machine learning”, to the construction of classifiers and why the researcher is focusing on a machine

learning approach. This section also describes characteristics of the domain of text categorization that make this task difficult for a machine learning approach.

2.1.1. A Definition of Text Categorization

Text classification (also known as text categorization) is the automated assignment of natural language text to appropriate thematic categories, based on its content. A set of categories is predefined manually (Arzucan, 2002).

Two different types of text categorization task can be identified depending on the number of categories that could be assigned to each document. The first type, in which exactly one category is assigned to each dj $\in$ D, is regarded as the single-class (or nonoverlapping categories) text categorization task. The second type, in which any number of categories from zero to |C| may be assigned to each dj $\in$D, is called the multi-class (or overlapping categories) task (Sebastiani, 2002). A special type of multi-class text categorization is one where each document is assigned to the same number k, where k > 1, of categories. The answer to the question of which type of text categorization should be adopted for a given text categorization system depends on the application and characteristics of the corpus (kang, 2003).

Most semi-supervised learning methods are extensions of existing supervised and unsupervised algorithms. Therefore, before introducing the developments in semi-supervised learning literature, it is useful to briefly review supervised and unsupervised learning approaches.

2.1.2. Ambiguities in Natural Language Text

In most content-based text classification systems, an important issue is how they can capture the meaning of the natural language texts. Obtaining accurate classifiers requires the system to understand the natural languages at some level. Understanding natural languages, however, is a difficult task due to ambiguities in them:

1. The same sentence may have different meanings. For example, consider a sentence like "Salespeople sold the dog biscuits" (an example from Charniak, 2003). This sentence can be interpreted in two different ways: (1) the salespeople are selling the dog-biscuits and (2) the salespeople are selling biscuits to dogs.
2. There is the large number of synonyms – syntactically different words with the same or similar meanings in natural languages. It is regarded as good writing style not to

repeatedly use the same word for expressing a particular idea (or concept). Synonyms allow the same idea to be expressed by different words that have a similar meaning.

3. Polysemy refers to an ambiguity where words which are spelled the same can have different meanings in different sentences or documents. For example, the word "bat" may mean (1) an implement used in sports to hit the ball or (2) a flying mammal.

Resolving such ambiguities is probably beneficial to text categorization when there are many words in common across categories, even though it may not have a huge impact on the overall text categorization performance.

2.1.3. Knowledge Engineering versus Machine Learning Approach

There are two different ways of constructing classifiers, the function: $D \times C \{T, F\}$. They are "knowledge engineering" and "machine learning" approaches. In the knowledge engineering approach, human experts (including knowledge engineers and domain experts) manually create a set of rules that correctly categorize previously unseen documents under given categories. While allowing for semantically-oriented text categorization, by defining controlled vocabularies which can be interpreted by the text categorization system (Brasethvik, 2001), manually determining such a solution imposes a considerable workload on human experts. This makes it time consuming and expensive.

Also, this manual approach may cause inconsistency since human experts often disagree on the assigned categories of documents and even one person may categorize documents inconsistently (Apte, 1994). As a result, these problems for the knowledge engineering approach cause the bottleneck of encoding large amounts of incomplete and potentially conflicting expert knowledge.

The machine learning approach to text categorization is to automatically build the classifiers by learning the concept descriptions of the categories. One type of machine learning, applied to text categorization, is "supervised learning". This requires a set of pre-labeled (pre-categorized) training documents for generating classifiers. By contrast, "unsupervised learning" refers to the task of automatically identifying a set of categories from a set of unlabeled documents and grouping these unlabeled documents under these identified categories (Merkl, 1998).

The advantages of the machine learning approach over knowledge engineering are the considerable reduction in the volume of work required from human experts, consistent text categorization, and the capability of easily adjusting the generated classifier to handle different types of documents (such as newspaper articles, newsgroup postings, electronic mails, etc.) and even languages other than English.

2.1.4. Difficulties for the Machine Learning Approach

The unstructured format of natural language text and the diversity of target concepts associated with the categories, present interesting challenges to the content based application of machine learning algorithms. The large number of input features, that seem necessary for the construction of classifiers, overwhelms most text categorization systems. For most machine learning algorithms, increasing the number of features means that they have to use more training examples to obtain the same level of text categorization performance. This large number of training examples and features may be computationally intractable for most machine learning algorithms, by requiring unacceptably large processing time and memory.(kang, 2003)

Of the large number of features, there are usually many features that appear in most documents. These words can be considered irrelevant, in the sense that such features are evenly distributed throughout documents and, as a result, have no discriminating power. It is important for the efficiency and effectiveness of the system to select an efficient subset of features, by removing these irrelevant ones. However, it is a difficult task since a reasonable feature subset size might be different across the categories and some informative features for a given category could be distributed across several categories. For example, depending on the level of concept complexity, some categories require a large number of features to describe their concepts while others need a relatively small number of features. Also, informative features in the overlapping categories might be evenly distributed across such overlapping categories and could be considered as irrelevant ones.(kang, 2003)

2.2. Text categorization approaches

2.2.1. Supervised learning

Supervised learning aims to learn a mapping function f: X → Y, where X and Y are input and output spaces, respectively (e.g. classification and regression (Duda, 2000). The process of learning the mapping function is called training and the set of labeled objects used is called the training data or the training set. The mapping, once learned, can be used to predict the labels of the objects that were not seen during the training phase. Several pattern recognition (Duda, 2000) and machine learning (Mitchell, 1998) textbooks discuss supervised learning extensively. A brief overview of supervised learning algorithms is presented in this section.

Supervised learning methods can be broadly divided into generative or discriminative approaches. Generative models assume that the data is independently and identically distributed and is generated by a parameterized probability density function. The parameters are estimated using methods like the Maximum Likelihood Estimation (MLE), Maximum A Posteriori estimation (MAP) (Duda, 2000), Empirical Bayes and Variational Bayes (Bishop, 2006). Probabilistic methods could further be divided into frequentst or Bayesian. Frequentst methods estimate parameters based on the observed data alone, while Bayesian methods allow for inclusion of prior knowledge about the unknown parameters. Examples of this approach include the Naive Bayes classifier, Bayesian linear and quadratic discriminants to name a few.

Instead of modeling the data generation process, discriminative methods directly model the decision boundary between the classes. The decision boundary is represented as a parametric function of data, and the parameters are learned by minimizing the classification error on the training set (Duda, 2000). Empirical Risk Minimization (ERM) is a widely adopted principle in discriminative supervised learning. This is largely the approach taken by Neural Networks (Bishop, 2005) and Logistic Regression (Bishop, 2006). As opposed to probabilistic methods, these do not assume any specific distribution on the generation of data, but model the decision boundary directly.

Most methods following the ERM principle suffer from poor generalization performance. This was overcome by Vapnik's (kang, 2003) Structural Risk Minimization (SRM) principle which adds a regularity criterion to the empirical risk that selects a classifier with good generalization

ability. This led to the development of Support Vector Machines (SVMs) which regularize the complexity of classifiers while simultaneously minimizing the empirical error. Methods following ERM such as neural networks and Logistic Regression are extended to their regularized versions that follow SRM (Bishop, 2006).

2.2.1.1. Classification Algorithms

Supervised algorithms assume that the category structure or hierarchy of a text database is already known. They require a training set of labeled documents and return a function that maps documents to the pre-defined class labels. As discussed previously, knowing the category structure in advance and generation of correctly labeled training set are very challenging or even impossible in large and dynamic text databases.

A wide range of classification algorithms have been developed through time with different underlying models and different theories of how a classifier should be built. These algorithms have different inductive biases that affect their performance on a data set, and consequently, it is important to find the inductive bias that best fits the data set. This can be done empirically by applying a set of different machine learning algorithms and selecting the algorithm that performs the best(Stig-Erland, 2007).

In this section the paper discuss the most popular supervised algorithms.

2.2.1.1.1. Bayes

Bayesian algorithms are based on Bayes' Theorem, which is defined as

$$P(h|d) = \frac{P(d|h)Pr(h)}{P(d)} \quad \text{.....equation 1}$$

where the h corresponds to a hypothesis, namely a prediction of a particular class, and the d represents the attributes of the unlabeled instance.

Naive Bayes

The naive Bayes (NB) classifier is a probabilistic model that uses the joint probabilities of terms and categories to estimate the probabilities of categories given a test document (Mitchell, 1999). The naive part of the classifier comes from the simplifying assumption that all terms are conditionally independent of each other given a category. Because of this independence

assumption, the parameters for each term can be learned separately and this simplifies and speeds the computation operations compared to non-naive Bayes classifiers.

There are two common event models for NB text classification, discussed by McCallum and Nigam , multinomial model and multivariate Bernoulli model. In both models classification of test documents is performed by applying the Bayes' rule (Mitchell, 1999):

$$P(cj|jdi) = \frac{P(cj) \cdot P(dij|cj)}{P(di)} \quad \text{……equation 2}$$

where di is a test document and cj is a category. The posterior probability of each category cj given the test document di, i.e. P(cj|di), is calculated and the category with the highest probability is assigned to di. In order to calculate P(cj|di), P(cj) and P(di|cj) have to be estimated from the training set of documents. Note that P(di) is same for each category so we can eliminate it from the computation.

Multinomial Model

In the multinomial model a document di is an ordered sequence of term events, drawn from the term space T. The naive Bayes assumption is that the probability of each term event is independent of term's context, position in the document, and length of the document. So, each document di is drawn from a multinomial distribution of terms with number of independent trials equal to the length of di(kang, 2003).

Multivariate Bernoulli Model

Multivariate Bernoulli model for naive Bayes classification is the event model. In this model a document is represented by a vector of binary features indicating the terms that occur and that do not occur in the document.

Here, the document is the event and absence or presences of terms are the attributes of the event. The naive Bayes assumption is that the probability of each term being present in a document is independent of the presence of other terms in a document.

To state differently, the absence or presence of each term is dependent only on the category of the document.

Different from the multinomial model, the multivariate Bernoulli model does not take into account the number of times each term occurs in the document, and it explicitly includes the non-occurrence probability of terms that are absent in the document (McCallum,1998).

2.2.1.1.2. Lazy

Lazy learning algorithms differ from the other classification algorithms in that the training of the classifier is postponed until classification. This allows the classifier to be customized according to each unlabeled instance at the expense of being computational intensive if there are many instances to classify(Stig-Erland, 2007).

IB1

IB1 is a nearest neighbour algorithm that determines the class of an unlabeled instance according to the class of the nearest training instance. The distance between two instances are calculated using the euclidean distance(Stig-Erland, 2007).

$$\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$ ……equation 3

where ai and bi are the attributes i of the instances a and b.

IBk

IBk is similar to IB1, but it uses the k nearest neighbours instead of only one. The predicted class is determined by the majority vote where each instance places a vote on its corresponding class(Schmitter, 2006).

K*

K* is a nearest neighbour algorithm that employs, instead the Euclidean distance, an entropic distance function computing the probability of randomly transforming one instance into another. Each class receives a vote from each instance with a weight equal to the distance from it to the unlabeled instance, and the class with the most votes is selected (Liszka,1999).

Locally Weighted Learning

Locally weighted learning (LWL) selects a subset of the training instances, where each instance is weighted according to the unlabeled instance.

A k nearest neighbour algorithm is applied to select the subset of instances, and the weight is calculated by a weighting function taking the euclidean distance as input.

2.2.1.1.3. Functions

These algorithms have mathematical or statistical foundations and create models that can be represented mathematically through functions. They are a mix of regression and classification algorithms (Stig-Erland, 2007).

Linear Regression

Linear regression (LinReg) is a standard linear regression algorithm that expresses the numerical class as a linear combination of the attributes. The coefficients of these attributes are calculated using the least-square method (Stig-Erland, 2007).

Logistic

Logistic builds logistic regression models and is implemented according to with some modifications. These models have similar properties to linear regression models, but the target attribute is transformed using the logit function and the weights are found by maximizing the log-likelihood instead of minimizing the sum of squared errors (Wang, 1993).

Simple Logistic

Simple Logistic (SLogistic) also builds logistic regression models, but it uses another strategy than Logistic involving LogitBoost and a base learner constructing simple regression models containing only the attribute yielding the minimum squared error. The number of boosting iterations used is determined by cross-validation (Witten, 2005).

Multilayer Perceptron

Multilayer Perceptron (MP) is a neural network algorithm that optimizes the weights of neural network using back propagation. The input layer comprises a bias node in addition to a node for each attribute after the nominal attributes have been converted to binary attributes. The hidden layer also contains a bias node in addition to n nodes determined by the following expression

$$\frac{i+o}{2}$$ ……equation 4

where i and o is the number of nodes in the input and output layer. The output layer is composed of a node for each class (Stig-Erland, 2007). The activation function for the nodes in the hidden and output layer is the sigmoid function.

RBF Network

RBF Network (RBFN) trains a radial basis function network, which is a type neural network. The network has three layers: an input layer with a node for each attribute; a hidden layer where each node has a Gaussian radial basis function as activation function, created using a clustering method called KMeans (Martin, 1995); and an output layer containing a node for each class with sigmoid as activation function.

SMO

SMO, proposed by John Platt, is a sequential minimum optimization algorithm for training support vector machines (SVM). The algorithm finds the maximum margin hyperplane represented as a set of vectors known as support vectors. In order to solve non-linear problems with this linear classifier, the instance space is transformed using a non-linear kernel function.
We chose to use the default polynomial kernel (Stig-Erland, 2007).

Support Vector Machines

Support Vector Machines (SVM) is a technique introduced by Vapnik in 1995, which is based on the Structural Risk Minimization principle. It is designed for solving two-class pattern recognition problems. The problem is to find the decision surface that separates the positive and negative training examples of a category with maximum margin.

For the linearly separable case, the decision surface is a hyperplane that can be written as (Yang, 1999):

$w * d + b = 0$ ……equation 5

where d is a document to be classified, and vector w and constant b are learned from the training set. The SVM problem is to find w and b that satisfy the following constraints (Joachims, 1998):

Minimize $\|w\|^2$

so that $\forall i : yi(w * d + b) \geq 1$……equation 6

Here, $i \in \{1, 2 \ldots N\}$, where N is the number of documents in the training set; and y_i equals +1 if document di is a positive example for the category being considered and equals -1 otherwise.

Most of the classifiers implicitly or explicitly require the data to be represented as a vector in a suitable vector space, and are not directly applicable to nominal and ordinal features (Tan, 2005). Also, most discriminative classifiers have been developed for only two classes. Multiclass classifiers are realized by combining multiple binary (2-class) classifiers, or using coding methods (Cohen, 1996).

Voted Perceptron

Voted perceptron (VP) (Freund, 1998) transforms the input space using a polynomial kernel as SMO, but it uses the perceptron (Rosenblatt, 1988) algorithm to train the classifier.

During training, it stores all the intermediate prediction vectors, namely the coefficients of the attributes, along with a weight of how many iterations they persisted without change. When classifying, each prediction vector votes on a class according to its weight, and the majority vote determines the predicted class.

2.2.1.1.4. Trees

These algorithms induce decision trees as classifiers, which basically contains two types of nodes: decision nodes and leaf nodes. Decision nodes are internal nodes containing a test on a specific attribute that determines which of the underlying branches an unlabeled instance should follow. Traversal continues from the root until a leaf node is encountered, and the leaf node predicts the class of the instance by utilizing a prediction function.

Decision trees is one of the earliest classifier (Sahami, 1998), that can handle handle a variety of data with a mix of both real, nominal, missing features and multiple classes. It also provides interpretable classifiers, which give a user an insight about which features are contributing for a particular class being predicted for a given input example. Decision trees could produce complex decision rules, and are sensitive to noise in the data. Their complexity can be controlled by using approaches like pruning; however, in practice classifiers like SVM or Nearest Neighbor have been shown to outperform decision trees on vector data.

Decision tree learning is composed of building and pruning. A decision tree is typically built by recursively selecting the most promising attribute and splitting the training set accordingly until

all instances belong to the same class or all attributes have already been used. The most promising attribute is determined by the attribute maximizing the splitting criterion.

The role of pruning is to simplify the decision tree either during or after building(Stig-Erland, 2007).

ID3

ID3 is one of the first decision tree learners proposed, and it employs information gain as splitting criterion. Since it does not support continuous or missing attribute values, it can only solve a limited set of problems (Quinlan, 1986).

J4.8

J4.8 is an implementation of Quinlan's popular C4.5 (Quinlan, 1993) decision tree learner and it improves upon ID3 in several areas. First, it replaces the information gain splitting criterion with gain ratio since information gain favors attributes with many values. Second, it supports both continuous and missing values, and it performs pruning using error based pruning (EBP).

REPTree

REPTree is a fast decision tree learner. it uses information gain instead of gain ratio and reduce error pruning instead of EBP.

NBTree

NBTree is a hybrid algorithm that creates decision trees with Naive Bayes classifiers at the leaves learned from the training instances reaching the node. It follows the standard decision tree learning algorithm and uses the mean accuracy of creating a Naive Bayes classifier at a given node according to 10-fold cross-validation as splitting criterion(Kohavi, 1996).

Logistic Model Trees

Logistic Model Trees (LMT) (Landwehr, 2005) builds decision trees with logistic regression models at the leaves, which are iteratively created using Simple Logistic. The trees are built similarly to C4.5 by selecting attributes according to the gain ratio splitting criterion until there are no more attributes, all the instances have the same class or there are less than 16 instances. Pruning is performed using the pruning algorithm employed by the decision tree learner, CART (Breiman, 1984).

M5'

M5' (E. Frank, 1998) is a reconstruction of Quinlan's M5 (Quinlan, 1992) that creates decision trees with linear regression models at the leaves. It chooses the attribute at each decision node

that maximizes the standard deviation reduction of the class of the training instances reaching the node. When the tree is built, it traverses upwards from the leaves, while adding linear regression models at the nodes and possibly removing nodes if necessary. The predicted class value of an unlabeled instance is determined based on the output of all the linear regression models encountered when traversing the tree.

Decision Stump

Decision Stump (DS) induces simple decision trees, known as decision stumps, with only a single decision node. This node has a boolean test, which for a nominal attribute tests whether the attribute is equal to a specific value and for a continuous attribute tests whether the attribute is less or equal to a threshold. This algorithm is normally executed through ensemble algorithms like bagging and boosting (Stig-Erland, 2007).

Random Forest

Random Forest (RF) (Breiman, 2001) uses bagging in combination with a random tree inducer. The random tree inducer builds a tree by choosing at a given node the best attribute among a set of randomly selected attributes.

ADTree

ADTree (Freund, 1999) creates what is known as an alternative decision tree by using boosting to add the different branches. An alternative decision tree is simply a set of interconnected decision stumps with numerical leaves, where each leaf may be connected to a set of other stumps.

The tree is used to classify unlabeled instances with binary classes by summing all the numerical nodes encountered while following the different paths of the tree applicable for the instances. The sign of this value determines the predicted class.

2.2.1.1.5. Rules

This group contains algorithms that create classifiers which are rule sets. Rule sets are intuitive and easier for humans to interpret than other classifiers like decision trees.

JRip

JRIP is an implementation of RIPPER (Cohen, 1995) with some minor modifications added to fix what appear to be two bugs in the original algorithm. It induces each rule of the final rule set in two steps. Firstly, the rule is grown by continually adding antecedents until it matches only

training instances with a specific class. Secondly, the rule is iteratively pruned by processing the antecedents in reverse order.

OneR

OneR is a simple algorithm that creates a rule set for each attribute and chooses the rule set with the lowest error rate on the training data. Each rule set comprises a rule for each value of a particular attribute that predicts the majority class of the training instances matching the rule(Stig-Erland, 2007).

ZeroR

ZeroR is the simplest of all classification algorithms, and it only predicts the majority class of the training set. This algorithm provides an upper bound of the error rate that all other classification algorithms should be smaller than (Stig-Erland, 2007).

DecisionTable

DecisionTable (DT) (Kohavi, 1995) constructs a decision table classifier, which simply a table is containing the training instances with only a subset of their attributes included. The optimal subset of the attributes is found using best-first search combined with cross-validation where the DecisionTable algorithm is executed for different subsets. An unlabeled instance is classified as the majority class of the matching instances in the table, but if there are no matching instances, the majority class of all training instances is predicted instead.

PART

PART (Frank, 1998) creates a rule set by repeatably creating pruned decision trees using J4.8, converting them to rules and removing the training instances matching the rule until all training instances are covered by at least one rule.

Each rule is created according to the path from the root of the decision tree to leaf covering the most training instances. In order to preserve computational resources, only partial decision trees are constructed where branches are expanded as needed.

M5Rules

M5Rules (Holmes, 1999) builds regression rules using the same algorithm as Part except it generates trees using M5' instead of J4.8.

Ridor

Ridor is a RIpple DOwn Rule learner that first creates a default rule predicting the majority class of the training instances and then recursively adds exceptions to this rule until all training instances are classified correctly according to the rule set. A separate validation set is utilized to find the most accurate exception at each step (Stig-Erland, 2007).

NNge

NNge is a nearest neighbour algorithm forming non-nested general exemplars. A general exemplar is a hyper-rectangle that encompasses a set of training instances sharing the same class. In this way, each general exemplar is like a rule, and the nearest exemplar determines the class of an unlabeled instance(Stig-Erland, 2007).

2.2.1.1.6. Misc

This group contains the algorithms that do not fit naturally into any of the other groups.

HyperPipes

HyperPipes (HP) is a simple and extremely fast classification algorithm that constructs a set of attribute ranges for each class. For nominal attributes, the range is the set of values observed for a particular attribute of the training instances matching a specific class. The range is found similarly for continuous attributes except the range is not a subset, but an interval ranging from the minimum to the maximum observed attribute value. Classification is performed by selecting the class with the most matching attribute ranges (Stig-Erland, 2007).

VFI

VFI (G. Demiroz , 1997) constructs a set of intervals for each attribute similarly to Hyper- Pipes, but these intervals are not bound to a specific class. Thus, each interval contains a class count for each class according to the training instances that fall into it. Continuous attributes are basically

discretised into a set of intervals, and an interval for nominal attributes is defined as a single attribute value. An unlabeled instance is classified using the majority vote, where each matching attribute interval is allowed to vote.

2.2.1.1.7. Ensemble

Ensemble classifiers are meta-classification algorithms that combine multiple component classifiers (called base classifiers) to obtain a meta-classifier with the hope that they will perform better than any of the individual component classifiers. Bagging (Breiman, 1996) and Boosting (Freund, 1996) are the two most popular methods in this class. Bagging is a short form for bootstrap aggregation, which trains multiple instances of a classifier on different subsamples (bootstrap samples) of the training data. The decision on an unseen test example is taken by a majority vote among the base classifiers. Boosting, on the other hand, samples training data more intelligently by sampling examples that are difficult for the existing ensemble to classify with a higher preference.

Ensemble algorithms use a base learning algorithm to create an ensemble of classifiers and combine these classifiers to reach a prediction. These algorithms differ in how the base learning algorithm is applied and how they combine the classifiers. The first two algorithms enhance the abilities of the base learner, making it possible to solve previously unsupported problems, while the last two enhance the performance of the base learning algorithm(Stig-Erland, 2007).

ClassificationViaRegression

ClassificationViaRegression allows a regression algorithm to solve classification problems. It creates a data set for each class using a 1-against-all encoding where the class is 1 if it is equal to the current class and 0 otherwise. A regression model is created for each class based on these data sets, and classification is performed by predicting the class belonging to the model yielding the greatest value (Stig-Erland, 2007).

MultiClassClassifier

MultiClassClassifier makes it possible to solve multi-class problems with algorithms that only support binary classes. This is possible through several methods, but we chose the default 1-against-all method explained in the previous section.

Bagging

Bagging (Breiman, 1996) creates an ensemble of classifiers in order to increase the accuracy by stabilizing the base learning algorithm, or in other words decrease its variance. This is done by generating a set of ”new” training sets using bootstrapping and applying the base learning algorithm on these data sets.

Prediction is determined by the majority vote of the ensemble. The success of bagging depends heavily on the properties of the base learning algorithm. It should be unstable, meaning that it is sensitive to small changes in the training set, so that its variance can be decreased.

AdaBoost.M1

AdaBoost.M1 (Freund, 1996) is a boosting algorithm that builds an ensemble of classifiers by forcing the base learning algorithm to focus on the instances that the previous classifiers had problems classifying correctly. This is done by accompanying every training instance with a weight representing the severity of misclassification such that the error rate is calculated as the sum of the weights of the misclassified instances divided by the sum of all the weights.

Initially, each instance has equal weights, but after a new classifier is induced, the weights are updated so that misclassified instances increase in weight while the other instances decrease.

K -Nearest Neighbor Classification

K-NN (k-nearest neighbor) classification is a popular instance-based learning method (Mitchell) that has been shown to be a strong performer in the task of text categorization (Yang, 1999).

The algorithm works as follows: First, given a test document x, the k nearest neighbors among the training documents are found. The category labels of these neighbors are used to estimate the category of the test document. In the traditional approach, the most common category label among the k-nearest neighbors is assigned to the test document.

Weighted k-NN is a refinement to the traditional approach. In weighted k-NN, the contribution of each of the k nearest neighbors is weighted according to its similarity to the test document x. Then, for each category, the similarities of the neighbors belonging to that category are summed

to obtain the score of the category for x. That is, the score of category cj for the test document x is(Arzucan, 2002)

$$score(c_j, \mathbf{x}) = \sum_{\mathbf{d_i} \in N(\mathbf{x})} cos(\mathbf{x}, \mathbf{d_i}) \cdot y(\mathbf{d_i}, c_j)$$ .....equation 7

where di is a training document; N(x) is the set of the k training documents nearest to x; cos(x; di) is the cosine similarity between the test document x and the training document di; and y(di; cj) is a function whose value is 1 if di belongs to category cj and 0 otherwise. The test document x is assigned to the category with the highest score.

2.2.2. Unsupervised learning

Unsupervised learning or clustering is a significantly more difficult problem than classification because of the absence of labels on the training data. Given a set of objects, or a set of pair wise similarities between the objects, the goal of clustering is to find natural groupings (clusters) in the data. The mathematical definition of what is considered a natural grouping defines the clustering algorithm. A very large number of clustering algorithms have already been published, and new ones continue to appear (Jain, 1999). There are different clustering algorithms and the following are a few representative.

Parametric mixture models are well known in statistics and machine learning communities (McLachlan, 1987). A mixture of parametric distributions, in particular, GMM (McLachlan, 2000) has been extensively used for clustering. GMMs are limited by the assumption that each component is homogeneous, unimodal, and generated using a Gaussian density. Latent Dirichlet Allocation (Blei, 2003) is a multinomial mixture model that has become the de facto standard for text clustering.

Several mixture models have been extended to their non-parametric form by taking the number of components to infinity in the limit (Teh, 2006). A non-parametric prior is used in the generative process of these infinite models (e.g. Dirichlet Process) for clustering in (Teh, 2006). One of the key advantages offered by the non-parametric prior based approaches is that they adjust their complexity to fit the data by choosing the appropriate number of parametric components. Hierarchical Topic Models (Blei, 2004) are clustering approaches that have seen huge success in clustering text data.

Kernel K-means is a related kernel based algorithm, which generalizes the Euclidean distance based K-means to arbitrary metrics in the feature space. Using the kernel trick, the data is first mapped into a higher dimensional space using a possibly non-linear map, and a K-means clustering is performed in the higher dimensional space.

Non-parametric density based methods are popular in the data mining community.

Mean-shift clustering (Comaniciu, 2002) is a widely used non-parametric density based clustering algorithm. The objective of Mean-shift is to identify the modes in the kernel-density, seeking the nearest mode for each point in the input space. Several density based methods like DBSCAN also rely on empirical probability estimates, but their performance degrades heavily when the data is high dimensional. A recent segmentation algorithm (Andreetto, 2007) uses a hybrid mixture model, where each mixture component is a convex combination of a parametric and non-parametric density estimates.

Hierarchical clustering algorithms are popular non-parametric algorithms that iteratively build a cluster tree from a given pairwise similarity matrix. Agglomerative algorithms such as Single Link, Complete Link, Average Link (Jain 1988), Bayesian Hierarchical Clustering (Heller, 2005), start with each data point in a single cluster, and merge them successively into larger clusters based on different similarity criteria at each iteration. Divisive algorithms start with a single cluster, and successively divide the clusters at each iteration.

2.2.2.1. Unsupervised Techniques for Document Clustering

In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters. Traditional clustering techniques can be categorized into two major groups as partitional and hierarchical.

Partitional Clustering Techniques

Partitional algorithms produce un-nested, non-overlapping partitions of documents that usually locally optimize a clustering criterion. The general methodology is as follows: given the number of clusters k, an initial partition is constructed; next the clustering solution is refined iteratively by moving documents from one cluster to another.

The following sub-sections discusses the most popular partitional algorithm k-means, and its variant bisecting k-means which has been applied to cluster documents by Steinbach et al. (Steinbach, 1999) and has been shown to generally outperform agglomerative hierarchical algorithms.

Expectation maximization

The theoretical basis for expectation-maximization shows that with sufficiently large amounts of unlabeled data generated by the model class in question, a more probable model can be found than if using just the labeled data alone. If the classification task is to predict the latent variable of the generative model, then with sufficient data a more probable model will also result in a more accurate classifier. Here, expectation-maximization finds more likely models and improved classification accuracy. Expectation-maximization (EM) is used to fit the mixture model to the negative examples(Olivier, 2006).
It uses the k-medoids which is similar to K-means except it takes one member of the cluster as a centriod. It uses real contexts word from the dataset as a basis for clustering. K-mean takes the round space (not a member) to make centriod

K-Means Clustering

The idea behind the k-means algorithm, discussed by Hartigan (Hartigan, 1975), is that each of k clusters can be represented by the mean of the documents assigned to that cluster, which is called the centroid of that cluster.

K-means (Cohen, 1996), (Yang, 1999), arguably, is the most popular and widely used clustering algorithm. K-means is an example of a sum of squared error (SSE) minimization algorithm. Each cluster is represented by its centroid. The goal of K-means is to find the centroids and the cluster labels for the data points such that the sum-of-squared error between each data point and its

closest centroid is minimized. K-means is initialized with a set of random cluster centers, that are iteratively updated by assigning the closest data point to each center, and recomputing the centroids. ISODATA (Hartigan, J., 1975) and Linear Vector Quantization (Berkhin, 2002) are closely related SSE minimization algorithms that are independently proposed in different disciplines.

It is discussed by Berkhin (Berkhin, 2002) that there are two versions of k-means algorithm known. The first version is the batch version and is also known as Forgy's algorithm (Forgy, 1965). It consists of the following two-step major iterations:

(1) Reassign all the documents to their nearest centroids

(2) Recompute centroids of newly assembled groups

Before the iterations start, firstly k documents are selected as the initial centroids.

Iterations continue until a stopping criterion such as no reassignments occur is achieved.

Initially, k documents from the corpus are selected randomly as the initial centroids. Then, iteratively documents are assigned to their nearest centroid and centroids are updated incrementally, i.e., after each assignment of a document to its nearest centroid. Iterations stop, when no reassignments of documents occur.

The centroid vector c of cluster C of documents is define as follows(Arzucan, 2002):

$$\mathbf{c} = \frac{\sum_{d \in C} \mathbf{d}}{|C|}$$

……equation 8

So, c is obtained by averaging the weights of the terms of the documents in C. Analogously, the similarity between a document d and a centroid vector c by cosine similarity measure defined as (Arzucan, 2002)

$$cos(\mathbf{d}, \mathbf{c}) = \frac{\mathbf{d} \bullet \mathbf{c}}{\|\mathbf{d}\| \|\mathbf{c}\|}$$

……equation 9

Note that although documents are of unit length, centroid vectors are not necessarily of unit length.

Bisecting K-Means

Although bisecting k-means is actually a divisive clustering algorithm that achieves a hierarchy of clusters by repeatedly applying the basic k-means algorithm, the researcher discuss it in this section as it is a variant of k-means.

In each step of bisecting k-means a cluster is selected to be split and it is split into two by applying basic k-means for k = 2. The largest cluster, that is the cluster containing the maximum number of documents, or the cluster with the least overall similarity can be chosen to be split.

Hierarchical Clustering Techniques

Hierarchical clustering algorithms produce a cluster hierarchy named a dendrogram (Berkhin, 2002). These algorithms can be categorized as divisive (top-down) and agglomerative (bottom-up) (Jain, 1999) (Berkhin, 2002). We discuss these approaches in the following sub-sections.

Divisive Hierarchical Clustering

Divisive algorithms start with one cluster of all documents and at each iteration split the most appropriate cluster until a stopping criterion such as a requested number k of clusters is achieved.

A method to implement a divisive hierarchical algorithm is described by Kaufman and Rousseeuw. In this technique in each step the cluster with the largest diameter is split, i.e. the cluster containing the most distant pair of documents. As we use document similarity instead of distance as a proximity measure, the cluster to be split is the one containing the least similar pair of documents. Within this cluster the document with the least average similarity to the other documents is removed to form a new singleton cluster. The algorithm proceeds by iteratively assigning the documents in the cluster being split to the new cluster if they have greater average similarity to the documents in the new cluster (Kang, 2003).

Agglomerative Hierarchical Clustering

Agglomerative clustering algorithms start with each document in a separate cluster and at each iteration merge the most similar clusters until the stopping criterion is met. They are mainly

categorized as single-link, complete-link and average-link depending on the method they define inter-cluster similarity.

Single-link The single-link method defines the similarity of two clusters C_i and C_j as the similarity of the two most similar documents $d_i \in C_i$ and $d_j \in C_j$ (Arzucan, 2002):

$$similarity_{single-link}(C_i, C_j) = \max_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})|$$

Complete-link The complete-link method defines the similarity of two clusters

C_i and C_j as the similarity of the two least similar documents $d_i \in C_i$ and $d_j \in C_j$ (Arzucan, 2002):

$$similarity_{complete-link}(C_i, C_j) = \min_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})| \quad \text{.....equation 10}$$

Average-link The average-link method defines the similarity of two clusters

C_i and C_j as the average of the pairwise similarities of the documents from each cluster (Arzucan, 2002):

$$similarity_{average-link}(C_i, C_j) = \frac{\sum_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})|}{n_i n_j} \quad \text{.....equation 11}$$

where n_i and n_j are sizes of clusters C_i and C_j respectively.

2.2.3. Semi-supervised learning

Semi-supervised learning algorithms can be broadly classified based on the role the available side information plays in providing the solution to supervised or unsupervised learning.

2.2.3.1. Semi-supervised classification

While semi-supervised classification is a relatively new research area, the idea of using unlabeled samples to augment labeled examples for prediction was conceived several decades ago.

The initial work in semi-supervised learning is attributed to Scudders for his work on "selflearning". An earlier work by Robbins and Monro on sequential learning can also be viewed

as related to semi-supervised learning. Vapnik's Overall Risk Minimization (ORM) principle advocates minimizing the risk over the labeled training data as well as the unlabelled data, as opposed to the Empirical Risk Minimization, and resulted in transductive Support Vector Machines (Pavan, 2010).

Given a set of labeled data, a decision boundary may be learned using any of the supervised learning methods. When a large number of unlabeled data is provided in addition to the labeled data, the true structure of each class is revealed through the distribution of the unlabeled data. The unlabeled data defines a "natural region" for each class, and the region is labeled by the labeled data. The task now is no longer just limited to separating the labeled data, but to separate the regions to which the labeled data belong. The definition of this "region" constitutes some of the fundamental assumptions in semi-supervised learning (Kang 2003).

Existing semi-supervised classification algorithms may be classified into two categories based on their underlying assumptions. An algorithm is said to satisfy the manifold assumption if it utilizes the fact that the data lie on a low-dimensional manifold in the input space. Usually, the underlying geometry of the data is captured by representing the data as a graph, with samples as the vertices, and the pairwise similarities between the samples as edge-weights. Several graph based algorithms such as Label propagation, Markov random walks, Graph cut algorithms, Spectral graph transducer, and Low density separation are based on this assumption. The second assumption is called the cluster assumption. It states that the data samples with high similarity between them, must share the same label. This may be equivalently expressed as a condition that the decision boundary between the classes must pass through low density regions. This assumption allows the unlabeled data to regularize the decision boundary, which in turn influences the choice of the classification models. Many successful semi-supervised algorithms like TSVM and Semi-supervised SVM follow this approach (Pavan, 2010). These algorithms assume a model for the decision boundary, resulting in an inductive classifier.

Bootstrapping Classifiers from Unlabeled data

One of the first uses of unlabeled data was to bootstrap an existing supervised learner using unlabeled data iteratively. The unlabeled data is labeled using a supervised learner trained on the labeled data, and the training set is augmented by the most confident labeled samples.

This process is repeated until all the unlabeled data have been processed. This is popularly known as "Self-training", which was first proposed by Scudders (Scudder, 1965). Yarowsky (Yarowsky, 1995) applied self-learning to the "word sense" disambiguation problem. Rosenberg et al. (Rosenberg, 2005) applied self-training for object detection.

Several classifiers proposed later follow the bootstrapping architecture similar to that of self-training, but with a more robust and well-guided selection procedure for the unlabeled samples for inclusion in the training data. Semi-supervised generative models using EM (Dempster, 1977), for instance, the Semi-supervised Naive Bayes (Nigam, 2000), is a "soft" version of self-training. Many ensemble classification methods, in particular, those following the semi-supervised boosting approach (Bennet, 2002), (Mallapragada, 2009) use specific selection procedures for the unlabeled data, and use a weighted combination of classifiers instead of choosing the final classifier.

Margin based classifiers

The success of margin based methods in supervised classification motivated a significant amount of research in their extension to semi-supervised learning. The key idea of margin based semi-supervised classifiers is to model the change in the definition of margin in the presence of unlabeled data. Margin based classifiers are usually extensions of Support Vector Machines (SVM). An SVM minimizes the empirical error on the training set, along with a regularization term that attempts to select the classifier with maximum margin.

Vapnik (Yang, 1999) first formulated this problem and proposed a branch and bound algorithm.

A Mixed Integer Programming based solution is presented in (Mallapragada, 2009), which is called Semi-supervised SVM or S^3VM. Fung and Mangasarian (Pavan, 2010) proposed a successive linear approximation to the min (.) function in the loss function, and proposed VS^3VM. None of these methods are applicable to real datasets (even small size datasets) owing to their high computational complexity.

Transductive SVM (TSVM) is one of the early attempts to develop a practically usable algorithm for semi-supervised SVM. TSVM provides an approximate solution to the combinatorial optimization problem of semi-supervised SVM by first labeling the unlabeled data with an SVM

trained on the labeled data, followed by switching the individual labels of unlabeled data such that the objective function is minimized. Gradient descent was used in (Pavan, 2010) to minimize the same objective function, while defining an appropriate subgradient for the min(.) function. This approach was called ∇TSVM, and its performance is shown to be comparable to that of the other optimization schemes discussed above.

Graph Connectivity

Graph theory has been known to be powerful tool for modeling unsupervised learning (clustering) problems since its inception to relatively recent Normalized Cuts and Spectral clustering (Ng, 2002), and shown to perform well in practice (Brandes, 2003). Graph based methods represent the data as a weighted graph, where the nodes in the graph represent the data points, and the edge weights represent the similarity between the corresponding pair of data points. The success of graph based algorithms in unsupervised learning motivates its use in semi-supervised learning (SSL) problems.

The edge weight between a pair of samples is set to ∞ if they share the same label, to ensure that they remain in the same partition after partitioning the graph. Szummer and Jakkola (Szummer , 2001) and Zhu and Ghaharamani (Zhu, 2002) model the graph as a discrete Markov random field, where the normalized weight of each edge represents the probability of a label (state) jumping from one data point to the other. The solution is modeled as the probability of a label (from a labeled data point) reaching an unlabeled data point in a finite number of steps. Zhu et al., (Zhu, 2003) relax the Markov random field with a discrete state space (labels) to a Gaussian random field with continuous state space, thereby achieving an approximate solution with lower computational requirements.

Most graph based semi-supervised learning methods are non-parametric and transductive in nature, and can be shown as solutions to the discrete Green's function, defined using the discrete Graph Laplacian (Yang, 1999).

2.2.3.2. Semi-supervised clustering

Clustering aims to identify groups of data such that the points within each group are more similar to each other than the points between different groups. Clustering problem is ill-posed, and hence multiple solutions exist that can be considered equally valid and acceptable. Semi-supervised

clustering utilizes any additional information, called side information, which is available to disambiguate between the solutions. The side information is usually present in the form of instance level pairwise constraints (Wagstaff , 2000). Pair wise constraints are of two types – must-link constraints and cannot-link constraints. Given a pair of points, must link constraints require the clustering algorithm to assign the same label to the points. On the other hand, cannot-link constraints require the clustering algorithm to assign different labels to the points.

Penalizing Constraints

One of the earliest constrained clustering algorithms was developed by Wagstaff and Cardie (Wagstaff, 2000), (Wagstaff, 2001), called the COP K-means algorithm. The cluster assignment step of Kmeans algorithm was modified with an additional check for constraint violations. However, when constraints are noisy or inconsistent, it is possible that there are some points that are not assigned to any cluster. This was mitigated in an approach by Basu et. al. (Basu, 2004) which penalizes constraint violations instead of imposing them in a hard manner. A constrained clustering problem is modeled using a Hidden Markov Random Field (HMRF) which is defined over the data and the labels, with labels as the hidden states that generate the data points. The constraints are imposed on the values of the hidden states. Inference is carried out by an algorithm similar to that of K-means which penalizes the constraint violations.

Generative models are very popular in clustering. Gaussian mixture model (GMM) is one of the well-known models used for clustering (Dempster, 1977), (Figueiredo, 2002). Shental et al. (Shental, 2004) incorporated pairwise constraints into the GMMs. To achieve this, groups of points connected by must-link constraints are defined as chunklets and each chunklet is treated as a single point for clustering purposes. Zhao and Miller (Zhao, 2005) proposed an extension to GMM which penalizes constraint violations. A method to automatically estimate the number of clusters in the data using the constraint information was proposed. Lu and Leen (Lu, 2005) incorporate the constraints into the prior over all possible clustering.

In many approaches that enforce constraints in a hard manner (including those that penalize them), non-smooth solutions are obtained. A solution is called non-smooth when a data point takes a cluster label that is different from all of its surrounding neighbors. As noted in (Law, 2005), it is possible that the hypothesis that fits the constraints well may not fit the data well.

Therefore, a tradeoff between satisfying the constraints and fit to the data is required. Lange et al. (Lange, 2005) alleviate this problem by involving all the data points into a constraint through a smooth label.

Adapting the Similarity

Several semi-supervised clustering methods operate by directly modifying the entries of the pair-wise similarity matrix that are involved in constraints. All these algorithms reduce the distance between data points connected by must-link constraints and increase the distance between those connected by must-not link by a small value. Spectral Learning algorithm by Kamvar et al. (Kamvar, 2003) modifies the normalized affinity matrix by replacing the values corresponding to must-link constraints by 1 and must-not link constraints by 0. The specific normalization they use ensures that the resulting matrix is positive definite. The remaining steps of the algorithm are the same as the Spectral clustering algorithm by Ng et al. (Ng, 2002). Klien et. al. (Klien, 2002) modified the dissimilarity metric by replacing the entries participating in must-link constraints with 0 and replaced the entries participating in cannot-link constraints by maximum pairwise distance incremented by 1. This is followed by a complete link clustering on the modified similarity matrix. Kulis et al. (Kulis, 2007) propose a generalization of Spectral Learning via semi-supervised extensions to the popular normalized cut (Shi, 2000), ratio cut and ratio association (Hagen,1992). To ensure positive definiteness of the similarity matrix, they simply add an arbitrary positive quantity to the diagonal.

The specific values of increments had chosen in the above algorithms impacts the performance of the clustering algorithm. In order to apply spectral algorithms, we need the pairwise similarity matrix to be positive semi-definite. Arbitrary changes (especially decrements) to the similarity matrix may not retain its positive semi-definiteness. Some methods avoid using spectral algorithms, while some update the similarity matrix carefully to retain the essential properties. The similarity adaptation methods are adhoc in nature, and are superseded by the similarity learning approaches presented in the next section.

Learning the Similarity

The performance of a clustering algorithm depends primarily on the similarity metric defined between the samples. It is usually difficult to design a similarity metric that suits all the clustering scenarios. For this reason, attempts have been made to directly learn the similarity metric from the data using the side information. Similarity metric learning is not a new problem, and has been considered before in both unsupervised dimensionality reduction methods (LLE (Roweis , 2000), ISOMAP (Silva, 2003) and supervised methods like Fisher Linear Discriminate (Cohen, 1996), Large Margin Distance Metric Learning (Weinberger,2006) and Neighborhood Component Analysis (Goldberger, 2005). Only those methods that learn the distance metric in a semi-supervised setting, i.e., using pairwise constraints and unlabeled data are reviewed here.

Once a similarity metric is learned, standard classification algorithms may later be applied with the learned similarity metric. The distance metric learning problem can be posed in its generality as follows: learn a function $f : X \times X \rightarrow R$ such that the distance between points linked by must-link constraints is smaller than that between the points linked by must-not link constraints overall. The distance function is usually parametrized in its quadratic form, i.e $fA(xi, xj) = xTi Axj$, where A is the unknown parameter to be estimated from the constraints.

Xing et al. (kang, 2003) formulated distance metric learning as a constrained optimization problem, where A is estimated such that the sum of distances between points connected by must-link constraints is minimized, while constraining the sum of distances between points connected by must-not link to be greater than a fixed constant. Bar-Hillel et al. (Bar-Hillel, 2005) proposed Relevant Component Analysis (RCA), which estimates a global transformation of the feature space by reducing the weights of irrelevant features such that the groups of data points linked by must-link constraints (called chunklets) are closer to each other. A modified version of the constrained K-means algorithm that learns a parametrized distance function is presented in (Bilenko, 2004).

Yang et al. (Yang, 2006) learn a local distance metric by using an alternating optimization scheme that iteratively selects the local constraints, and fits the distance metric to the constraints.

They parametrize the kernel similarity matrix in terms of the eigenvalues of the top few eigenvectors of the pairwise similarity matrix computed using the RBF kernel. Hoi et al. (Hoi,

2007) present a non-parametric distance metric learning algorithm that addresses the limitations of quadratic distance functions used by almost all the other approaches. Lee et al. (Lee, Jin, 2008) proposed an efficient distance metric learning algorithm and applied it to a content based image retrieval task showing significant performance gains.

There has been a recent surge in the interest in online learning algorithms due to the large volume of datasets that need to be processed. Shalev-shwartz et al. (Shalev-Shwartz, 2004) present an online distance metric learning algorithm called POLA, that learns a quadratic distance function (parametrized by the covariance matrix) from pairwise constraints. A batch version of the algorithm is obtained by multiple epochs of the online algorithm on the training data. Davis et al. (Davis, 2007) present online and batch versions of an algorithm that searches for the parameterized covariance matrix A that satisfes the constraints maximally. Additionally, a log-determinant regularizer is added to prevent A from moving too far away from the initial similarity metric A0.

2.3. Document Preprocessing and Representation

In order to cluster or classify text documents by applying machine learning techniques, documents should first be preprocessed. In the preprocessing step, the documents should be transformed into a representation suitable for applying the learning algorithms. The most widely used method for document representation is the vector space model introduced by Salton et. al. In this model, each document is represented as a vector d. Each dimension in the vector d stands for a distinct term in the term space of the document collection (Arzucan, 2002).

A term in the document collection can stand for a distinct single-word, a stemmed word or a phrase. Phrases consist of multiple words such as “data mining” or “mobile phone” and constitute a different context than when used separately. Phrases can be extracted by using statistical or Natural Language Processing (NLP) techniques. By statistical methods phrases can be extracted by considering the frequently appearing sequences of words in the document collection (Cohen, 1996). A research on extracting phrases by using NLP techniques for text categorization is discussed by Fuernkranz et al. (Fuernkranz, 1998).

In vector space representation, defining terms as distinct single words is referred to as “bag of words” representation. Some researchers state that using phrases rather than single words to

define terms produce more accurate classification results (Cohen, 1996); whereas others argue that using single words as terms does not produce worse results (Dumais, 1998)(Sahami, M., 1998). As “bag of words” representation is the most frequently used method for defining terms and it is computationally more efficient than the phrase representation.

One challenge emerging when terms are defined as single words is that the feature space becomes very high dimensional. In addition, words which are in the same context such as biology and biologist are defined as different terms. So, in order to define words that are in the same context with the same term and consequently to reduce dimensionality the researcher have decided to define the terms as stemmed words. To stem the words, the researcher has chosen to use Porter’s Stemming Algorithm (Porter, 1980), which is the most commonly used algorithm for word stemming in English.

Preprocessing and document representation phase, which is implemented in python, consists of the following steps:

- Tokenization
- Removing stop words
- Stemming
- Term weighting
- Dimensionality reduction

These steps will be described briefly in the following sections.

Tokenization

Tokenization is the process of breaking down of documents into individual tokens.

In the tokenization process irrelevant and noisy features for the classification process such as punctuation marks and any irrelevant characters removed from documents in the collection. This is because these features are not relevant to represent the content of documents and they have no contribution in discriminating one document or category from the other. (Arzucan, 2002).

Removing Stop words

There are words, such as pronouns, prepositions and conjunctions that are used to provide structure in the language rather than content. These words, which are encountered very frequently and carry no useful information about the content and thus the category of documents, are called stopwords. Removing stopwords from the documents is very common in information retrieval. In this paper stope words are eliminated from the documents, which will lead to a drastic reduction in the dimensionality of the feature space(Arzucan, 2002).

Stemming

In order to define words that are in the same context with the same term and consequently to reduce dimensionality. Porter's stemming Algorithm which is the most commonly used algorithm for word stemming in English. For instance, we reduce the similar terms computer", "computers", and "computing" to the word stem "compute". Implementation of Porter's Stemming Algorithm in python is developed. This algorithm is embedded to the preprocessing system.

After stemming, terms that are shorter than two characters are also removed as they do not carry much information about the content of a document.

Term Weighting

We represent each document vector d as

d=(w1, w2,....., wn)

Where wi is the weight of i^{th} term of document d. There are various term weighting approaches most of which are based on the following observations (Arzucan, 2002):

- ✓ The relevance of a word to the topic of a document is proportional to the number of times it appears in the document.
- ✓ The discriminating power of a word between documents is less, if it appears in most of the documents in the document collection.

A comparative study of different term weighting approaches in automatic text retrieval is presented by Salton and Buckley (Arzucan, 2002). The term weighting approach applied in the

study and some other standard term weighting functions are discussed in the following subsections. In the study terms are defined as follows:

tfi as the raw frequency of term i in document d;

N as the total number of documents in the document corpus;

ni as the number of documents in the corpus where term i appears; and M as the number of terms in the document collection (after stopword removal and stemming is performed).

Boolean Weighting

Boolean weighting is the simplest method for term weighting. In this approach, the weight of a term is assigned to be 1 if the term appears in the document and it is assigned to be 0 if the term does not appear in the document(Arzucan, 2002).

$$w_i = \begin{cases} 1 & \text{if } tf_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Term Frequency (TF) Weighting

Term frequency weighting is also a simple method for term weighting. In this method, the weight of a term in a document is equal to the number of times the term appears in the document, i.e. to the raw frequency of the term in the document (Arzucan, 2002).

wi = tfi

Term Frequency * Inverse Document Frequency (TF*IDF) Weighting

Boolean weighting and term frequency weighting do not consider the frequency of the term throughout all the documents in the document corpus. TF*IDF weighting is the most common method used for term weighting that takes into account this property.

In this approach, the weight of term i in document d is assigned proportionally to the number of times the term appears in the document, and in inverse proportion to the number of documents in the corpus in which the term appears (Arzucan, 2002).

wi = tfi * log(N/ni) ……equation 12

TF*IDF weighting approach weights the frequency of a term in a document with a factor that discounts its importance if it appears in most of the documents, as in this case the term is assumed to have little discriminating power.

TF£IDF Weighting With Length Normalization

In this approach, to account for documents of different lengths each document vector is normalized so that it is of unit length.

$wi = tf_i * \log(N/n_i)$ ……equation 13

Salton and Buckley discuss that TF£IDF weighting with length normalization generally performs better than the other techniques (Arzucan, 2002). Therefore, we applied this weighting approach in our study.

Dimensionality Reduction

There are various methods applied for dimensionality reduction in document categorization.

Some common examples are Information Gain (IG), Mutual Information (MI), Chi-Square Statistic, Term Strength (TS), and Document Frequency (DF) Thresholding. The study discuss these techniques briefly in the following subsections.

Information Gain (IG)

Information gain measures the number of bits of information gained for category prediction when the presence or absence of a term in a document is known. When the set of possible categories is fc1; c2; …; cmg, the IG for each unique term t is calculated as follows (Joachims):

$$IG(t) = -\sum_{i=1}^{m} P(c_i)\cdot \log P(c_i) + P(t)\cdot \sum_{i=1}^{m} P(c_i|t)\cdot \log P(c_i|t) + P(\bar{t})\cdot \sum_{i=1}^{m} P(c_i|\bar{t})\cdot \log P(c_i|\bar{t})$$ ..equation 14

As seen from Equation, IG calculates the decrease in entropy when the feature is given vs. absent. $P(c_i)$ is the prior probability of category c_i. It can be estimated from the fraction of documents in the training set belonging to category ci. P(t) is the prior probability of term t. It can be estimated from the fraction of documents in the training set in which term t is present. Likewise, P(t) can be estimated from the fraction of documents in the training set in which term t

is absent. Terms whose IGs are less than some predetermined threshold are removed from the feature space.

Mutual Information (MI)

Mutual information is a technique frequently used in statistical language modeling of word associations and related applications. MI between term t and category c is defined to be (Arzucan, 2002):

$$MI(t,c) = \log \frac{P(t \wedge c)}{P(t) \times P(c)} \quad \text{.....equation 15}$$

It is estimated by using (Arzucan, 2002):

$$MI(t,c) \approx \log \frac{A \times N}{(A+R) \times (A+B)} \quad \text{.....equation 16}$$

Here, A is the number of times t and c co-occur, B is the number of times t occurs without c, R is the number of times c occurs without t, and N is the total number of documents. When t and c are independent MI (t; c) is equal to zero.

We can write equation in the following equivalent form:

$$MI(t,c) = \log P(t|c) - \log P(t) \quad \text{.....equation 17}$$

It is seen from equation for terms that have an equal conditional probability, rare terms will have a higher MI value than common terms. So, MI technique has the drawback that MI values are not comparable among terms with large frequency gaps.

Category specific MI scores for a term t can be combined into a global MI score for that term in the following two ways(Arzucan, 2002):

$$MI_{avg}(t) = \sum_{i=1}^{m} P(c_i) \times MI(t,c_i) \quad \text{.....equation 18}$$

or

$$MI_{max}(t) = \max_{i=1}^{m}\{MI(t, c_i)\} \quad (2.10)$$ ……equation 19

Terms that have lower MI values than a predetermined threshold are eliminated.

Term Strength (TS)

Term strength method, estimates term importance based on how commonly a term is likely to appear in closely related documents (Yang, 1999). The first step in this method is to use a training set of documents to find document pairs which have a similarity larger than a predetermined threshold. In the next step TS is calculated based on the estimated conditional probability that a term appears in the second document given that it appears in the first one. Suppose, x and y are any pair of distinct but related documents. Then the TS of term t is defined to be (Yang, 1999):

$$TS(t) = P(t \in y | t \in x)$$ ……equation 20

Unlike IG, MI, and X^2 statistic, TS is an unsupervised dimensionality reduction technique where document categories are not used. It is based on document clustering and assumes that documents with many shared words are related and the terms that are heavily shared among these related documents are relatively informative.

Document Frequency Thresholding (DF)

Document frequency (DF) of a term is the number of documents that term appears. In this technique, the document frequency of each unique term is computed and terms whose document frequencies are less than a predetermined threshold are eliminated. The basic assumption behind this technique is that rare terms are either non-informative for document categorization or they do not have much weight in global performance. This technique can also lead to improvement in categorization accuracy in case rare terms are noise terms. However, DF is usually not used for aggressive term elimination because there is another widely accepted assumption in information retrieval that low-DF terms are distinctive and thus relatively informative and for this reason should not be removed aggressively (Yang, 1999).

A comparative study of feature selection in text categorization is presented by Yang and Pedersen (Yang, Y). It has been reported that IG and X^2 statistic performed the best. However,

DF, the simplest and most efficient method in terms of computational complexity, performed similar to IG and X^2 statistics. It has been suggested that DF can be reliably used instead of IG and X^2 statistics when computation performances of the latter two are too expensive.

Another point to consider is that IG, MI and X^2 statistics are supervised techniques and use information about term-category associations. As our main focus is on unsupervised techniques for document organization, these methods are not suitable to be applied in our study. To reduce the dimensionality of the data, we apply DF Thresh holding. We define the document frequency threshold as 1 and hence remove the terms that appear in only one document.

Document Similarity Measure

To use a clustering or classification algorithm, a similarity measure between two documents must be defined. Cosine similarity measure is the most widely used similarity measure to calculate the similarity of two documents. This measure is defined as (Steinbach, 1999):

$$cos(\mathbf{d_1}, \mathbf{d_2}) = \frac{\mathbf{d_1} \bullet \mathbf{d_2}}{\|\mathbf{d_1}\|\|\mathbf{d_2}\|}$$ ……equation 21

That is, it is the dot product of d1 and d2 divided by the lengths of d1 and d2.

Related works

There are a lot of works done on classification of Amharic document. From these the following works are included:

Zelalem Sentayehu has worked on supervised Amharic news text classification in 2001. The overall result of his research has showed that statical technique can be used to analyze Amharic news items and classify automatically in to predefind classes. After training the classifier classified 273 out of 321 news items correctly (Zelalem, 2001).

Surafel Teklu has worked on supervised Amharic news text classification in 2003. The objective of the researcher 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 (Surafel, 2003).

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 (Surafel, 2003).

Yohannes has also worked on supervised Amharic text classification in 2007. Because of the high dimensionality of the source data, classifier algorithms that are suitable for high-dimensional data the researcher used, Decision Tree and Support Vector Machine (SVM) for the

research experiment. The researcher also used the open source Weka package for the automatic classification of the preprocessed data. Out of the many classifier algorithms available in Weka, the Logic Model Tree (LMT) and the Library of SVM (LibSVM) classifiers were used for performance testing (Yohannes, 2007).

Both LMT and LibSVM classifier showed good classification accuracy correctly classifying 79.72% and 81.15% of the test instance into the 15 news categories, respectively. However, the computational cost of the automatic classification was very high - taking several hours in high capacity computers. The classification performance measures indicate the need for additional works in developing tools and methods for mining Amharic data. (Yohannes, 2007)

Lakechew yayeh has done on unsupervised Amharic text news classification in 2011 and the researcher used k-means, bisecting k-means and average link clustering algorithms. The performances of document clustering algorithms: k-means, bisecting k-means and average link were compared for the 4, 7 and 10 clustering solutions using entropy, purity and overall similarity evaluation metrics over the different pre-defined data sets. The performances of k-means and bisecting k-means are similar in terms of the overall similarity measure in all number of clusters and they produced similar clustering solutions. However, the results of the findings indicate that the bisecting k-means produced better clustering solutions consistently according to the entropy and purity evaluation measures.

The results also shows that k-means and bisecting k-means clustering algorithms consistently produced clusters that are most similar to pre-defined classes at different data sets. Moreover, both k-means and bisecting k-means clustering algorithms produced clusters relatively with similar cluster size (number of documents), while the agglomerative hierarchical clustering algorithms generally produced clusters that are not similar to pre-defined classes and clusters with unbalanced cluster size (number of documents).

Agglomerative hierarchical clustering algorithms produced low quality results as compared to the k-means and the bisecting k-means clustering algorithms. Among the agglomerative clustering algorithms, the average link achieved the best performance as compared to single link and complete link in all evaluation measures.

In this study, the potential application of unsupervised Learning techniques for the classification of Amharic text documents was explored.

The effect of the number of clusters and the size of documents used on the performance and efficiency of clustering algorithms was tested and compared using different data sets.

Moreover, the performances of these clustering algorithms were also tested at increasing number of clusters using the same data set. The agreement between the number of predefined classes and the number of clusters discovered by the agglomerative clustering algorithm was also tested for 10 clusters over the whole document collection.

Based on the experiments done in this thesis, the following concluding remarks were made.

As the number of clusters and documents increase, the clustering solutions produced by k-means and bisecting k-means become more internally cohesive and externally isolated. However, the clustering results do not match better with the pre-defined classes and requires relatively high computational requirements.

Moreover; the purity values of single link, complete link and average link decrease.

According to the results obtained, it was difficult to determine the entropy and the overall similarity values of the three agglomerative approaches at increasing number of clusters and documents.

All the clustering algorithms: k-means, bisecting k-means, single link, complete link and average link achieved better clustering quality as the number of clusters increases with the same data set. The clustering solutions became more internally cohesive, externally isolated and match better with the pre-defined classes (Lakechew, 2011)

CHAPTER THREE

THE AMHARIC LANGUAGE AND ITS WRITING SYSTEM

3.1. The Amharic Language

The name Amharic (አማርኛ - amarəññña) comes from the district of Amhara (አማራ) in northern Ethiopia, which is thought to be the historic centre of the language. Amharic is a Semitic language and the national language of Ethiopia (ኢትዮጵያ). The majority of the 25 million or so speakers of Amharic can be found in Ethiopia, but there are also speakers in a number of other countries, particularly Eritrea (ኤርትራ), Canada, the USA and Sweden.Amharic is the working language of the Federal Government of Ethiopia and is spoken and written as a first or second language in many parts of the country (Yohannes, 2007).

Amharic, like other languages that use the Ethiopic script (Gurage, Harari, Tigre, and Tigrniya), use characters derived mainly from Geez.

The Ethiopic script was first displayed on a computer around 1986. At the time the challenge in the computer representation of the script was developing a software package that can handle character design, keyboard layout and printer set-up. The work by ESTC started an enthusiastic rush to develop Ethiopic software by different IT companies and teams of individuals which led to the problem of lack of standardization. Now a day there are more than 35 Ethiopic software products available, each with its own character set, encoding system, typeface names and keyboard layout.

The recent development of the introduction of the Ethiopic range with the Unicode standard could help in standardizing the different incompatible software products.

3.2. The Amharic writing system

The writing system of Amharic is taken from Geez (Bender, 1976; Aklilu, 1984) that in turn evolved out of Sabaean Language the descendent of South Semitic Script. It was brought to highlands of Ethiopia by immigrants from South Arabia in the first century A.D (Bender, 1976).

Geez, which remained the ecclesiastical and literary expression in Ethiopia until the 16th century, gradually gave way to Amharic that was used both in spoken and writing in the royal courts. It began to be used for literary purposes at the beginning of the 19th century as the administrative state changed its way of communication from oral to written one(Surafel, 2003).

Up to 350 A.D Geez scripts have no vowel indications. Latter, however, vocalized consonant signs had come into being by undergoing a variety of changes in the structure of the consonantal symbols. The structural changes added six additional forms to each basic consonant increasing the total number of symbols to 182(26x7). Since then, vowels became an integral part of Ethiopic writing (Surafel, 2003).

By the time Geez was replaced by Amharic, in addition to the 26 symbols that were used in the Geez language, it added symbols by deriving them from the already existing Geez alphabets.

ሸ From ሰ

ቸ From ተ

ኘ From ነ

ዠ From ዘ

ጀ From ደ

ጨ From ጠ

ኸ From ከ

This increased the total number of fundamental characters used in Amharic writing system to 34; out of which 33 are core characters and 1 is a special character (Million, 2000).

3.3. The Amharic Characters (ፊደል)

In Amharic writing system there are a total of 231 characters, 33 of the characters are the ‘core’ characters and one is ‘special’ character. Each character has seven different forms called orders that reflect the seven vowel sounds (e, u, i, a, e, i, o); one basic form and six non – basic forms representing syllable combinations consisting of a consonant and vowel. It is shown in appendix 1

There exists other character in addition to the 231 core characters that are indicated in appendix 3. The syllables with the vowel transliterated as (i) are pronounced (ə), except in final position when the vowel is not pronounced.

Characteristics of the Amharic Character

Amharic writing system is often called syllabary rather than an alphabet because the seven orders of Amharic characters indicated above represent syllable combination consisting of consonant and following vowel. The non basic forms (vocalization) are derived from the basic forms (consonants) by attaching small appendages (diacritic marks) to the right, left, top, or bottom in more or less regular modification. Some are formed by adding strokes, others by adding loops or other forms of differentiation to each core character. The writing system is difficult and vulnerable to various problems; it is difficult to automate information retrieval system for Amharic language. These writing problems have a negative effect on the performance of different machine learning approaches in text classification and text clustering. Some of the problems are discussed in the following sections.

Formation of Compound Nouns

Bender stated that compound nouns are sometimes written as two separate words (Bender, 1976). For example, ብርድ-ልብስ which means "blanket" may be written as ብርድ ልብስ or ብርድልብስ and; ክፍለከተማ as ክፍለ-ከተማ which means "sub city". This happened to be inconsistent in Amharic texts and should be considered in automatic classification (Surafel, 2003).

Character Redundancy

Out of 275 Amharic characters 231 are actually necessary to represent Amharic because the other characters are redundant, i.e., by using only one character from a group of characters with the same sound (Yohannes, 2007).

Spelling variations of a word would unnecessarily increase the number of words representing a document which could reduce the efficiency and accuracy. Amharic document processing for feature selection should therefore normalize word variants (spelling differences) caused by inconsistent usage of redundant characters.(Yohannes, 2007)

During the pre-processing stage of Amharic documents for this research, the different forms of a character that have the same sound are changed to one common form.

Consonants	Other symbols with the same sound
ሀ(hä)	ሐ፣ኀ፣ሃ፣ሓandኃ
ሰ(sä)	ሠ
አ(ä)	ዐ፣ኣ andዓ
ጸ(tsä)	ፀ

Table 1 shows a sample of redundant characters where more than one symbol is used for a given sound.

Inconsistency of Abbreviations

To write Amharic words in abbreviation people use different symbols. Forward slash ("/") and period (".") are the most common symbols used to write words in shorter form. For example the short form of the word ፍርድ ቤት can be written as "ፍ/ቤት", "ፍ.ቤት" or "ፍ-ቤት" which result in an inconsistency of abbreviating Amharic words. These different representations of the same word create high dimensional vector space and it has a negative effect on the performance of learning algorithms.(Lakechew, 2010)

Variations due to Pronunciations

The usage of foreign language words in Amharic is also found to be another source of word spelling variations. Most of the time different writers use different spellings in the writings of words adapted from foreign languages. This writing problem also has a negative effect on the performance of different machine learning approaches in text classification and text clustering

For example, the word ላብራቶሪ (laboratory) is found to have different Amharic spellings like ላቭራቶሪ፣ ላቦራቶሪ in the source data.(Yohannes, 2007)

Other Cases of Word Variations:

Usage of different affixing and suffixing style for same word causes word spelling variations. In most cases different writers use different affix and suffix spellings in the writings of words. For example difference in suffixing would result in the two writings ኢትዮጵያዊ and ኢትዮጵያአዊto refer

to human intellect while difference in prefixing would give the two writings ጥዳት and ፅዳት to mean ‘sanitary’ (Yohannes, 2007).

Punctuation

In Amharic language words are separated by two dots (: ሁለት ነጥብ), however, blank spaces are generally used. 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 from foreign languages such as (?, !, “ , ”, ‘, /, \, etc.). According to Beletu (Beletu, 1982) there are about 17 punctuation marks used in Amharic language. However, the existing Amharic software does not make use some of them. It is shown in appendix 4

Numerals

According to Bender et al., Amharic number 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. Numbering starts from one and has single characters for numbers one to ten, more than one character for multiples of ten (twenty to ninety), hundred, and thousand. There is no symbol for zero in the Amharic script. Ethiopic numbers are used mostly in writing dates and page numbers in text (Bender, 1976). Amharic number characters are indicated in appendix 2

3.4. Computerizing the Amharic Script

The ASCII code does not recognize Amharic scripts and thus cannot assign numeric codes to the scripts. For ease of preprocessing and compatibility reasons, the Amharic text was transliterated into an ASCII representation using SERA.

A SERA is a scheme for transliterating Amharic characters. The fundamentals of SERA are discussed in Daniel (1996). SERA is a convention for transliteration of Amharic characters (Fidel) script into Latin script that insures the integrity of the format and content of the original document, and that can be fully transportable across all computer mediums.

CHAPTER FOUR

4. METHODOLOGY

Introduction

The machine learning approach to text categorization is to automatically build the classifiers by learning the concept descriptions of the categories. One type of machine learning, applied to text categorization, is "supervised learning". This requires a set of pre-labeled (pre-categorized) training documents for generating classifiers. In contrast, "unsupervised learning" refers to the task of automatically identifying a set of categories from a set of unlabeled documents and grouping these unlabeled documents under these identified categories (Merkl , 1998). This task is typically called document clustering. Semi-supervised classification algorithms train a classifier given both labeled and unlabeled data. The goal is to label only the unlabeled data available during training

In order to cluster or classify text documents by applying machine learning techniques, documents should first be preprocessed. In the preprocessing step, the documents should be transformed into a representation suitable for applying the learning algorithms. The most widely used method for document representation is the vector space model introduced by Salton et. al.(Salton, 1975).

In this model, each document is represented as a vector d. Each dimension in the vector d stands for a distinct term in the term space of the document collection.

A term in the document collection can stand for a distinct single-word, a stemmed word or a phrase. Phrases consist of multiple words such as "data mining" or "mobile phone" and constitute a different context than when used separately. Phrases can be extracted by using statistical or Natural Language Processing (NLP) techniques. By statistical methods phrases can be extracted by considering the frequently appearing sequences of words in the document collection (Cohen, 1996). A research on extracting phrases by using NLP techniques for text categorization is discussed by Fuernkranz et al. (Fuernkranz, 1998).

In vector space representation, defining terms as distinct single words is referred to as "bag of words" representation. Some researchers state that using phrases rather than single words to define terms produce more accurate classification results; whereas others argue that using single words as terms does not produce worse results (Sahami, 1998). As "bag of words" representation

is the most frequently used method for defining terms and it is computationally more efficient than the phrase representation.

The next phase of the approach used is clustering the documents based on their similarity. Next to this phase classify the clusters in to their predefined categories. In this stage of the KDT process, experimentations were conducted using the most commonly used semi-supervised machine learning algorithms and finally the outputs produced were evaluated using different evaluation metrics.

4.1. Architecture of Amharic Text News classification

Amharic news classification using semi-supervised has its own architecture. The architecture is composed of five components. These are document collection, document preprocessing and representation, clustering, classification and evaluating the results. The architecture of Amharic document classification system is described in Figure 1.

Some of the documents are collected from ENA manually and then preprocessing of documents was held.

In the document preprocessing stage transliteration, tokenization, normalization, stop words and numbers removal, stemming and dimension reduction were done.

Once all these document preprocessing and representation activities were done, the datasets were prepared in an appropriate format and given to the learning algorithms. The learning algorithms process this dataset and group them into the appropriate clusters and classify to its category and finally the performances of those classification algorithms were evaluated using different classification evaluation metrics. The details of each phase are discussed in the following sections.

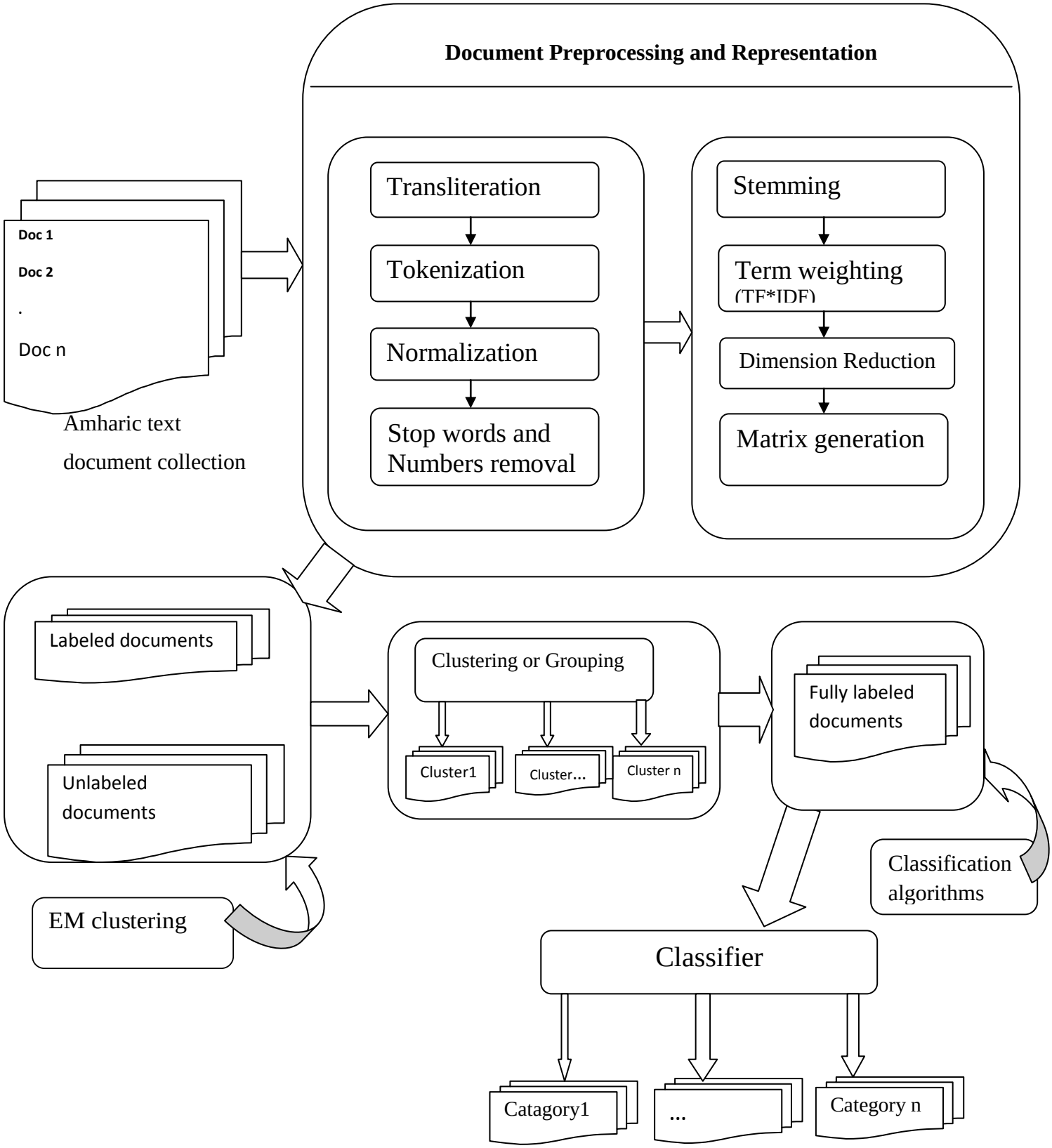


Figure 1Semi-Supervised Amharic text classification Architecture

4.2. Document Collection

The document data set that was used for the experiments were Amharic text News which was collected from ENA and used by other previous researchers. Even though, classification of news items is done manually, ENA uses software called ENASoft to make the management of news items easy. Once the classification task is done manually, ENASoft is used to dispatch news items into different Media such as Ethiopian Radio and Television, Addis Zemen, Sheger FM and others. The total number of categories collected and considered in this study are 10; with a total 3,154 Amharic news items or documents.

4.3. Document Preprocessing

Document preprocessing is important to improve the accuracy, efficiency, and scalability of the classification process. In order to get better experiment results, language dependent document 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 includes a removal of non-informative words or characters from the text. It is the first step in the preparation of documents to present them in a format suitable for classification.

The process of tokenization, normalization and stemming is language-dependent and in this thesis the different characteristics or features of the Amharic language were considered in the development of the algorithms. The document preprocessing task was implemented using python programming language (Python 3.1). The document preprocessing activities done in this thesis are presented in the following subsections.

4.4. Amharic Document Transliteration

For ease of use and compatibility purposes, the Amharic documents originally written using the Amharic script fidel were transliterated to an ASCII representation using a file conversion utility called g2 command. g2 command was made available to us through Daniel Yacob of the

Ge'ez Frontier Foundation (Daniel). Both document preprocessing activities and the experiments were done using the transliterated form in order to simplify spelling normalization of Amharic characters and to make it compatible with the classification tool used for the experiments.

Tokenization

Tokenization is the process of breaking down of documents into individual tokens.

In the tokenization process irrelevant and noisy features for the classification process such as punctuation marks and any non Amharic characters were removed from documents in the collection using python. This is because these features are not relevant to represent the content of documents and they have no contribution in discriminating one document or category from the other.

```
Read document file
Read punctuations list
Read unnecessary characters list
For each token in file
        If token ends with punctuation then
                Remove punctuation from file
        End if
        If token is in characters list
                Remove token from file
        End if
End for
```

Figure 2document tokenization algorithm

Normalization

In the Amharic writing system characters with the same sound have different symbols. These different symbols must be considered as similar because they do not have effect on meaning. As a result, in this study, all different symbols of the same sound were converted to one common form. In order to exploit this equivalence Lakechew's (Lakechew, 2011) algorithm was used. Thus, for example, if the character was one of ሐ፣ሓ፣ኀ፣ኻ ኃ or ኧ (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.

Read document file
Read characters list
For each token in file
    If token is ሐ or ሓ or ኀ or ኻ ኃ or ኧ *then*
        token is ሀ
    elseif ሠ
        token is ሰ
    elseif ዐ
        token is አ
    End if
End for

Figure 3 Normalization algorithm

Stemming

Natural language texts are characterized by variations in word forms. The most common ways of creating word variant are suffixing and prefixing. In general, word variants may be caused by factors including grammar requirements, national or local usage, transliteration, abbreviation, and spelling errors. Stemming might be used to normalize word variants by removing affixes through identification of word-stems from full words.

```
Read document file
Read exception list
Read prefix list
Read suffix list
Assign the first 1, 2, 3,… character(s) of the token to prefix
Assign the last 1, 2, 3,… character(s) of the token to suffix
For each token in file
        If token is not in exception list and prefix is in prefix list
                Remove prefix from token
        End if
        If token is not in exception list and suffix is in suffix list
                Remove suffix from token
        End if
End for
```

Figure 4 Stemming algorithm

In this study, tools for removal of common prefixes and suffixes, correction variations due to transliteration, correcting common spelling variation, and normalizing different forms of words are adapted from Nega Alemayehu(Nega, 2002).

Stop Word Removal

In case of irrelevant attributes in the dataset, attribute subset selection can be used to find a reduced set of attributes while keeping the original data class distribution as much as possible (Yohannes, 2007).

After a document is processed and its features identified, different techniques are used to select the features that adequately represent the document for the purpose of text classification.

Removal of stop words is one method of feature selection. Stop words are sometimes defined as function words. Function words have important role in grammar but carry little meaning, and, therefore, do not contribute much to categorization (Yu, 2005).

The stop words are of two kinds: those which are common to Amharic language text and those Amharic news items. 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 are usually found at the end of a sentence. Hence, news specific common words of this type were used as a stopword list. Both types of common words were used by Lakechew (Lakechew, 2011) 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 stop words from each document during the indexing process.

```
Read document file
Read stop word list
For each token in file
        If token is in stop word list then
                Remove token from file
        End if
        If token is number then
                Remove token from file
        End if
End for
```

Figure 5 Stop word removals

Compound words and abbreviations expansion

Compound words and abbreviations have different ways writing styles leading to inconsistency in writing. This different and inconsistent representation of compound words and abbreviations was solved by expanding all the short forms into their expanded form.

Concatenation of Compound Words

There are different representations of compound words in Amharic writing which result in an increase in the dimension of the vector space. Hence, to solve this problem, algorithm 8 was used to convert the expanded form into a single common standard form after creating a list that contained such type of words.

```
Read document file
Read compound words list
For each token in file
        If token is in list then
                Concatenate token with the next token
        End if
End for
```

Figure 6 Concatenation of compound words

Term Weighting

For the purpose of classification and clustering, a document can be considered as a collection of key words. These key words are often called features or attributes of the document. All terms or words with in a document are not relevant equally to represent the contents of the document. Term weighing is used to weight representative terms that describe and summarize document content based on the importance of terms with in a document. Hence, in order to define the importance of a word within Amharic text documents, a vector representation was used, where for each word a numerical importance value is stored using the TF*IDF term weighting approach.

Dimension Reduction

Document representation using bag of words create a problem in that the feature space becomes very high dimensional which imposes a big challenge on the performance of clustering algorithms. The computational complexity of any operations with such feature vectors will be proportional to the size of the feature vector (Yang, 1997).

In addition, it has been shown that some specific words in specific languages only add noise to the data and removing them from the feature vector actually improves classification performance (Yang, 1997).

Feature selection not only reduces the high dimensionality of the feature space, but also provides better data understanding, which improves the classification and clustering result (Sebastiani, 2002). Hence it is important to reduce the size of the feature vector by selecting only relevant terms that leads to better clustering performance.

There are various methods applied for dimensionality reduction in document categorization.

Some common examples are Information Gain (IG), Mutual Information (MI), Chi-Square Statistic, Term Strength (TS), and Document Frequency (DF) thresh holding.

4.5. Document classification and Evaluation

A number of different techniques are used to reliably estimate the accuracy of classifiers. The techniques include Naivesbayes, Hyperpipes and RBF network for classification and EM for clustering purpose. In the experimental part of this study cross-validation technique is used.

Cross-validation

The cross-validation method can be generalized into two as k-fold cross-validation and stratified cross-validation.

In k-fold cross-validation the initial data are randomly partitioned into mutually exclusive subsets (folds), D_1, D_2, …, D_k each of approximately equal size. In iteration i, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model, which will continue for all K iterations. Classification accuracy is estimated by dividing the overall number of correct classification from the iterations by the total number of instances (documents) in the initial data (Arzucan, 2002).

In stratified cross-validation, the folds are stratified so that the class distribution of the documents in each fold is approximately the same as that in the initial data.

Generally in practice 10-fold cross-validation is employed for estimating accuracy due to its relatively low bias and variance. The 10-fold cross-validation is used for all experiments in this research.

Evaluation of a classifier can be conducted by measuring its efficiency and its effectiveness. Efficiency is typically measured by using the elapsed processor time and it refers to the ability of a classifier to run fast. Efficiency of a classifier can usually be measured on two dimensions: learning efficiency (i.e., the time a machine learning algorithm takes to generate a classifier from a set of training examples) and categorization efficiency (i.e., the time the classifier takes to assign appropriate categories to a new document). Because of the unstable nature of parameters on which the evaluation depends, efficiency is rarely used as the singular performance measure in text categorization. However, efficiency is important for the practical application of the system.

A much more common evaluation method for text categorization systems is effectiveness: this refers to the ability to take the right decisions on the categorization of new incoming documents. There are several commonly used performance measures of effectiveness. However, there is no agreement on one single measure for use in all applications. Indeed, the type of measure that is preferable depends on the characteristics of the test data set and on the user's interests. The absence of one optimal measure of effectiveness makes it very difficult to compare the relative effectiveness of classifiers (Arzucan, 2002).

In the next section, the study will discuss various performance measures of effectiveness that have been widely used for the evaluation of text categorization systems.

4.6. Performance Measures of Effectiveness

While a number of different conventional performance measures are available for the effectiveness evaluation for text categorization, the definition of almost all measures is based on the same 2×2 contingency table model that is constructed as shown in following Table 2

In this table, 'YES' and 'NO' represent a binary decision given to each document dj under category ci. Each entry in the table indicates the number of documents of the specified type:

- TP_i: the numbers of true positive documents that the system predicted was YES, and were in fact in the category c_i.
- FP_i: the number of false positive documents that the system predicted were YES, but actually were not in the category c_i.
- FN_i: the numbers of false negative documents that the system predicted were NO, but were in fact in the category c_i.
- TN_i: the numbers of true negative documents that the system predicted were NO, and actually were not in the category ci.

Here, note that the larger TP_i and TN_i values are (or the smaller FP_i and FN_i values are), the more effective c_i is.

category c_i		label by human expert	
		YES is correct	**NO** is correct
label by the system	predicted **YES**	TP_i	FP_i
	predicted **NO**	FN_i	TN_i

Table 2 effectiveness evaluation for text categorization

Given such a two-way contingency table, most conventional performance measures compute a single value from the four values in the table. The standard performance measures for classic information retrieval research are recall and precision that has been also frequently adopted for the evaluation for the text categorization.

These measures are computed as follows.

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad \text{if } TP_i + FN_i > 0$$

……equation 22

$$Precision = \frac{TP_i}{TP_i + FP_i} \quad \text{if } TP_i + FP_i > 0$$

……equation 23

Recall measures the proportion of documents that are predicted to be YES and correct, against all documents that are actually correct. While, the precision is the proportion of documents which are both predicted to be YES and are actually correct, against all documents that are predicted YES. In general, the higher precision is, the lower recall becomes, and vice versa. For example, we can achieve very high precision by rarely predicting 'YES' (i.e., by setting a very high threshold value) or very high recall by rarely predicting 'NO' (i.e., by setting very low threshold value). For this reason, they are seldom used alone as a sole measure of effectiveness. Instead, it is common in the literature to show two associated values of recall and precision at each level.

Other performance measures that are purely based on the contingency table are accuracy and error. They are defined as follows:

$$Accuracy = \frac{TP_i +}{|D|} \quad \text{where } |D| = TP_i + FP_i + FN_i + TN_i > 0$$

……equation 24

Accuracy and error are also used for performance measures in text categorization. The accuracy and error are defined as the proportion of documents that are correctly predicted and the proportion of documents that are wrongly predicted, respectively. Both measures, in common, have |D| which is the total number documents in their denominator.

CHAPTER FIVE

EXPERIMENT AND PERFORMANCE EVALUATION

Introduction

This chapter discusses the results obtained from the experiment. The experiments are performed based on the concepts discussed in the previous chapters.

The experiments were done using three document classification and one clustering algorithms: Naivesbay's, Hyperpipes and RBF Network classification algorithm and EM clustering algorithms. The results obtained from these classifications and clustering algorithms are discussed and a comparison of these classification algorithms was done to select the best classification solution among the algorithms.

5.1. Experimentations setup for supervised

For supervised experiment the researcher used the same data with semi-supervised, shown in Table 3, a total of 10 classes and 3154 documents were used in the experimentation process. The all documents are labeled to its pre-defined classes with the corresponding provided by ENA.

To test the performances of Naives bay's, Hyperpipes and RBF Network classification algorithms at increasing number of classes and documents, the different pre-defined number of classes and the corresponding pre-classified documents were used to conduct the experiments. The 10 categories were divided into three and the experiments were done on 4, 7 and 10 number of classes using 1250, 2200 and 3154 documents respectively as shown in Table 3. The first experiment was don on four classes: 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected. The second experiment was performed on seven classes: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', and maheberawiguday. The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Experiments	No.	List of Classes used	Number of documents	Algorithms used
On four classes	1	1. Economy 2. politica 3. sport 4. tena	292 301 335 322 Total 1250	1. Naives baye's 2. Hyperpipes 3. RBF Network
On seven classes	2	1. Economy 2. politica 3. sport 4. tena 5. bahelnaturism 6. science 7. maheberawiguday	292 301 335 322 335 301 314 Total 2200	1.Naives baye's 2.Hyperpipes 3.RBF Network
On ten classes	3	1. Economy 2. politica 3. sport 4. tena 5. bahelnaturism 6. science 7. maheberawiguday 8. Tmhert 9. Heg 10. Adega	292 301 335 322 335 301 314 297 319 338 Total 3154	1.Naives baye's 2.Hyperpipes 3.RBF Network

Table 3 Experimentations setup

5.1.1. Naïve Bays Test

As explained in chapter two Naïve Bays is one of the simple algorithms of machine learning. A naive Bayes classifier could be defined as an independent feature model deals with a simple probabilistic classifier based on applying Bays' theorem with strong independence assumptions. The test results for the naïve Bays classifier is discussed in the following sections.

Experiment on four classes

Four classes 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected; where 1250 news items were used. 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. The following confusion matrix details are for the four classes:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         956               76.48 %
Incorrectly Classified Instances       294               23.52 %

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 209  38  19  26 |   a = economy
  91 194  29  22 |   b = politica
   6   2 264  12 |   c = sport
  14   5  30 289 |   d = tena
```

Figure 7 confusion matrix for four classes using Naivebays

The first row indicates that 209 documents are classified correctly as the category 'economy';83 documents from this category are misclassified as other category.38 as 'politica'; 19 as 'sport' and 26 as 'tena'. The second row indicates 91 documents from the category 'politica' are classified incorrectly to the category 'economy'; 194 documents are classified correctly; 29 documents classified incorrectly to the category 'sport' and 22 documents are classified incorrectly to the category 'tena'. The third row indicates 6 documents from the category 'sport' are classified incorrectly to the category 'economy'; 2 documents from the category 'sport' are classified incorrectly to the category 'politica'; 264 documents are classified correctly; and 12 documents classified incorrectly to the category 'tena'. In the same manner, for the fourth row, category 'tena', 14 documents classified incorrectly as a category 'economy', 5 documents are

classified incorrectly to the category ‘politica’ 30 document is classified incorrectly to the category ‘sport’ and 289 documents classified correctly in the category.

As we can see from the above experiment result the algorithm classified 76.48% of the documents correctly and 23.52 % of the document incorrectly. That is correctly classified news items are 956 out of 1251. The highest confusion (91) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on Seven classes

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1330	60.4545 %
Incorrectly Classified Instances	870	39.5455 %

=== Confusion Matrix ===

```
   a   b   c   d   e   f   g   <-- classified as
 214  33  19  33  28  41   6 |   a = science
  39  90  52  49  29  44   6 |   b = economy
  24  38 195  17  27  21  22 |   c = politica
   2   2   0 187  17  16   0 |   d = tena
  16  17   9  34 210  28   7 |   e = bahelnaturism
  28  24  11  47  36 182   6 |   f = maheberawiguday
   4   4   5   3   9  17 252 |   g = sport
```

Figure 8 confusion matrix for four classes using Naivebays

As we can see from the above experiment result the algorithm classified 60.4545 % of the documents correctly and 39.5455 % of the document incorrectly. The highest confusion (49) happened between tena and economy followed by tena and maheberawiguday (47). This shows that these classes have a lot in common.

Experiment on ten classes

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        2223               69.7084 %
Incorrectly Classified Instances       966               30.2916 %
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 284   3   1   2   1   0   3   5   2   4 |   a = politica
  11 168  32  36  26  23  34  17  16  11 |   b = economy
   4  17 235   6  23  19  17  27  32  10 |   c = heg
   3  23  14 168  10  18  18  13  16   5 |   d = science
   0   4  10   1 264  11   7   3  10   0 |   e = tena
   1  12  30  11  31 138  13  21  18   5 |   f = maheberawiguday
   2   4  11   4  13   3 283   3   2   0 |   g = tmhert
   6   7  26   3   8  14  11 266   9   8 |   h = bahelnaturism
   2   4  24  11  18  17   6  14 198   4 |   i = adega
   2   2   8   2   4   5   1  12   6 219 |   j = sport
```

Figure 9 confusion matrix for ten classes using Naivebays

From the above experiment result we can see that the algorithm classified 69.7084 % of the documents correctly and 30.2916 % of the document incorrectly. From the confusion matrix the highest confusion (36) happened between science and economy followed by tmhert and economy (34). This shows that these classes are more related.

5.1.2. Hyperpipes

HyperPipes is a very simple algorithm that constructs a "hyperpipe" for every class in the data set; each hyperpipe contains each attribute-value found in the examples from the class it was built to cover. An example is classified by finding which hyperpipes covers it the best. Extremely simple algorithm, but has the advantage of being extremely fast, and works quite well when you have lots of attributes.

Experiment on four classes

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances         925               74
%
Incorrectly Classified Instances       325               26
%
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 202  56  11  23 |   a = economy
  97 204  12  23 |   b = politica
  16  10 249   9 |   c = sport
  25  18  25 270 |   d = tena
```

Figure 10 confusion matrix for four classes using hyperpipe

As we can see from the above experiment result the algorithm classified 74% of the documents correctly and 26% of the document incorrectly. From the confusion matrix the highest confusion (97) happened between politica and economy. This shows that these classes are more related.

Experiment on seven classes

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1171               53.2273 %
Incorrectly Classified Instances      1029               46.7727 %

=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 251  28  33  10  13  36   3 |   a = science
  85  86  55  19  22  38   4 |   b = economy
  52  74 154   5  19  18  22 |   c = politica
  27  22   3 126  25  21   0 |   d = tena
  41  29  13  19 184  31   4 |   e = bahelnaturism
  75    43    20    24    37  133      2  |      f  =
maheberawiguday
   9   9  10   1  11  17 237 |   g = sport
```

Figure 11 confusion matrix for seven classes using hyperpipe

As it is shown in the above the algorithm classified 53.2273% of the documents correctly and 46.7727 % of the document incorrectly. As we can see from confusion matrix the highest confusion (85) happened between politica and economy followed by maheberawiguday and economy (75). This indicates that these classes are more related each other.

Experiment on ten classes

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1484	46.535 %
Incorrectly Classified Instances	1705	53.465 %

=== Confusion Matrix ===

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 128  67  31  10   5   9  11  19   7  18 |   a = politica
  65 158  38  26  22  10  35  14   5   1 |   b = economy
  15  46 193  11  26  19  23  34  20   3 |   c = heg
   9  53  27 117  11  10  26  18  15   2 |   d = science
   3  17  28   7 203  17  21   2  12   0 |   e = tena
   7  25  56  17  34  81  22  25  11   2 |   f = maheberawiguday
   6  50  38   9  34  19 137  19  12   1 |   g = tmhert
  18  17  59  22  16  22  22 162  15   5 |   h = bahelnaturism
  12  16  60  14  27  18  13  16 122   0 |   i = adega
  16   5  19   0   4   7   5  14   8 183 |   j = sport
```

Figure 12 confusion matrix for ten classes using hyperpipe

As we can see from the above experiment result the algorithm classified 46.535% of the documents correctly and 53.465 % of the document incorrectly. From the confusion matrix the highest confusion (65) happened between politica and economy followed by heg and adega (60). This shows that these classes have a lot in common.

5.1.3. RBF network

Radial basis function (RBF) networks have a static Gaussian function as the nonlinearity for the hidden layer processing elements. The Gaussian function responds only to a small region of the input space where the Gaussian is centered. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions. The simulation starts with the training of an unsupervised layer. Its function is to derive the Gaussian centers and the widths from the input data. These centers are encoded within the weights of the unsupervised layer using competitive learning. During the unsupervised learning, the widths of the Gaussians are

computed based on the centers of their neighbors. The output of this layer is derived from the input data weighted by a Gaussian mixture (Stig-Erland, 2007).

The advantage of the radial basis function network is that it finds the input to output map using local approximators. Usually the supervised segment is simply a linear combination of the approximators. Since linear combiners have few weights, these networks train extremely fast and require fewer training samples.

Experiment on four classes

Time taken to build model: 1.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	902	72.16 %
Incorrectly Classified Instances	348	27.84 %

=== Confusion Matrix ===

```
   a   b   c   d   <-- classified as
 206  39  20  27 |   a = economy
  93 170  46  27 |   b = politica
   7   3 247  27 |   c = sport
  23   6  30 279 |   d = tena
```

Figure 13 confusion matrix for four classes using RBF Network

As it is depicted in the above the algorithm classified 72.16% of the documents correctly and 27.84 % of the document incorrectly. As we can see from the confusion matrix the highest confusion (93) happened between politica and economy. This shows that these classes are more related.

Experiment on seven classes

Time taken to build model: 1.77 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1243               56.5    %
Incorrectly Classified Instances       957               43.5    %
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 203  34  19  47  10  58   3 |   a = science
  35  88  48  64  12  59   3 |   b = economy
  20  42 189  26  16  36  15 |   c = politica
   3   2   0 183   7  29   0 |   d = tena
  16  13  11  47 140  93   1 |   e = bahelnaturism
  23  21  10  48  18 213   1 |   f = maheberawiguday
   3   6  11  15   4  28 227 |   g = sport
```

Figure 14 confusion matrix for seven classes using RBF Network

As we can see from the above experiment result the algorithm classified 56.5% of the documents correctly and 43.5 % of the document incorrectly. From the confusion matrix the highest confusion (64) happened between politica and economy followed by maheberawiguday and economy (59). This shows that these classes have a lot in common.

Experiment on ten classes

Time taken to build model: 48.89 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1890               59.2662 %
Incorrectly Classified Instances      1299               40.7338 %
=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 288   2   4   2   1   1   0   3   2   2 |   a = politica
  11 133  54  49  26  72  10   8  11   0 |   b = economy
   4  15 224  11  19  73   8   9  26   1 |   c = heg
   4  30  28 155  13  34   6   6  11   1 |   d = science
   0  18  17   4 243  16   1   3   8   0 |   e = tena
   1  29  47  11  29 142   6   3  11   1 |   f = maheberawiguday
   3  18  61  14  27  10 187   5   0   0 |   g = tmhert
   4  15  31   7  14  54  40 181   9   3 |   h = bahelnaturism
   3   5  49  11  20  31   4   9 163   3 |   i = adega
   3  14  21   4   6  24   0   3  12 174 |   j = sport
```

Figure 15 confusion matrix for ten classes using RBF Network

As we can see from the above experiment result the algorithm classified 59.2662 % of the documents correctly and 40.7338 % of the document incorrectly. The confusion matrix shows the highest confusion (54) happened between politica and economy followed by heg and adega (54). This shows that these classes have a lot in common.

Number of classes	Naivebays accuracy(%)	Hyperpipes accuracy(%)	RBF Network accuracy (%)
Four	76.48	74	72.16
Seven	60.4545	53.2273	56.5
Ten	69.7084	46.535	59.2662

Table 4 Comparision of algorithms at different class level

As shown in Table and figure above, Naivebays achieved the highest performance in terms of accuracy.

5.2. Experimentations setup for semi-supervised learning

As shown in Table 3, a total of 10 classes and 3154 documents were used in the experimentation process. The list of pre-defined classes with the corresponding documents is already provided by ENA. These pre-classified documents were used as a centroid before clustering.

To test the performances of Naives bay's, Hyperpipes and RBF Network classification algorithms at increasing number of classes and documents, the different pre-defined number of classes and the corresponding pre-classified documents were used to conduct the experiments. The predefined classes were arranged in the manner that each classes are not related. That means classes that have great similarity have smaller probability to be selected at the same time. The 10 categories were divided into three and the experiments were done on 4, 7 and 10 number of classes using 1250, 2200 and 3154 documents respectively as shown in Table 3.

5.2.1. Naïve Bayes Test

Experiment on four categories

Four classes 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected; where 1250 news items were used. 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. The following confusion matrix details are for the four classes:

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1043	83.44 %
Incorrectly Classified Instances	207	16.56 %
Total Number of Instances	1250	

=== Confusion Matrix ===

```
  a   b   c   d   <-- classified as
237  39   4  12 |   a = economy
 81 187  13  20 |   b = politica
  7   5 307  16 |   c = sport
  6   4   0 312 |   d = tena
```

Figure 16 confusion matrix for four classes using Naivesbays

The first row indicates that 237 documents are classified correctly as the category ‘economy’;55 documents from this category are misclassified as other category.39 as ‘politica’; 4 as ‘sport’ and 12 as ‘tena’. The second row indicates 81 documents from the category ‘politica’ are classified incorrectly to the category ‘economy’; 187 documents are classified correctly; 13 documents classified incorrectly to the category ‘sport’ and 20 documents are classified incorrectly to the category ‘tena’. The third row indicates 7 documents from the category ‘sport’ are classified incorrectly to the category ‘economy’; 5 documents from the category ‘sport’ are classified incorrectly to the category ‘politica’; 307 documents are classified correctly; and 16 documents classified incorrectly to the category ‘tena’. In the same manner, for the fourth row, category ‘tena’, 6 documents classified incorrectly as a category ‘economy’, 4 documents are classified incorrectly to the category ‘politica’ 0 document is classified incorrectly to the category ‘sport’ or there is no economy document that are predicted as politica and 312 documents classified correctly in the category. So, correctly classified news items are 1,052 out of 1251and the average accuracy is 83.44 percent. The highest confusion (81) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on Seven Categories

The second experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, and maheberawiguday.

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1576               71.6364 %
Incorrectly Classified Instances       624               28.3636 %
Total Number of Instances             2200
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 203  27   3  13  12  14  20 |   a = economy
  58 167  11  10  20  17  18 |   b = politica
   1   3 296   6  15   2  12 |   c = sport
   4   4   0 280   2   9  23 |   d = tena
   9   2   6  21 261   6  30 |   e = bahelnaturism
  19   6   2  23  15 205  31 |   f = science
  16      9      3    69    31    22  164  |      g  =
maheberawiguday
```

Figure 17 Confusion matrix for seven classes using Naivesbays

As we can see from the above experiment result the algorithm classified 71.6364% of the documents correctly and 28.3636 % of the document incorrectly. From the confusion matrix we can see that the highest confusion (69) is happened between tena and maheberawiguday followed by ecoinomy and politica(58). This shows that tena and maheberawiguday have a lot in common.

Experiment on ten Categories

The third experiment was performed on ten categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, ‘maheberawiguday’, ‘tmhert’,’heg’ and ‘adega’.

Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        2032               64.4261 %
Incorrectly Classified Instances      1122               35.5739 %
Total Number of Instances             3154
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 197  26   1   9  13   8  12  10   6  10 |   a = economy
  53 153   8   5  13  14  13   9  26   7 |   b = politica
   1   3 288   5  10   2  11   6   5   4 |   c = sport
   4   1   0 263   1   9  12  18   3  11 |   d = tena
   5   0   5  17 241   4  23  16  13  11 |   e = bahelnaturism
  12   5   0  11  11 179  18  47   6  12 |   f = science
  16   7   3  49  18  18 124  38  17  24 |   g = maheberawiguday
   2   3   0  31   6  17   9 222   3   4 |   h = tmhert
  10  15   2  17  29   7  34  29 146  30 |   i = heg
   3  14   1  23  20  13  20  14  11 219 |   j = adega
```

Figure 18 confusion matrix for ten classes using Naivesbays

As we can see from the above experiment result the algorithm classified 2032 (64.4261%) of the document correctly and 1122(35.5739%) of the documents incorrectly. As it is shown in the confusion matrix the highest confusion (53) is happened between ecoinomy and politica followed by tena and maheberawiguday). This shows that ecoinomy and politica have a lot in common.

Number of classes	Accuracy performance achieved
4	83.44 %
7	71.6364 %
10	64.4261 %

Table 5 accuracy performances achieved at different levels of class using Naivebays algorithm

From the above table, the highest accuracy is 83.44% and the lowest is 64.4261 % when the number of class is four and ten respectively. The seven class category yields an accuracy of 71.6364 %. From this we can deduce that when the number of class increase the accuracy goes in a reverse way. This is due to the increase of similarity between classes.

5.2.2. Hyper Pipes test

Experiment on four categories

The first experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, and ‘tena’.

HyperPipes classifier

Time taken to build model: 0.01 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1035	82.8 %
Incorrectly Classified Instances	215	17.2 %
Total Number of Instances	1250	

=== Confusion Matrix ===

```
   a   b   c   d   <-- classified as
 242  38   1  11 |   a = economy
  95 176  17  13 |   b = politica
   6   4 320   5 |   c = sport
  16   9   0 297 |   d = tena
```

Figure 19 confusion matrix for four classes using Hyperpipes

As we can see from the confusion matrix details 1035 news items out of 1250 are correctly classified and the average percent accuracy is 82.8%. The highest confusion (95) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on seven categories

The second experiment was performed on seven categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', and 'maheberawiguday'.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1529	69.5 %
Incorrectly Classified Instances	671	30.5 %
Total Number of Instances	2200	

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 211  34   4   7   8  16  12 |   a = economy
  70 166  15  12  17   7  14 |   b = politica
   4   3 304   2  11   2   9 |   c = sport
   9   5   0 261   7  10  30 |   d = tena
   9   9  10  23 259   9  16 |   e = bahelnaturism
  26  11   2  27  18 188  29 |   f = science
  25  13   7  62  37  30 140 |   g = maheberawiguday
```

Figure 20 confusion matrix for seven classes using HyperPipes

As shown from the above confusion matrix details 1529 news items out of 2,200 are correctly classified and the average percent accuracy is 69.5 percent. As we can see from the confusion matrix the highest confusion (70) happened between politica and economy followed by tena and maheberawiguday (62). This shows that these classes have a lot in common.

Experiment on ten categories

The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1944	61.636 %
Incorrectly Classified Instances	1210	38.364 %
Total Number of Instances	3154	

```
=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 196  32   3   6  11  13  12   7   6   6 |   a = economy
  69 151   9   6  16   9  10   8  18   5 |   b = politica
   2   3 298   2  12   4   8   4   1   1 |   c = sport
   9   4   0 238   3   8  23  21   2  14 |   d = tena
  10   5  10  19 248   6  19   8   6   4 |   e = bahelnaturism
  21   6   1  18   8 181  22  29   7   8 |   f = science
  16   9   7  48  27  20 125  24  21  17 |   g = maheberawiguday
  12   3   0  43  12  26  23 170   4   4 |   h = tmhert
  13  31   4  23  22  10  35  17 142  22 |   i = heg
  10  11   4  27  20  15  29   5  22 195 |   j = adega
```

Figure 21 confusion matrix for ten classes using Hyperpipes

As shown from the above figure confusion matrix details 1,944 news items out of 3,154 are correctly classified and the average percent accuracy is 61.636 percent. From the confusion matrix we can see that the highest confusion (69) happened between politica and economy followed by tmhert and tena(48). This shows that these classes have a lot in common each other.

Number of classes	Accuracy performance achieved
4	82.8 %
7	69.5%
10	61.636%

Table 6 Accuracy performance achieved at different levels of class using Hyperpipe algorithm

From the above table, the highest accuracy is 82.8% and the lowest is 61.636 % when the number of class is four and ten respectively. The seven class category yields an accuracy of 69.5%. From this we can conclude that when the number of class increase the accuracy goes in a reverse way.

5.2.3. Radial basis function network (RBF Network) Test

Experiment on four categories

The first experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, and ‘tena’.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Radial basis function network

Time taken to build model: 0.22 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1030	82.4 %
Incorrectly Classified Instances	220	17.6 %
Total Number of Instances	1250	

```
=== Confusion Matrix ===
   a   b   c   d   <-- classified as
 246  28   5  13 |   a = economy
  93 167  28  13 |   b = politica
   2   4 318  11 |   c = sport
  13   8   2 299 |   d = tena
```

Figure 22 confusion matrix for four classes using RBF Network

Correctly classified news items are 1,030 out of 1251and the average accuracy is 82.4%. The confusion matrix shows that the highest confusion (93) happened between politica and economy. This shows that these classes are more related.

Experiment on seven categories

The second experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, and ‘maheberawiguday’.

Time taken to build model: 2.2 seconds

=== Stratified cross-validation ===

=== Summary ===

```
Correctly Classified Instances        1501               68.2273 %
Incorrectly Classified Instances       699               31.7727 %
Total Number of Instances             2200

=== Confusion Matrix ===
   a   b   c   d   e   f   g   <-- classified as
 206  28  12   8  30   4   5 |   a = economy
  62 157  14  16  37  13   1 |   b = politica
  17   7 193  12  65   1   6 |   c = science
  12   6   7 213  77  11   9 |   d = bahelnaturism
  17   8  19  15 220   7  28 |   e = maheberawiguday
   3   4   1  10  28 288   4 |   f = sport
   6   7  15   6  61   0 224 |   g = tena
```

Figure 23 confusion matrix for seven classes using RBF Network

As shown from the above confusion matrix details 1501 news items out of 2,200 are correctly classified and the average percent accuracy is 69.5 percent. Confusion matrix of the experimental result indicates that the highest confusion (62) happened between politica and economy followed by tena and maheberawiguday (61). This shows that these classes are more related.

Experiment on ten categories

The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Time taken to build model: 10.09 seconds

=== Stratified cross-validation ===

=== Summary ===

```
Correctly Classified Instances        1673               53.0438 %
Incorrectly Classified Instances      1481               46.9562 %
Total Number of Instances             3154

=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 179  34   2  14   8   4  29  14   7   1 |   a = economy
  49 143   6  13   7   8  34  13  25   3 |   b = politica
   3   6 274   4  12   2  18   9   5   2 |   c = sport
   7   4   0 227   2   0  29  40   4   9 |   d = tena
   5   4   6  18 204   1  63  26   6   2 |   e = bahelnaturism
  21  15   4  21  18  53  45 107  13   4 |   f = science
  10   9   3  38  20   4 143  60  19   8 |   g = maheberawiguday
   3   4   0  30   5  26  20 203   5   1 |   h = tmhert
  13  19   1  15  24   7  92  23 114  11 |   i = heg
   6  22   1  38  18  10  48  29  33 133 |   j = adega
```

Figure 24 confusion matrix for ten classes using RBF Network

As shown from the above confusion matrix details 1673 news items out of 3154 are correctly classified and the average percent accuracy is 69.5%. From the confusion matrix the highest confusion (92) happened between heg and maheberawiguday followed by politica and economy (49). This shows that these classes have a lot in common.

Number of classes	Accuracy performance achieved
4	82.4 %
7	68.2273%
10	53.0438%

Table 7 accuracy performances achieved at different levels of class using RBF Network algorithm

From the above table, the highest accuracy is 82.4% and the lowest is 53.0438% when the number of class is four and ten respectively. The seven class category yields an accuracy of 68.2273%. From this we can say that when the number of classes increases the accuracy goes in a reverse way.

The above all experimental confusion matrix shows that politca and economy have a lot in common. This means these classes are more related followed by tena and maheberawiguday.

DISSCUSSIONS

The classification algorithms used in the experiments presented in this paper were implemented from WEKA package. These algorithms were chosen to include diverse set of paradigms, while high computational efficiency. Based on the above experimental results, we can clearly see that the highest accuracy is 83.44 % and the lowest is 82.4% when the number of class is four. The other algorithm yields an average accuracy of 82.8%. In fact, the highest accuracy belongs to the NaiveBayes classifier, followed by Hyperpipes and Radial basis function with a percentage of 82.8% and 82.4%.

5.3. Comparison of classification Algorithms

In this research different classification algorithms were used. The performances of each document classification algorithms: Naivebayes, Hyperpipe and RBF network were compared using their accuracy. Table below shows the comparison of the classification results obtained by Naivebayes, Hyperpipe and RBF network for the 4, 7 and 10 classes.

Number of classes	nave	Hyperpipe	RBF network
Four	83.44	82.8	82.4
Seven	71.636	69.5	68.2273
Ten	64.4261	61.636	53.0438

Table 8 performance evaluation at different class stages

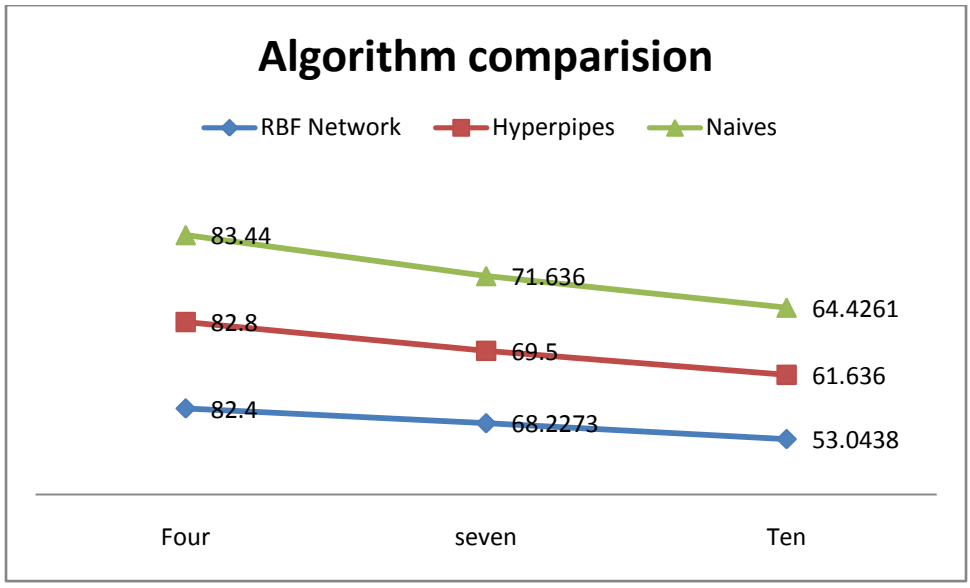


Figure 25 performance evaluation different classification algorithm and different class levels

As shown in Table and figure above, the Naivebays achieved the highest performance in terms of accuracy in all number of classes. However, the accuracy of all algorithms decreases as the number of categories increases.

In this simple experiment, from Figures, we can say Naivebays classifier requires the shortest time which is around 0.02 seconds compared to the others when class level is 10. Radial basis function (RBF) networks algorithm requires the longest model building which is 10.09 seconds.

5.1. Comparison of Supervised and Semi-supervised classification

Machine learning approaches	Number of classes	Naves	Hyperpipe	RBF network
Supervised	Four	76.48	74	72.16
	Seven	60.4545	53.2273	56.5
	Ten	69.7084	46.535	59.2662
Semi-Supervised	Four	83.44	82.8	82.4
	Seven	71.6364	69.5	68.2273
	Ten	55.4209	57.2618	51.9056

Table 9 performance comparison of semi-supervised and supervised performance

As shown experimental results above, all the results of semi-supervised classification approach is quite greater than supervised machine learning approaches. Therefore, this indicates that semi-supervised text classification is significantly better than supervised text classification.

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

In this study, the potential application of semi-supervised Learning approach for the classification of Amharic news documents was explored and is both feasible and important. The effect of the number of classes and the size of documents used on the performance and efficiency of classification algorithms was tested and compared using different data sets. Moreover, the performances of these classification algorithms were also tested at increasing number of classes using the same data set.

The study shows that semi-supervised approach for Amharic news classification significantly improve predictive accuracy over different classes, the semi-supervised approach was more successful (significantly better in for cases) than supervised and unsupervised approaches.

In this study, three classification algorithms including Naives bays, Hyperpipes and RBF Network are applied to Amharic news dataset.

Based on the experiments done in this study, the following concluding remarks were made.

- As the number of classes and documents increase, the accuracy produced by different classifiers, Naives bay's, Hyperpipes and RBF Network, become decreased and requires relatively high computational requirements. Moreover, it is learnt that considering categories with equal number of news items increases the performance of the classifiers.
- The best result obtained by the Naives bay's, Hyperpipes and RBF Network, classifiers is on four categories data (83.44% 82.8% and 82.4%) and the least performance is shown on the 10 categories data (64.4261% , 61.636, and 53.0438%) respectively. Compared to Hyperpipes and RBF Network Naives bay's classifier methods obtain a good result. Naives bay's shows that it can provide better results with larger training set. This paper indicated that naïve Bayes classifier is more applicable to Amharic news articles than Hyperpipes and RBF Network classifiers.
- All the classification algorithms: Naives bay's, Hyperpipes and RBF Network achieved better classification accuracy in semi-supervised than supervised. Therefore we can say that applying semi-supervised text classification better than supervised approach.

6.2. Recommendations

This study shows the potential application of semi-supervised machine learning techniques to the analysis of textual Amharic documents is both feasible and crucial. However, recommendations for further research are forwarded to improve the performance of document classification and to explore all algorithms and applications of semi-supervised document classification especially for local languages. Thus, the recommendations forwarded are organized as follows.

- ✓ The Naivesbayes, Hyperpipes and RBF Network classifiers used in this research have shown good accuracy. Therefore, there is a need to look for other classifiers with less processing cost and better accuracy.
- ✓ 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.
- ✓ As to the researcher's knowledge, there is no standard corpus open for researchers to apply different machine learning approaches. Researchers can devote much time on their work and explore more if standard corpus is prepared for Amharic classification experiments like 'Reuters-21578' for English.
- ✓ The bag of words representation approach which describes each document with its most significant terms was used in this thesis. However, future researchers may consider different document representation approaches such as phrase based and ontology based representations to select index or representative terms.
- ✓ Feature researchers can also compare the performance of semi-supervised learning approaches with the unsupervised approaches and two step approach using the same evaluation methods and document collections.
- ✓ Currently, few researches were conducted on automatic Amharic news classification and the results of the researches are promising. However, ENA still uses manual classification of News. So, it is better for the agency to review the different research works and to start the implementation of automatic classification of news.
- ✓ There is also a mismatch between the news categories provide by ENA and the clusters discovered by automatic clustering algorithms. Hence, it is better for the agency to revisit it news categories based on these findings.

- ✓ A number of researches were done on Amharic text document classification. However, as to the knowledge of the researcher, all the previous studies were conducted using Amharic text news item only. Future researchers can also explore document classification techniques to various real world problems such as classification and clustering of research papers and e- mail messages. Moreover, document classification and clustering techniques can also be extended to other local languages if huge collection of documents is available.

REFERENCES

Andreetto M., Zelnik Manor L., & Perona P. (2007). Non-parametric probabilistic image segmentation, in Proceedings of the International Conference on Computer Vision, pp.1–8.

Arzucan O. (2002), supervised and unsupervised machine learning techniques for text document categorization, Bo?gazi¸ci University.

Bar-Hillel, Hertz T., Shental N., & Weinshall D. (2005). Learning a mahalanobis metric from equivalence constraints, Journal of Machine Learning Research, vol. 6, pp. 937–965,.

Basu S., Bilenko M., & Mooney R. J. (2004). A probabilistic framework for semi-supervised clustering, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68.

Beletu R. (1982). A Graphemic Analysis of the Writing System of Amharic. Paper for the

Requirement of the Degree of bachelor of Art in Linguistics. Addis Ababa University.

Bennet K., Demiriz A., & Maclin R. (2002). Exploiting unlabeled data in ensemble methods, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 289–296.

Berkhin, P., (2002) Survey of clustering data mining techniques, Research paper, Accrue Software, http://www.accrue.com/products/researchpapers.html.

Bilenko M., Basu S., & Mooney R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering, in Proceedings of the International Conference on Machine Learning, vol. 69.

Bishop C. (2005). Neural Networks for Pattern Recognition. Oxford Univ.

Bishop C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Blei D. M. & Jordan M. I. (2004). Hierarchical topic models and the nested Chinese restaurant process, in Advances in Neural Information Processing Systems.

Blei D. M., Ng A. Y., & Jordan M. I. (2003). Latent dirichlet allocation, Journal of Machine Learning Research, vol. 3, pp. 993–1022.

Bradley C. L. (2002). 'Comparing supervised and unsupervised category learning': Psychonomic Bulletin & Review. University of Texas, Austin, Texas

Brandes U., Gaertler M., & Wagner D. (2003). Experiments on graph clustering algorithms, in Proceedings of the 11th European Symposiom of Algorithms, pp. 568– 579.

Brasethvik T. & Gulla J. A., (2001) Natural Language Analysis for Semantic Document Modeling. Data & Knowledge Engineering, 38, pages 45-62.

Breiman L. (1996). Bagging predictors, Machine Learning, vol. 24, no. 2, pp. 123–140.

Breiman L. (2001). Random forests, Machine Learning, vol. 45, no. 1, pp. 5–32,.

Breiman L. ,(1996). Bagging predictorsMachine Learning, vol. 24, no. 2, pp. 123–140.

Breiman L., Friedman J., Olshen R., & Stone C. (1984). Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks.

C. Apte, F. Damerau, & Weiss S. M. (1994). Automated Learning of Decision Rules for Text Categorization. ACM Transactions of Information Systems, 12(3), pages 233-251

Charniak E., (1997) Statistical Techniques for Natural Language Parsing. AI Magazine, 18(4), pages 33-44,

Cohen W. W. (1995). Fast effective rule induction, in Proc. of the 12th International Conference on Machine Learning, A. Prieditis and S. Russell, Eds. Tahoe City, CA: Morgan Kaufmann, July 9–12, pp. 115–123.

Cohen, W. W. & Singer Y. (1996). Context-sensitive learning methods for text categorization, Proceedings of the 19th Annual ACM SIGIR Conference.

Cohen, W. W. & Singer Y. (1996). Context-sensitive learning methods for text categorization. Proceedings of the 19th Annual ACM SIGIR Conference.

Comaniciu D. & Meer P. (2002). Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603–619.

Davis J., Kulis B., Jain P., Sra S., & Dhillon I.(2007). Information-theoretic metric learning, in Proceedings of the International Conference on Machine Learning, pp. 209– 216.

Demiroz G. & Guvenir H. A.(1997). Classification by voting feature intervals, in European Conference on Machine Learning, pp. 85–92.

Dempster P., Laird N. M., & Rubin D. B., (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: Series B, vol. 39, pp. 1–38.

Dempster, Laird N. M., & Rubin D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: Series B, vol. 39, pp. 1–38.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). 'A Probabilistic Theory of Pattern Recognition'. Springer Verlag.

Duda, Hart P. E., & Stork D. G. (2000). Pattern Classification. Wiley-Inter science Publication.

Dumais, S., Platt T., Heckermann D., & Sahami M. (1998). Inductive learning algorithms and representations for text categorization, Proceedings of the Seventh International Conference on Information and Knowledge Management.

Figueiredo M. & Jain A. (2002). Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 381–396,.

Forgy, E., (1965) Cluster analysis of multivariate data: Efficiency versus interpretability of classification, Biometrics, Vol. 21, pp. 768–780.

Frank E. & Witten I. H. (1998). Generating accurate rule sets without global optimization, in ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 144–151.

Frank E., Wang Y., Inglis S., Holmes G., & Witten I. H. (1998). Using model trees for

classification, Machine Learning, vol. 32, no. 1, pp. 63–76.

Freund Y. & Mason L. (1999). The alternating decision tree learning algorithm, in Proc. 16th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA , pp. 124–133.

Freund Y. and Schapire R. E. (1996). Experiments with a new boosting algorithm in Proceedings of the International Conference on Machine Learning, pp. 148–156.

Freund Y. and Schapire R. E. (1998) Large margin classification using the perceptron algorithm, in Computational Learing Theory, pp. 209– 217.

Fuernkranz, J., Mitchell T., & Riloff E. (1998). A case study in using linguistic phrases for text categorization on the www, Sahami, M., editor, In Learning for Text Categorization: Papers from the 1998 AAAI Workshop (Technical Report WS-98-05).

Geberhiwot A. B. (2011). A two step approach for Tigrigna text categorization. (Master's Thesis. Department of Information Science, Addis Ababa University, Ethiopia).

Ghani R., Jones R., & Rosenberg C. (Eds.) (2003). 'ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining', Washington, DC.

Goldberger J., Roweis S., Hinton G., & Salakhutdinov R. (2005). Neighbourhood components analysis, in Advances in Neural Information Processing Systems, vol. 17, pp. 513–520.

Hagen L. & Kahng A., (1992). New spectral methods for ratio cut partitioning and clustering, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 11, no. 9, pp. 1074–1085.

Ham F. & Park S., (2002). A Robust Neural Network Classifier for Infrasound Events Using Multiple Array data. IEEE International Joint Conference NN, vol. 3, 2615-2619.

Hartigan, J., (1975). Clustering Algorithms, John Wiley & Sons, New York, NY,.

Heller K. & Ghahramani Z. (2005). Bayesian hierarchical clustering, in Proceedings of the

International Conference on Machine Learning, vol. 22, p. 297.

Hirotoshi T., (2002). 'Text Categorization using Machine Learning'. Dissertation Thesis. Department of Information Science, Nara.

Hoi S., Jin R., & Lyu M. (2007). Learning nonparametric kernel matrices from pairwise constraints, in Proceedings of the International Conference on Machine Learning, pp. 361–368.

Holmes G., Hall M., & Frank E. (1999). Generating rule sets from model trees, in AI '99: Proceedings of the 12th Australian Joint Conference on Artificial Intelligence. London, UK: Springer-Verlag, pp. 1–12.

Jaakkola T., & Haussler, D. (1999). 'Exploiting generative models in discriminative classifiers'. In Advances in Neural Information Processing Systems 11, pp. 487–493.

Jain K. & Dubes R. C. (1988). Algorithms for Clustering Data. Prentice Hall,.

Jain K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666.

Jain K., Murty M. N., & Flynn P. J. (1999). Data clustering: A review, ACM Computing Surveys, vol. 31, no. 3, pp. 264–323.

Jain, A. K., Murty M. N., & Flynn P. J. (1999). Data clustering: A review, ACM Computing Surveys, Vol. 31, No. 3, pp. 264–323.

Joachims T. (1998). Text categorization with support vector machines: Learning with many relevant features. European Conference on Machine Learning (ECML).

Kamvar S., Klein D., & Manning C. (2003). Spectral learning, in International Joint Conference On Artificial Intelligence, vol. 18, pp. 561–566, Citeseer.

Kang H. L., (2003) Text Categorization with a Small Number of Labeled Training Examples, A

Thesis Presented, School of Information Technologies University of Sydney, Sydney

Klien, Kamvar S. D., & Manning C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering, in Proceedings of the International Conference on Machine Learning.

Kohavi R. (1995). The power of decision tables, in ECML '95: Proceedings of the 8th European Conference on Machine Learning. London, UK: Springer-Verlag, pp. 174–189.

Kohavi R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 202–207.

Lakechew Y.(2011) Unsupervised Amharic news classification. (Master's Thesis, Department of Information Science, Addis Ababa University, Ethiopia).

Landwehr N., Hall M., & Frank E. (2005). Logistic model trees, Machine Learning, vol. 59, no. 1-2, pp. 161–205.

Lange T., Law M. H., Jain A. K., & Buhmann J. (2005). Learning with constrained and unlabelled data, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 730–737.

Law M. H. C. (2006). Dimensionality Reduction and Side Information. (PhD thesis, Michigan State University).

Lee J., Jin R., & Jain A. (2008). Rank-based distance metric learning: An application to image retrieval, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Liszka L. & Holmström M.(1999). Extraction of a deterministic component from ROSAT X-ray data using a wavelet transform and the principal component analysis. Astron. Astrophys. Suppl. Ser. , 140, 125-134.

Lu Z. & Leen T. (2005). Semi-supervised learning with penalized probabilistic clustering,in Advances in Neural Information Processing Systems, p. 849856.

MacQueen J. B. (1967). Some methods for classification and analysis of multivariate observations, in Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1. Berkeley, University of California Press, pp. 281–297.

Mallapragada P. K., Jin R., Jain A. K., & Liu Y. (2009) SemiBoost: boosting for semi-supervised learning., IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 11, p. 2000.

McCallum, A. & K. Nigam. (1998). A comparison of event models for naive bayes text classification, AAAI-98 Workshop on Learning for Text Categorization.

McLachlan G. L. & Basford K. E. (1987). Mixture Models: Inference and Applications to Clustering. Marcel Dekker.

McLachlan G. L. & Peel D. (2000). Finite Mixture Models. Wiley,.

Merkl D., (1998) Text Classification with Self-Organizing Maps: Some Lessons Learned. Neurocomputing, 21:1-3, pages 61-77.

Merkl. (1998). Text Classification with Self-Organizing Maps: Some Lessons Learned. Neurocomputing, 21:1-3, pages 61-77.

MitchellT. M. , (1997), Machine Learning, McGraw Hill.

Nega A. & Peter W. (2002).Stemming of Amharic Words for Information Retrieval. Literary Linguistic Computing Vol. 17, No.1,.

Ng,M., Jordan, & Weiss Y. (2002). On spectral clustering: Analysis and an algorithm, in Advances in Neural Information Processing Systems, vol. 2, pp. 849–856.

Nigam K., McCallum A. K., Thrun S., & Mitchell T. M., (2000). Text classification from labeled and unlabeled documents using EM, Machine Learning, vol. 39, no. 2/3, pp. 103–134.

Olivier C., Bernhard S. & Alexander Z.(2006). Semi-Supervised Learning. Cambridge, Massachusetts London, England

P. Tan, Steinbach M., & Kumar V. (2005). Introduction to Data Mining. Pearson Addison Wesley Boston.

Pavan Kumar Mallapragada, (2010), Some contributions to semi-supervised learning, a dissertation, Computer Science, Michigan State University

Porter, M. F.(1980). An algorithm for suffix stripping, Program, Vol. 14, pp. 130–137.

Quinlan J. R. (1986). Induction of decision trees. Machine Learning, vol. 1, no. 1, pp. 81–106.

Quinlan J. R.(1992) Learning with Continuous Classes. In 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348.

Quinlan R.(1993).C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.

Rosenberg, Hebert M., & Schneiderman H. (2005). Semi-supervised self-training of object detection models, in Proceedings of the Workshop on Applications of Computer Vision, vol. 1, pp. 29–36.

Rosenblatt F. (1988). The perceptron: A probalistic model for information storage and orginazation in the brain, Psychological Review, vol. 65, pp. 386–407, 1958, (Reprinted in Neuro-computing).

Roweis S. & Saul L. (2000). Nonlinear dimensionality reduction by locally linear embedding, Science, vol. 290, no. 5500, pp. 2323–2326.

Sahami, M. (1998). Using Machine Learning to Improve Information Access, Ph.D. thesis, Stanford University.

Sahami, M., Dumais S., Heckerman D., & Horvitz E. (1998). A bayesian approach to filtering junk e-mail. Sahami, M., editor, Proceedings of AAAI-98 Workshop on Learning for

Text Categorization.

Salton G., Yang C., & Wong A. (1975). A vector-space model for automatic indexing, Communications of the ACM, Vol. 18, No. 11, pp. 613–620.

Schmitter, E. D. (2006). Characterisation and Classification of Natural Transients, Transactions on Engineering, Computing and Technology, vol. 13.

Scudder H. J. (1965). Probability of error of some adaptive pattern-recognition machines, IEEE Transactions on Information Theory, vol. 11, pp. 363–371.

Sebastiani F., (2002) Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1), pages 1-4.

Shalev-Shwartz S., Singer Y., & Ng A. (2004). Online and batch learning of pseudometrics, in Proceedings of the International Conference on Machine Learning,.

Shental N., Bar-Hillel A., Hertz T., & Weinshall D. (2004). Computing gaussian mixture models with EM using equivalence constraints, in Advances in Neural Information Processing Systems.

Shi J. & J. Malik, (2000). Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888–905.

Silva V. De & Tenenbaum J. (2003) Global versus local methods in nonlinear dimensionality reduction, in Advances in Neural Information Processing Systems, pp. 721– 728.

Steinbach, M., G. Karypis, & V. Kumar, (1999) A comparison of document clustering techniques, KDD Workshop on Text Mining.

Stig-Erland H. (2007). Solving Classification Problems through Automatic Programming, (Master Thesis, Department of Computer Science, Østfold University College, Halden, Norway)

Szummer M. & Jaakkola T. (2001). Partially labeled classification with Markov random walks,

in Advances in Neural Information Processing Systems, pp. 945–952.

Teh Y., Jordan M., Beal M., & Blei D. (2006). Hierarchical dirichlet processes, Journal of the American Statistical Association, vol. 101, no. 476, pp. 1566–1581.

Vapnik V. (1998). The Nature of Statistical Learning Theory. Wiley-Interscience.

Wagstaff K. & Cardie C. (2000). Clustering with instance-level constraints, in Proceedings of the International Conference on Machine Learning, pp. 1103–1110.

Wagstaff K., Cardie C., Rogers S., & Schroedl S., (2001). Constrained k-means clustering with background knowledge, in Proceedings of the International Conference on Machine Learning, pp. 577–584.

Wang D. (1993). Pattern Recognition: Neural Networks in Perspective. Ohio State University.

Weinberger K., Blitzer J., & Saul L. (2006). Distance metric learning for large margin nearest neighbor classification, in Advances in Neural Information Processing Systems, vol. 18, p. 1473.

Witten I. H. & Frank E. (2005). Data mining: Practical Machine Learning Tools and Techniques (2nd Ed.), Morgan Kaufmann Publishers, San Mateo, CA.

Yang L., Jin R., Sukthankar R., & Liu Y. (2006). An efficient algorithm for local distance metric learning, in Proceedings of the National Conference on Artificial Intelligence, p. 543.

Yang, Y. & Liu X. (1999). A re-examination of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval , Berkeley, US.

Yang, Y. (1999) . An evaluation of statistical approaches to text categorization. Journal of Information Retrieval , Vol. 1, No. 1–2, pp. 69–90.

Yarowsky D. (1995). Unsupervised word sense disambiguation rivalling supervised methods, in Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, p. 189-196.

Yohannes A.(2007). Automatic Amharic text classification. (Master's Thesis, department of information science Addis Ababa university, Ethiopia).

Yu. (2005). Comparative Literary Style Mining between Native and Non-native English Writers. Graduate School of Library and Information Science, University of SIllinois.

Zelalem S. (2001). Automatic classifications of Amharic news items. (Master's Thesis , department of information science Addis Ababa university,Ethiopia).

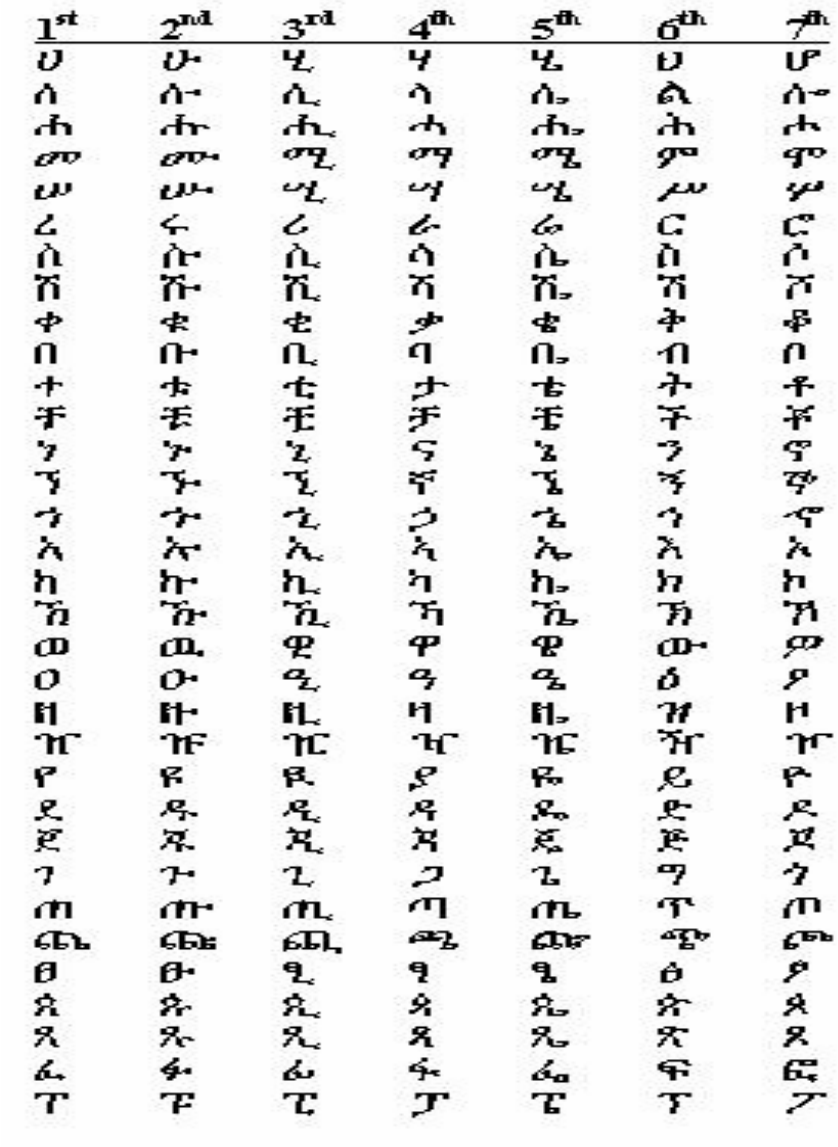
Zhao Q. & Miller D. (2005). Mixture modeling with pairwise, instance-level class constraints, Neural computation, vol. 17, no. 11, pp. 2482–2507.

Zhu & Ghahramani Z. (2002). Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107, Carnegie Mellon University

Zhu X., Ghahramani Z., & Lafferty J. (2003) Semi-supervised learning using gaussian fields and harmonic functions, in Proceedings of the International Conference on Machine Learning, pp. 912–919.

APPENDIXES

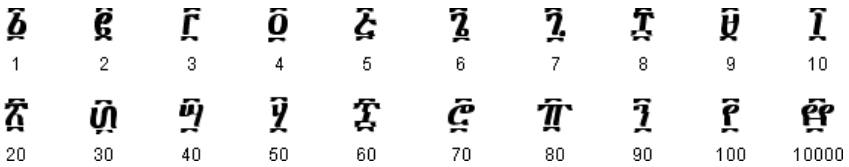
Appendix 1 Amharic alphabets



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

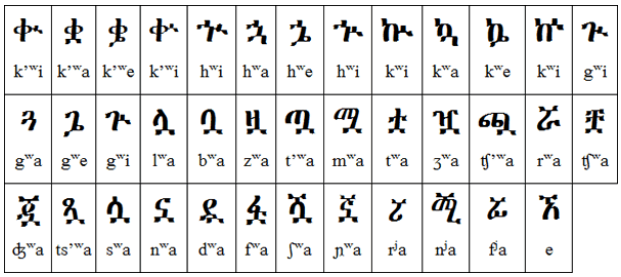
Shows Amharic characters

Appendix 2 Amharic number characters



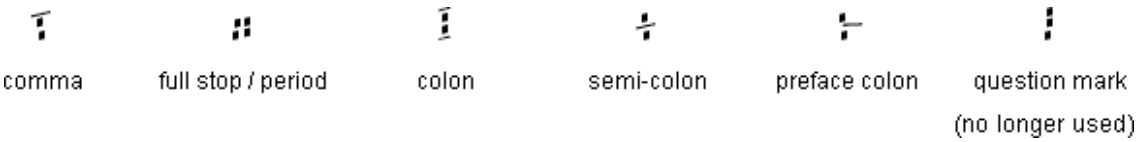
Shows Amharic number characters

Appendix 3 special Amharic characters



Shows special Amharic characters

Appendix 4 Amharic punctuation marks



Shows Amharic punctuation marks

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

BY

ANIMUT BELAY ASRES

JUNE 2012

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

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

BY

ANIMUT BELAY ASRES

JUNE 2012

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A SEMI- SUPERVISED APPROACH FOR AMHARIC NEWS CLASSIFICATION

BY

ANIMUT BELAY ASRES

Name and signature of Members of the examining board:

Name	**Title**	**signature**	**Date**
_____	Chairperson	_____	_____
Solomon Teferra (PhD)	Advisor	_____	_____
_____	Examiner	_____	_____

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university.

_____

Date

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

_____

Advisor

Acknowledgement

Above all, I would like to thank God for his invaluable helps in my entire life span.

Next, I would like to express my sincere appreciation to Dr. Solomon Tefera for advising this thesis. I appreciate the continuous support and timely advice he has given me. His encouragement helped shape the direction of my work. He has continuously encouraged me and bigheartedly guided me on semi-supervised machine learning approach.

I am deeply indebted to Dr. Million Meshesha since he has continuously generously guided and supported me on the fundamentals of semi-supervised machine learning.

I would like to express my gratitude to Ato Erimias. He supported and gave me most of the Amharic news corpus that I used for this research.

I would like to express my gratitude to my best friend Ato Getahun wassie for his valuable suggestions and helpful comments.

I wish to thank my colleagues in Information Science department; especially Ato Tigabu Akal, Girma Debele and Ato Amare Mekonnen from Electrical and computer engineering department supported me and prevent memory problems that I faced by giving their computer.

Finally, I wish to thank my parents specially my mother W/o Tachawt Tadele and my father Ato Belay Asres for their continuous encouragement and support.

List of Acronyms and Abbreviations

SVM	Support Vector Machine
NB	NaiveBayes
K-NN	K-Nearest Neighbor
KDT	Knowledge Discovery in Text
ENA	Ethiopian News Agency
EM	Expectation Maximization
GMM	Gaussian Mixture Model
SSL	Semi-Supervised Learning
RBFN	Radial Basis Function Network
DF	Document Frequency
IDF	Inverse Document Frequency
SMO	Sequential Minimal Optimization
MI	Mutual Information Gain
TS	Term Strength
IG	Information Gain

Table of Contents

List of Tables

List of Figures

List of appendixes

Abstract

Text classification is getting more attention and there is an increasing need for text classification technique that provides automatic, fast, and accurate classification with the least human interaction with such systems. Many techniques of supervised learning and unsupervised learning do exist in the literature for data classification. Semi-supervised learning is halfway between the supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information but not necessarily for all example data.

The paper explored the semi-supervised text classification which is applied to different types of vectors that are generated from the Amharic text documents. 3,154 news articles were used to do this research. To come up with good results document preparation and preprocessing was done. Weka package is used for the classification of the preprocessed data. Machine learning techniques, Expectation maximization clustering algorithm with Naïve Bayes, Hyperpipe, and RBF Network classification algorithm were used to categorize the Amharic news items.

The accuracy of the classifiers was better when the number of classes is less. The best result was obtained by the Naïve Bayes , Hyperpipe and RBF Networks classifiers with four classes (83.44 %, 82.8 and 82.4%) and the least performance is shown on the 10 categories (55.42%,57.26% and 51.9%) respectively. This research indicated that Naïve Bayes is more applicable to semi-supervised categorization of Amharic news items.

Keywords: Text categorization, semi-supervised machine Learning, Naïve Bayes, Hyperpipe and RBF Networks

CHAPTER ONE

INTRODUCTION

1.1. Background

In today's world, communicating with others via internet has become an integral part of life. It is hard to find a college student, professional, or any educated person for that matter, who does not use internet and send or receive e-mails. Also, it is an established fact that a lot of the communication that occurs within companies and organizations is nowadays done by e-mails, rather than memos or common bulletin boards. Modern Information Technologies and Web-based services are faced with the problem of selecting, filtering and managing growing amounts of textual information to which access is usually critical. Information Retrieval (IR) is seen as a suitable methodology for automated management of information/knowledge as it includes several techniques that support an accurate retrieval of information and the consequent user satisfaction. Among others, the classification of electronic documents in general categories (e.g., Sport, Politic, Economy...) is an interesting means to improve the performances of IR systems (Hirotoshi, 2002). It helps users to more easily browse the set of documents of their own interests; sophisticated IR models can also take advantages of the categorized data. Automatic organization of documents has become an important research issue since the explosion of digital and online text information.

From the early 1990s a lot of work has been made in document classification tasks. The effectiveness of many studies has dramatically improved thanks to the introduction of Machine Learning methods into the Text Classification community.

Document classification can be defined as the process of assigning text documents to predefined classes. Text classification can be made manually or automatically. Each of them has advantage and disadvantage to the user (Gebrehiwot, 2011).Two of the most widely-used methods in machine learning for prediction and data analysis are classification and clustering (Duda, 1997).

Classification is a supervised task, where supervision is provided in the form of a set of labeled training data, each data point having a class label selected from a fixed set of classes (Mitchell,

1997). The goal in classification is to learn a function from the training data that gives the best prediction of the class label of unseen (test) data points.

Generative models for classification learn the joint distribution of the data and class variables by assuming a particular parametric form of the underlying distribution that generated the data points in each class, and then apply Bayes Rule to obtain class conditional probabilities that are used to predict the class labels for test points drawn from the same distribution, with unknown class labels (Ng, 2002). In the discriminative framework, the focus is on learning the discriminant function for the class boundaries or a posterior probability for the class labels directly without learning the underlying generative densities (Jaakkola, 1999). It can be shown that the discriminative model of classification has better generalization error than the generative model under certain assumptions (Vapnik, 1998), which has made discriminative classifiers, e.g., support vector machines (Joachims, 1999) and nearest neighbor classifiers (Devroye, 1996), very popular for the classification task.

Clustering is an unsupervised learning problem, which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular similarity metric (Jain, 1988). Here, the learning algorithm just observes a set of points without observing any corresponding class/category labels. Clustering problems can also be categorized as generative or discriminative. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the most general formulation, the number of clusters K is also considered to be an unknown parameter. Such a clustering formulation is called a "model selection" framework, since it has to choose the best value of K under which the clustering model fits the data. In the discriminative clustering setting (e.g., graph-theoretic clustering), the clustering algorithm tries to cluster the data so as to maximize within-cluster similarity and minimize between-cluster similarity based on a particular similarity metric, where it is not necessary to consider an underlying parametric data generation model. In both the generative and discriminative models, clustering algorithms are generally posed as optimization problems and solved by iterative methods like EM (Dempster, 1977), approximation algorithms like KMedian. As the number of clusters and documents increase, the clustering solutions produced by k-means and bisecting k-means become more internally cohesive and externally isolated. However, the

clustering results do not match better with the pre-defined classes and requires relatively high computational requirements (Lakechew, 2011).

Both of them have their own advantages and disadvantages. Supervised algorithms assume that the category structure or hierarchy of a text database is already known. They require a training set of labeled documents and return a function that maps documents to the pre-defined class labels. Knowing the category structure in advance and generation of correctly labeled training set are very challenging or even impossible in large and dynamic text databases.

In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters.

Recently, there has been a lot of interest in the continuum between completely supervised and unsupervised learning (Ghani, 2003). Many document classification tasks are most of all supervised learning problems, since the process learns from pre-classified labeled documents.

1.2. Statement of the Problem

In many practical learning domains (e.g. text processing, bioinformatics), there is a large supply of unlabeled Amharic data but limited labeled Amharic data, which can be expensive to generate. To prevent this problems numerous researches have been conducted such as Zelalem (2001), Surafel (2003), Yohannes (2007), Worku (2009), Alemu (2010), Zeleke (2010) and Lakechew(2011) were conducted. Except the last researcher, who did on unsupervised text categorization all researchers have done on supervised text categorization. Supervised text categorization has the following limitations

First, it can be ambiguous: objects might have non-unique labeling or the labeling themselves may be unreliable due to a disagreement among experts.

Second, it uses limited vocabulary: Typical labeling setting involves selecting a label from a list of pre-specified labels which may not completely or precisely describe an object.

Third, supervised learning algorithms require a large, often excessive, number of labeled training documents for the accurate learning (Kohavi, 1996). Since the application area of automatic text categorization has diversified from articles and web pages to electronic mails and newsgroup postings, it is a difficult task to create training data for each application area (Nigam, 2000).

According to Nigam et al. (Nigam, 2000), in supervised text classification, obtaining training labels is expensive in huge volume of document collection. This is because in supervised text classification labeling of training data is done by a person manually and this is a time consuming, cumbersome and error prone process.

Fourth, Ozgur (Ozgur, 2004) concludes that unsupervised text classification techniques perform better in terms of time complexity and the quality of clusters produced as compared to supervised techniques. This shows that the overall similarities of the clustering solutions obtained by the unsupervised techniques are higher than the supervised ones.

Fifth, supervised text classification algorithms are expensive and time consuming to organize documents in to their categories. As Nigam et al. (Nigam, 2000) suggests, text clustering is a useful and inexpensive way to organize vast text repositories into meaningful topic categories. Furthermore, text clustering offers a low cost alternative to supervised classification, which relies on expensive and difficult handwork to label training data (Massey, 2004).

On the other hand unsupervised learning has the following limitations

First, unsupervised learning is more difficult problem than supervised learning due to the lack of a well-defined user-independent objective. Due to this reason, it is usually considered an ill-posed problem that is exploratory in nature; that is, the users are expected to validate the output of the unsupervised learning process. Devising a fully automatic unsupervised learning algorithm that is applicable in a variety of data settings is an extremely difficult problem, and possibly infeasible (Kang, 2003).

Second, unsupervised learning is less accurate than supervised text classifier. Since unsupervised learning is natural grouping because of nosy data different documents may be classified in the same group.

Third, unsupervised text classification algorithm is presumably the drive to create and apply explicit rules led to increase study and test phase response times when compared with the incidental conditions (Bradley, 2002).

Forth, interestingly, the subjects in the intentional conditions also performed worse in the filler task involving arithmetic problems, possibly indicating increased fatigue or the attempted rehearsal of study-phase items (Bradley, 2002).

Different from the unsupervised techniques, the supervised techniques use class label information in addition to the similarity information between documents. For this reason, it is

expected that the clusters (groups) obtained by the supervised techniques are of higher quality compared to the unsupervised techniques. However, the best performers of the unsupervised techniques k-means and bisecting k-means achieve generally better performance than NaiveBayes and not much worse performance than k-NN, which are supervised techniques, in terms of entropy, purity, overall similarity and F-measure. In the supervised document classification there may be bias or misclassification. Another observation is that, compared with the supervised techniques the unsupervised techniques generally achieve higher overall similarity performance. This is due to the fact that they make decisions depending only on the similarity information between documents. On the other hand the supervised techniques use a labeled training set. This observation has made us think that there may be some outliers in the labeled training set that leads to decrease in the overall similarity of the clusters obtained and unsupervised techniques can be used to enhance the task of pre-defining categories and labeling documents in the training set. Consequently, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. Unlabeled data is available in abundance, but it is difficult to learn the underlying structure of the data. Labeled data is scarce but is easier to learn from. Semi-supervised learning is designed to alleviate the problems of supervised and unsupervised learning problems, and has gained significant interest in the machine learning research community. The framework of semi-supervised approach is applicable to both classification and clustering. Semi-supervised classification algorithms train a classifier given both labeled and unlabeled data.

Most semi-supervised learning algorithms developed to modify existing supervised or unsupervised algorithms, or devise new approaches. Semi-supervised classification has received significant amount of interest, as it provides a way to utilize the large amount of readily available unlabeled data for improving the classifier performance. Semi-supervised classification has been successfully applied to various applications in computer vision and machine learning, such as text classification, human computer interaction, content based image retrieval , object detection, person identification , relevance feedback, computational linguistics and protein categorization, to name a few. Similarly, side-information such as pairwise constraints has been utilized to improve the performance of clustering algorithms by aiding them in arriving at a clustering desired by the user. Semi-supervised learning continues to pose both theoretical and practical questions to researchers in the machine learning. There is also an increasing interest in the fields

of cognitive sciences and human psychology since there are demonstrated settings where humans performed semi-supervised learning (Kang, 2003).

In order to maximize the accuracy, similarity performance, and other things the researcher has explored application of semi-supervised approach for text classification for Amharic documents.

To this end, this study attempts to address the following research questions:

- ✓ Which classification algorithm is best for classifying Amharic text documents?
- ✓ Which clustering algorithm is more suitable for labeling unlabeled Amharic text documents?
- ✓ Which learning approach is more appropriate for creating classification model that helps in Amharic text categorization?
- ✓ To what extent the semi-supervised model is able to predict according to the experts judgment?

1.3. Objective of the study

1.3.1. General Objective

The main objective of this study is to explore automatic Amharic text classification, using both labeled and unlabeled data or supervised and unsupervised machine learning approaches.

1.3.2. Specific objective

The specific objectives are:

- To apply semi-supervised learning approach that improves the existing and established supervised and unsupervised learning algorithms without having to modify them. That is an algorithm that utilizes unlabeled data along with the labeled data while training classifiers.
- To review literature on the concepts, techniques and tools of text classification particularly in the area of semi-supervised learning.
- To select suitable techniques for classifying Amharic text news.
- To design Amharic news text classification architecture
- To design a prototype Amharic text news classifier.
- To evaluate the performances of the prototype Amharic text news classification system.
- To recommend research direction for future work(s) in the area of automatic Amharic text classification.

1.4. Methodology

The methodology that was used for this study includes knowledge discovery in text (KDT) approach that is recommended by Karanikas et al. (Arzucan, 2002). KDT is a multi-step process, which includes all the tasks from the gathering of documents to the visualization of the extracted information. In this research, the three phase KDT process used to achieve the above objectives and it is adapted from the previous researcher.

1.5. Literature review

To get more information about text categorization and the tools and techniques that the researchers used before the researcher referred to text books and also discussion with ENA workers was held.

1.6. Data source and data set preparation

The data sets collected from Ethiopian News Agency (ENA) because in this organization almost all works or classifications are made manually, which is tedious. The researcher used the data collected and used by the previous researchers who did their research in ENA and also collect additional data from ENA. The format of data sets converted in to text for pre processing. The preprocessing had two steps. The first step was removing the irrelevant characters from the corpus. The second step was performing the preprocessing. Both tasks were done using python.

1.7. Design procedures

Categorization of Amharic documents has been developed using three steps. These are preprocessing, clustering and classification. The first task is to preprocess, which includes tokenization, normalization, stop words removal and stemming. The second task is to cluster documents in a structured organization of semi labeled classes.

The idea is to perform a preliminary classification of documents using only the labels associated with the categories and the relationships between classes. This is the unsupervised classification problem. The third task is to classify documents in a structured organization of full labeled classes.

1.8. Tools and Techniques

As it is mentioned, the main objective of this study is to categorize Amharic documents. Text categorization includes clustering and classification of different Amharic documents. This task handled by using weka, because it is easy to use and has a graphical user interface.

1.9. Evaluation Techniques

We need evaluation methods to compare various text classifiers. Evaluation of a classifier can be conducted by measuring its efficiency and its effectiveness. Efficiency is typically measured by using the elapsed processor time and it refers to the ability of a classifier to run fast. Efficiency of a classifier can usually be measured on two dimensions: learning efficiency (i.e., the time a machine learning algorithm takes to generate a classifier from a set of training examples) and categorization efficiency (i.e., the time the classifier takes to assign appropriate categories to a new document). Because of the unstable nature of parameters on which the evaluation depends, efficiency is rarely used as the singular performance measure in text categorization. However, efficiency is important for the practical application of the system.

A much more common evaluation method for text categorization systems is effectiveness: this refers to the ability to take the right decisions on the categorization of new incoming documents. There are several commonly used performance measures of effectiveness. However, there is no agreement on one single measure for use in all applications. Indeed, the type of measure that is preferable depends on the characteristics of the test data set and on the user's interests. The absence of one optimal measure of effectiveness makes it very difficult to compare the relative effectiveness of classifiers. The system has been evaluated based on its efficiency and effectiveness.

1.10. Scope of the study

The scope of this study is limited to investigate the feasibility of designing Amharic news text classification system using semi-supervised approach. In this study, different classification algorithms, such as, Naive Bayes, HyperPipe, RBF Networks and the EM clustering algorithms were used. The study is limited only to classification of text news items from ENA. HTML documents, image documents and others were not considered in this study.

1.11. Significance of the study

Semi-supervised Learning (SSL) takes advantage of a large amount of unlabeled data to enhance classification accuracy. Its application to text categorization is stimulated by the easy availability of an overwhelming number of unannotated documents, in contrast to the limited number of annotated ones. Intuitively, corpora with different topics may not be content wise related, however, word usage exhibits consistent patterns within a language. The main purpose of this work is to address Amharic news classification task. This can be explored in a process that exploits a taxonomic structure, approaching both the unsupervised and the supervised problem. The first goal can be conceived as the activity of finding a hypothesis of the right location of a document taking into account only the structured organization of classes. Then locate each cluster in different categories. This enables the user to find and locate Amharic documents in simple and easy way.

1.12. Thesis Organization

This thesis is organized into six chapters: Chapter 1 - Introduction; Chapter 2- Literature Review; Chapter 3 – The Amharic Language and its Writing system; Chapter 4 - Methodology Chapter-5 Experiment and Performance Evaluation and Chapter 6 – Conclusion and Recommendations.

Chapter one includes background, statement of the problem, objectives of the study, methodology, scope and applications of the study. Chapter 2 discuses different text classification approaches, document preprocessing and representation, overview of the different classification and clustering algorithms and evaluation techniques. Chapter 3 gives highlight about Amharic writing system. Chapter 4 discusses details of methodology adopted and chapter 5 presents the experimental results and findings of the study. In chapter 6 summarizes findings of this study and recommendations are given for further research.

CHAPTER TWO

LITERATURE REVIEW

Introduction

With the ever-increasing volume of text data from various online sources, it is an important task to categorize or classify these text documents into categories that are manageable and easy to understand. Text classification is the task of assigning previously unseen documents to appropriate predefined categories. The task is commonly described as follows: Given a set of labeled training documents of n classes, the system uses this training set to build a classifier, which is then used to classify new documents into the n classes. The problem has been studied extensively in information retrieval, machine learning and natural language processing. The supervised machine learning approach makes this automatic, by learning classifiers from a set of training examples. For most supervised learning algorithms, building accurate classifiers needs a large volume of manually labeled examples. This manual labeling process is time-consuming, expensive, and will have some level of inconsistency (kang, 2003). This problem motivates the researcher work towards a text categorization system that can achieve a satisfactory level of performance with fewer training examples.

Many machine learning algorithms have been developed and applied to the construction of classifiers. They can usually be grouped into rule-based, probability based, and similarity-based learning algorithms. This thesis focuses on the similarity based approach. This builds upon the large volume of previous work in the area of text categorization that has adopted this approach. The similarity-based approach offers the possibility of exploring statistical information that may capture the target concepts hidden in documents (kang, 2003).

2.1. Text Categorization

In this section, the paper gives an overview of text categorization. First defines the text categorization task and discusses ambiguities in most natural languages on which classifiers should be built. Then, discusses two general approaches, “knowledge engineering” and “machine learning”, to the construction of classifiers and why the researcher is focusing on a machine

learning approach. This section also describes characteristics of the domain of text categorization that make this task difficult for a machine learning approach.

2.1.1. A Definition of Text Categorization

Text classification (also known as text categorization) is the automated assignment of natural language text to appropriate thematic categories, based on its content. A set of categories is predefined manually (Arzucan, 2002).

Two different types of text categorization task can be identified depending on the number of categories that could be assigned to each document. The first type, in which exactly one category is assigned to each dj $\in$ D, is regarded as the single-class (or nonoverlapping categories) text categorization task. The second type, in which any number of categories from zero to |C| may be assigned to each dj $\in$D, is called the multi-class (or overlapping categories) task (Sebastiani, 2002). A special type of multi-class text categorization is one where each document is assigned to the same number k, where k > 1, of categories. The answer to the question of which type of text categorization should be adopted for a given text categorization system depends on the application and characteristics of the corpus (kang, 2003).

Most semi-supervised learning methods are extensions of existing supervised and unsupervised algorithms. Therefore, before introducing the developments in semi-supervised learning literature, it is useful to briefly review supervised and unsupervised learning approaches.

2.1.2. Ambiguities in Natural Language Text

In most content-based text classification systems, an important issue is how they can capture the meaning of the natural language texts. Obtaining accurate classifiers requires the system to understand the natural languages at some level. Understanding natural languages, however, is a difficult task due to ambiguities in them:

1. The same sentence may have different meanings. For example, consider a sentence like "Salespeople sold the dog biscuits" (an example from Charniak, 2003). This sentence can be interpreted in two different ways: (1) the salespeople are selling the dog-biscuits and (2) the salespeople are selling biscuits to dogs.
2. There is the large number of synonyms – syntactically different words with the same or similar meanings in natural languages. It is regarded as good writing style not to

repeatedly use the same word for expressing a particular idea (or concept). Synonyms allow the same idea to be expressed by different words that have a similar meaning.

3. Polysemy refers to an ambiguity where words which are spelled the same can have different meanings in different sentences or documents. For example, the word "bat" may mean (1) an implement used in sports to hit the ball or (2) a flying mammal.

Resolving such ambiguities is probably beneficial to text categorization when there are many words in common across categories, even though it may not have a huge impact on the overall text categorization performance.

2.1.3. Knowledge Engineering versus Machine Learning Approach

There are two different ways of constructing classifiers, the function: $D \times C \{T, F\}$. They are "knowledge engineering" and "machine learning" approaches. In the knowledge engineering approach, human experts (including knowledge engineers and domain experts) manually create a set of rules that correctly categorize previously unseen documents under given categories. While allowing for semantically-oriented text categorization, by defining controlled vocabularies which can be interpreted by the text categorization system (Brasethvik, 2001), manually determining such a solution imposes a considerable workload on human experts. This makes it time consuming and expensive.

Also, this manual approach may cause inconsistency since human experts often disagree on the assigned categories of documents and even one person may categorize documents inconsistently (Apte, 1994). As a result, these problems for the knowledge engineering approach cause the bottleneck of encoding large amounts of incomplete and potentially conflicting expert knowledge.

The machine learning approach to text categorization is to automatically build the classifiers by learning the concept descriptions of the categories. One type of machine learning, applied to text categorization, is "supervised learning". This requires a set of pre-labeled (pre-categorized) training documents for generating classifiers. By contrast, "unsupervised learning" refers to the task of automatically identifying a set of categories from a set of unlabeled documents and grouping these unlabeled documents under these identified categories (Merkl, 1998).

The advantages of the machine learning approach over knowledge engineering are the considerable reduction in the volume of work required from human experts, consistent text categorization, and the capability of easily adjusting the generated classifier to handle different types of documents (such as newspaper articles, newsgroup postings, electronic mails, etc.) and even languages other than English.

2.1.4. Difficulties for the Machine Learning Approach

The unstructured format of natural language text and the diversity of target concepts associated with the categories, present interesting challenges to the content based application of machine learning algorithms. The large number of input features, that seem necessary for the construction of classifiers, overwhelms most text categorization systems. For most machine learning algorithms, increasing the number of features means that they have to use more training examples to obtain the same level of text categorization performance. This large number of training examples and features may be computationally intractable for most machine learning algorithms, by requiring unacceptably large processing time and memory.(kang, 2003)

Of the large number of features, there are usually many features that appear in most documents. These words can be considered irrelevant, in the sense that such features are evenly distributed throughout documents and, as a result, have no discriminating power. It is important for the efficiency and effectiveness of the system to select an efficient subset of features, by removing these irrelevant ones. However, it is a difficult task since a reasonable feature subset size might be different across the categories and some informative features for a given category could be distributed across several categories. For example, depending on the level of concept complexity, some categories require a large number of features to describe their concepts while others need a relatively small number of features. Also, informative features in the overlapping categories might be evenly distributed across such overlapping categories and could be considered as irrelevant ones.(kang, 2003)

2.2. Text categorization approaches

2.2.1. Supervised learning

Supervised learning aims to learn a mapping function f: X → Y, where X and Y are input and output spaces, respectively (e.g. classification and regression (Duda, 2000). The process of learning the mapping function is called training and the set of labeled objects used is called the training data or the training set. The mapping, once learned, can be used to predict the labels of the objects that were not seen during the training phase. Several pattern recognition (Duda, 2000) and machine learning (Mitchell, 1998) textbooks discuss supervised learning extensively. A brief overview of supervised learning algorithms is presented in this section.

Supervised learning methods can be broadly divided into generative or discriminative approaches. Generative models assume that the data is independently and identically distributed and is generated by a parameterized probability density function. The parameters are estimated using methods like the Maximum Likelihood Estimation (MLE), Maximum A Posteriori estimation (MAP) (Duda, 2000), Empirical Bayes and Variational Bayes (Bishop, 2006). Probabilistic methods could further be divided into frequentst or Bayesian. Frequentst methods estimate parameters based on the observed data alone, while Bayesian methods allow for inclusion of prior knowledge about the unknown parameters. Examples of this approach include the Naive Bayes classifier, Bayesian linear and quadratic discriminants to name a few.

Instead of modeling the data generation process, discriminative methods directly model the decision boundary between the classes. The decision boundary is represented as a parametric function of data, and the parameters are learned by minimizing the classification error on the training set (Duda, 2000). Empirical Risk Minimization (ERM) is a widely adopted principle in discriminative supervised learning. This is largely the approach taken by Neural Networks (Bishop, 2005) and Logistic Regression (Bishop, 2006). As opposed to probabilistic methods, these do not assume any specific distribution on the generation of data, but model the decision boundary directly.

Most methods following the ERM principle suffer from poor generalization performance. This was overcome by Vapnik's (kang, 2003) Structural Risk Minimization (SRM) principle which adds a regularity criterion to the empirical risk that selects a classifier with good generalization

ability. This led to the development of Support Vector Machines (SVMs) which regularize the complexity of classifiers while simultaneously minimizing the empirical error. Methods following ERM such as neural networks and Logistic Regression are extended to their regularized versions that follow SRM (Bishop, 2006).

2.2.1.1. Classification Algorithms

Supervised algorithms assume that the category structure or hierarchy of a text database is already known. They require a training set of labeled documents and return a function that maps documents to the pre-defined class labels. As discussed previously, knowing the category structure in advance and generation of correctly labeled training set are very challenging or even impossible in large and dynamic text databases.

A wide range of classification algorithms have been developed through time with different underlying models and different theories of how a classifier should be built. These algorithms have different inductive biases that affect their performance on a data set, and consequently, it is important to find the inductive bias that best fits the data set. This can be done empirically by applying a set of different machine learning algorithms and selecting the algorithm that performs the best(Stig-Erland, 2007).

In this section the paper discuss the most popular supervised algorithms.

2.2.1.1.1. Bayes

Bayesian algorithms are based on Bayes' Theorem, which is defined as

$$P(h|d) = \frac{P(d|h)Pr(h)}{P(d)} \quad \text{.....equation 1}$$

where the h corresponds to a hypothesis, namely a prediction of a particular class, and the d represents the attributes of the unlabeled instance.

Naive Bayes

The naive Bayes (NB) classifier is a probabilistic model that uses the joint probabilities of terms and categories to estimate the probabilities of categories given a test document (Mitchell, 1999). The naive part of the classifier comes from the simplifying assumption that all terms are conditionally independent of each other given a category. Because of this independence

assumption, the parameters for each term can be learned separately and this simplifies and speeds the computation operations compared to non-naive Bayes classifiers.

There are two common event models for NB text classification, discussed by McCallum and Nigam , multinomial model and multivariate Bernoulli model. In both models classification of test documents is performed by applying the Bayes' rule (Mitchell, 1999):

$$P(cj|jdi) = \frac{P(cj) \cdot P(dij|cj)}{P(di)} \quad \text{……equation 2}$$

where di is a test document and cj is a category. The posterior probability of each category cj given the test document di, i.e. P(cj|di), is calculated and the category with the highest probability is assigned to di. In order to calculate P(cj|di), P(cj) and P(di|cj) have to be estimated from the training set of documents. Note that P(di) is same for each category so we can eliminate it from the computation.

Multinomial Model

In the multinomial model a document di is an ordered sequence of term events, drawn from the term space T. The naive Bayes assumption is that the probability of each term event is independent of term's context, position in the document, and length of the document. So, each document di is drawn from a multinomial distribution of terms with number of independent trials equal to the length of di(kang, 2003).

Multivariate Bernoulli Model

Multivariate Bernoulli model for naive Bayes classification is the event model. In this model a document is represented by a vector of binary features indicating the terms that occur and that do not occur in the document.

Here, the document is the event and absence or presences of terms are the attributes of the event. The naive Bayes assumption is that the probability of each term being present in a document is independent of the presence of other terms in a document.

To state differently, the absence or presence of each term is dependent only on the category of the document.

Different from the multinomial model, the multivariate Bernoulli model does not take into account the number of times each term occurs in the document, and it explicitly includes the non-occurrence probability of terms that are absent in the document (McCallum,1998).

2.2.1.1.2. Lazy

Lazy learning algorithms differ from the other classification algorithms in that the training of the classifier is postponed until classification. This allows the classifier to be customized according to each unlabeled instance at the expense of being computational intensive if there are many instances to classify(Stig-Erland, 2007).

IB1

IB1 is a nearest neighbour algorithm that determines the class of an unlabeled instance according to the class of the nearest training instance. The distance between two instances are calculated using the euclidean distance(Stig-Erland, 2007).

$$\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$ ……equation 3

where ai and bi are the attributes i of the instances a and b.

IBk

IBk is similar to IB1, but it uses the k nearest neighbours instead of only one. The predicted class is determined by the majority vote where each instance places a vote on its corresponding class(Schmitter, 2006).

K*

K* is a nearest neighbour algorithm that employs, instead the Euclidean distance, an entropic distance function computing the probability of randomly transforming one instance into another. Each class receives a vote from each instance with a weight equal to the distance from it to the unlabeled instance, and the class with the most votes is selected (Liszka,1999).

Locally Weighted Learning

Locally weighted learning (LWL) selects a subset of the training instances, where each instance is weighted according to the unlabeled instance.

A k nearest neighbour algorithm is applied to select the subset of instances, and the weight is calculated by a weighting function taking the euclidean distance as input.

2.2.1.1.3. Functions

These algorithms have mathematical or statistical foundations and create models that can be represented mathematically through functions. They are a mix of regression and classification algorithms (Stig-Erland, 2007).

Linear Regression

Linear regression (LinReg) is a standard linear regression algorithm that expresses the numerical class as a linear combination of the attributes. The coefficients of these attributes are calculated using the least-square method (Stig-Erland, 2007).

Logistic

Logistic builds logistic regression models and is implemented according to with some modifications. These models have similar properties to linear regression models, but the target attribute is transformed using the logit function and the weights are found by maximizing the log-likelihood instead of minimizing the sum of squared errors (Wang, 1993).

Simple Logistic

Simple Logistic (SLogistic) also builds logistic regression models, but it uses another strategy than Logistic involving LogitBoost and a base learner constructing simple regression models containing only the attribute yielding the minimum squared error. The number of boosting iterations used is determined by cross-validation (Witten, 2005).

Multilayer Perceptron

Multilayer Perceptron (MP) is a neural network algorithm that optimizes the weights of neural network using back propagation. The input layer comprises a bias node in addition to a node for each attribute after the nominal attributes have been converted to binary attributes. The hidden layer also contains a bias node in addition to n nodes determined by the following expression

$$\frac{i+o}{2}$$ ……equation 4

where i and o is the number of nodes in the input and output layer. The output layer is composed of a node for each class (Stig-Erland, 2007). The activation function for the nodes in the hidden and output layer is the sigmoid function.

RBF Network

RBF Network (RBFN) trains a radial basis function network, which is a type neural network. The network has three layers: an input layer with a node for each attribute; a hidden layer where each node has a Gaussian radial basis function as activation function, created using a clustering method called KMeans (Martin, 1995); and an output layer containing a node for each class with sigmoid as activation function.

SMO

SMO, proposed by John Platt, is a sequential minimum optimization algorithm for training support vector machines (SVM). The algorithm finds the maximum margin hyperplane represented as a set of vectors known as support vectors. In order to solve non-linear problems with this linear classifier, the instance space is transformed using a non-linear kernel function.
We chose to use the default polynomial kernel (Stig-Erland, 2007).

Support Vector Machines

Support Vector Machines (SVM) is a technique introduced by Vapnik in 1995, which is based on the Structural Risk Minimization principle. It is designed for solving two-class pattern recognition problems. The problem is to find the decision surface that separates the positive and negative training examples of a category with maximum margin.

For the linearly separable case, the decision surface is a hyperplane that can be written as (Yang, 1999):

$w * d + b = 0$ ……equation 5

where d is a document to be classified, and vector w and constant b are learned from the training set. The SVM problem is to find w and b that satisfy the following constraints (Joachims, 1998):

Minimize $\|w\|^2$

so that $\forall i : yi(w * d + b) \geq 1$……equation 6

Here, $i \in \{1, 2 \ldots N\}$, where N is the number of documents in the training set; and y_i equals +1 if document di is a positive example for the category being considered and equals -1 otherwise.

Most of the classifiers implicitly or explicitly require the data to be represented as a vector in a suitable vector space, and are not directly applicable to nominal and ordinal features (Tan, 2005). Also, most discriminative classifiers have been developed for only two classes. Multiclass classifiers are realized by combining multiple binary (2-class) classifiers, or using coding methods (Cohen, 1996).

Voted Perceptron

Voted perceptron (VP) (Freund, 1998) transforms the input space using a polynomial kernel as SMO, but it uses the perceptron (Rosenblatt, 1988) algorithm to train the classifier.

During training, it stores all the intermediate prediction vectors, namely the coefficients of the attributes, along with a weight of how many iterations they persisted without change. When classifying, each prediction vector votes on a class according to its weight, and the majority vote determines the predicted class.

2.2.1.1.4. Trees

These algorithms induce decision trees as classifiers, which basically contains two types of nodes: decision nodes and leaf nodes. Decision nodes are internal nodes containing a test on a specific attribute that determines which of the underlying branches an unlabeled instance should follow. Traversal continues from the root until a leaf node is encountered, and the leaf node predicts the class of the instance by utilizing a prediction function.

Decision trees is one of the earliest classifier (Sahami, 1998), that can handle handle a variety of data with a mix of both real, nominal, missing features and multiple classes. It also provides interpretable classifiers, which give a user an insight about which features are contributing for a particular class being predicted for a given input example. Decision trees could produce complex decision rules, and are sensitive to noise in the data. Their complexity can be controlled by using approaches like pruning; however, in practice classifiers like SVM or Nearest Neighbor have been shown to outperform decision trees on vector data.

Decision tree learning is composed of building and pruning. A decision tree is typically built by recursively selecting the most promising attribute and splitting the training set accordingly until

all instances belong to the same class or all attributes have already been used. The most promising attribute is determined by the attribute maximizing the splitting criterion.

The role of pruning is to simplify the decision tree either during or after building(Stig-Erland, 2007).

ID3

ID3 is one of the first decision tree learners proposed, and it employs information gain as splitting criterion. Since it does not support continuous or missing attribute values, it can only solve a limited set of problems (Quinlan, 1986).

J4.8

J4.8 is an implementation of Quinlan's popular C4.5 (Quinlan, 1993) decision tree learner and it improves upon ID3 in several areas. First, it replaces the information gain splitting criterion with gain ratio since information gain favors attributes with many values. Second, it supports both continuous and missing values, and it performs pruning using error based pruning (EBP).

REPTree

REPTree is a fast decision tree learner. it uses information gain instead of gain ratio and reduce error pruning instead of EBP.

NBTree

NBTree is a hybrid algorithm that creates decision trees with Naive Bayes classifiers at the leaves learned from the training instances reaching the node. It follows the standard decision tree learning algorithm and uses the mean accuracy of creating a Naive Bayes classifier at a given node according to 10-fold cross-validation as splitting criterion(Kohavi, 1996).

Logistic Model Trees

Logistic Model Trees (LMT) (Landwehr, 2005) builds decision trees with logistic regression models at the leaves, which are iteratively created using Simple Logistic. The trees are built similarly to C4.5 by selecting attributes according to the gain ratio splitting criterion until there are no more attributes, all the instances have the same class or there are less than 16 instances. Pruning is performed using the pruning algorithm employed by the decision tree learner, CART (Breiman, 1984).

M5'

M5' (E. Frank, 1998) is a reconstruction of Quinlan's M5 (Quinlan, 1992) that creates decision trees with linear regression models at the leaves. It chooses the attribute at each decision node

that maximizes the standard deviation reduction of the class of the training instances reaching the node. When the tree is built, it traverses upwards from the leaves, while adding linear regression models at the nodes and possibly removing nodes if necessary. The predicted class value of an unlabeled instance is determined based on the output of all the linear regression models encountered when traversing the tree.

Decision Stump

Decision Stump (DS) induces simple decision trees, known as decision stumps, with only a single decision node. This node has a boolean test, which for a nominal attribute tests whether the attribute is equal to a specific value and for a continuous attribute tests whether the attribute is less or equal to a threshold. This algorithm is normally executed through ensemble algorithms like bagging and boosting (Stig-Erland, 2007).

Random Forest

Random Forest (RF) (Breiman, 2001) uses bagging in combination with a random tree inducer. The random tree inducer builds a tree by choosing at a given node the best attribute among a set of randomly selected attributes.

ADTree

ADTree (Freund, 1999) creates what is known as an alternative decision tree by using boosting to add the different branches. An alternative decision tree is simply a set of interconnected decision stumps with numerical leaves, where each leaf may be connected to a set of other stumps.

The tree is used to classify unlabeled instances with binary classes by summing all the numerical nodes encountered while following the different paths of the tree applicable for the instances. The sign of this value determines the predicted class.

2.2.1.1.5. Rules

This group contains algorithms that create classifiers which are rule sets. Rule sets are intuitive and easier for humans to interpret than other classifiers like decision trees.

JRip

JRIP is an implementation of RIPPER (Cohen, 1995) with some minor modifications added to fix what appear to be two bugs in the original algorithm. It induces each rule of the final rule set in two steps. Firstly, the rule is grown by continually adding antecedents until it matches only

training instances with a specific class. Secondly, the rule is iteratively pruned by processing the antecedents in reverse order.

OneR

OneR is a simple algorithm that creates a rule set for each attribute and chooses the rule set with the lowest error rate on the training data. Each rule set comprises a rule for each value of a particular attribute that predicts the majority class of the training instances matching the rule(Stig-Erland, 2007).

ZeroR

ZeroR is the simplest of all classification algorithms, and it only predicts the majority class of the training set. This algorithm provides an upper bound of the error rate that all other classification algorithms should be smaller than (Stig-Erland, 2007).

DecisionTable

DecisionTable (DT) (Kohavi, 1995) constructs a decision table classifier, which simply a table is containing the training instances with only a subset of their attributes included. The optimal subset of the attributes is found using best-first search combined with cross-validation where the DecisionTable algorithm is executed for different subsets. An unlabeled instance is classified as the majority class of the matching instances in the table, but if there are no matching instances, the majority class of all training instances is predicted instead.

PART

PART (Frank, 1998) creates a rule set by repeatably creating pruned decision trees using J4.8, converting them to rules and removing the training instances matching the rule until all training instances are covered by at least one rule.

Each rule is created according to the path from the root of the decision tree to leaf covering the most training instances. In order to preserve computational resources, only partial decision trees are constructed where branches are expanded as needed.

M5Rules

M5Rules (Holmes, 1999) builds regression rules using the same algorithm as Part except it generates trees using M5' instead of J4.8.

Ridor

Ridor is a RIpple DOwn Rule learner that first creates a default rule predicting the majority class of the training instances and then recursively adds exceptions to this rule until all training instances are classified correctly according to the rule set. A separate validation set is utilized to find the most accurate exception at each step (Stig-Erland, 2007).

NNge

NNge is a nearest neighbour algorithm forming non-nested general exemplars. A general exemplar is a hyper-rectangle that encompasses a set of training instances sharing the same class. In this way, each general exemplar is like a rule, and the nearest exemplar determines the class of an unlabeled instance(Stig-Erland, 2007).

2.2.1.1.6. Misc

This group contains the algorithms that do not fit naturally into any of the other groups.

HyperPipes

HyperPipes (HP) is a simple and extremely fast classification algorithm that constructs a set of attribute ranges for each class. For nominal attributes, the range is the set of values observed for a particular attribute of the training instances matching a specific class. The range is found similarly for continuous attributes except the range is not a subset, but an interval ranging from the minimum to the maximum observed attribute value. Classification is performed by selecting the class with the most matching attribute ranges (Stig-Erland, 2007).

VFI

VFI (G. Demiroz , 1997) constructs a set of intervals for each attribute similarly to Hyper- Pipes, but these intervals are not bound to a specific class. Thus, each interval contains a class count for each class according to the training instances that fall into it. Continuous attributes are basically

discretised into a set of intervals, and an interval for nominal attributes is defined as a single attribute value. An unlabeled instance is classified using the majority vote, where each matching attribute interval is allowed to vote.

2.2.1.1.7. Ensemble

Ensemble classifiers are meta-classification algorithms that combine multiple component classifiers (called base classifiers) to obtain a meta-classifier with the hope that they will perform better than any of the individual component classifiers. Bagging (Breiman, 1996) and Boosting (Freund, 1996) are the two most popular methods in this class. Bagging is a short form for bootstrap aggregation, which trains multiple instances of a classifier on different subsamples (bootstrap samples) of the training data. The decision on an unseen test example is taken by a majority vote among the base classifiers. Boosting, on the other hand, samples training data more intelligently by sampling examples that are difficult for the existing ensemble to classify with a higher preference.

Ensemble algorithms use a base learning algorithm to create an ensemble of classifiers and combine these classifiers to reach a prediction. These algorithms differ in how the base learning algorithm is applied and how they combine the classifiers. The first two algorithms enhance the abilities of the base learner, making it possible to solve previously unsupported problems, while the last two enhance the performance of the base learning algorithm(Stig-Erland, 2007).

ClassificationViaRegression

ClassificationViaRegression allows a regression algorithm to solve classification problems. It creates a data set for each class using a 1-against-all encoding where the class is 1 if it is equal to the current class and 0 otherwise. A regression model is created for each class based on these data sets, and classification is performed by predicting the class belonging to the model yielding the greatest value (Stig-Erland, 2007).

MultiClassClassifier

MultiClassClassifier makes it possible to solve multi-class problems with algorithms that only support binary classes. This is possible through several methods, but we chose the default 1-against-all method explained in the previous section.

Bagging

Bagging (Breiman, 1996) creates an ensemble of classifiers in order to increase the accuracy by stabilizing the base learning algorithm, or in other words decrease its variance. This is done by generating a set of ”new” training sets using bootstrapping and applying the base learning algorithm on these data sets.

Prediction is determined by the majority vote of the ensemble. The success of bagging depends heavily on the properties of the base learning algorithm. It should be unstable, meaning that it is sensitive to small changes in the training set, so that its variance can be decreased.

AdaBoost.M1

AdaBoost.M1 (Freund, 1996) is a boosting algorithm that builds an ensemble of classifiers by forcing the base learning algorithm to focus on the instances that the previous classifiers had problems classifying correctly. This is done by accompanying every training instance with a weight representing the severity of misclassification such that the error rate is calculated as the sum of the weights of the misclassified instances divided by the sum of all the weights.

Initially, each instance has equal weights, but after a new classifier is induced, the weights are updated so that misclassified instances increase in weight while the other instances decrease.

K -Nearest Neighbor Classification

K-NN (k-nearest neighbor) classification is a popular instance-based learning method (Mitchell) that has been shown to be a strong performer in the task of text categorization (Yang, 1999).

The algorithm works as follows: First, given a test document x, the k nearest neighbors among the training documents are found. The category labels of these neighbors are used to estimate the category of the test document. In the traditional approach, the most common category label among the k-nearest neighbors is assigned to the test document.

Weighted k-NN is a refinement to the traditional approach. In weighted k-NN, the contribution of each of the k nearest neighbors is weighted according to its similarity to the test document x. Then, for each category, the similarities of the neighbors belonging to that category are summed

to obtain the score of the category for x. That is, the score of category cj for the test document x is(Arzucan, 2002)

$$score(c_j, \mathbf{x}) = \sum_{\mathbf{d_i} \in N(\mathbf{x})} cos(\mathbf{x}, \mathbf{d_i}) \cdot y(\mathbf{d_i}, c_j)$$ .....equation 7

where di is a training document; N(x) is the set of the k training documents nearest to x; cos(x; di) is the cosine similarity between the test document x and the training document di; and y(di; cj) is a function whose value is 1 if di belongs to category cj and 0 otherwise. The test document x is assigned to the category with the highest score.

2.2.2. Unsupervised learning

Unsupervised learning or clustering is a significantly more difficult problem than classification because of the absence of labels on the training data. Given a set of objects, or a set of pair wise similarities between the objects, the goal of clustering is to find natural groupings (clusters) in the data. The mathematical definition of what is considered a natural grouping defines the clustering algorithm. A very large number of clustering algorithms have already been published, and new ones continue to appear (Jain, 1999). There are different clustering algorithms and the following are a few representative.

Parametric mixture models are well known in statistics and machine learning communities (McLachlan, 1987). A mixture of parametric distributions, in particular, GMM (McLachlan, 2000) has been extensively used for clustering. GMMs are limited by the assumption that each component is homogeneous, unimodal, and generated using a Gaussian density. Latent Dirichlet Allocation (Blei, 2003) is a multinomial mixture model that has become the de facto standard for text clustering.

Several mixture models have been extended to their non-parametric form by taking the number of components to infinity in the limit (Teh, 2006). A non-parametric prior is used in the generative process of these infinite models (e.g. Dirichlet Process) for clustering in (Teh, 2006). One of the key advantages offered by the non-parametric prior based approaches is that they adjust their complexity to fit the data by choosing the appropriate number of parametric components. Hierarchical Topic Models (Blei, 2004) are clustering approaches that have seen huge success in clustering text data.

Kernel K-means is a related kernel based algorithm, which generalizes the Euclidean distance based K-means to arbitrary metrics in the feature space. Using the kernel trick, the data is first mapped into a higher dimensional space using a possibly non-linear map, and a K-means clustering is performed in the higher dimensional space.

Non-parametric density based methods are popular in the data mining community.

Mean-shift clustering (Comaniciu, 2002) is a widely used non-parametric density based clustering algorithm. The objective of Mean-shift is to identify the modes in the kernel-density, seeking the nearest mode for each point in the input space. Several density based methods like DBSCAN also rely on empirical probability estimates, but their performance degrades heavily when the data is high dimensional. A recent segmentation algorithm (Andreetto, 2007) uses a hybrid mixture model, where each mixture component is a convex combination of a parametric and non-parametric density estimates.

Hierarchical clustering algorithms are popular non-parametric algorithms that iteratively build a cluster tree from a given pairwise similarity matrix. Agglomerative algorithms such as Single Link, Complete Link, Average Link (Jain 1988), Bayesian Hierarchical Clustering (Heller, 2005), start with each data point in a single cluster, and merge them successively into larger clusters based on different similarity criteria at each iteration. Divisive algorithms start with a single cluster, and successively divide the clusters at each iteration.

2.2.2.1. Unsupervised Techniques for Document Clustering

In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters. Traditional clustering techniques can be categorized into two major groups as partitional and hierarchical.

Partitional Clustering Techniques

Partitional algorithms produce un-nested, non-overlapping partitions of documents that usually locally optimize a clustering criterion. The general methodology is as follows: given the number of clusters k, an initial partition is constructed; next the clustering solution is refined iteratively by moving documents from one cluster to another.

The following sub-sections discusses the most popular partitional algorithm k-means, and its variant bisecting k-means which has been applied to cluster documents by Steinbach et al. (Steinbach, 1999) and has been shown to generally outperform agglomerative hierarchical algorithms.

Expectation maximization

The theoretical basis for expectation-maximization shows that with sufficiently large amounts of unlabeled data generated by the model class in question, a more probable model can be found than if using just the labeled data alone. If the classification task is to predict the latent variable of the generative model, then with sufficient data a more probable model will also result in a more accurate classifier. Here, expectation-maximization finds more likely models and improved classification accuracy. Expectation-maximization (EM) is used to fit the mixture model to the negative examples(Olivier, 2006).
It uses the k-medoids which is similar to K-means except it takes one member of the cluster as a centriod. It uses real contexts word from the dataset as a basis for clustering. K-mean takes the round space (not a member) to make centriod

K-Means Clustering

The idea behind the k-means algorithm, discussed by Hartigan (Hartigan, 1975), is that each of k clusters can be represented by the mean of the documents assigned to that cluster, which is called the centroid of that cluster.

K-means (Cohen, 1996), (Yang, 1999), arguably, is the most popular and widely used clustering algorithm. K-means is an example of a sum of squared error (SSE) minimization algorithm. Each cluster is represented by its centroid. The goal of K-means is to find the centroids and the cluster labels for the data points such that the sum-of-squared error between each data point and its

closest centroid is minimized. K-means is initialized with a set of random cluster centers, that are iteratively updated by assigning the closest data point to each center, and recomputing the centroids. ISODATA (Hartigan, J., 1975) and Linear Vector Quantization (Berkhin, 2002) are closely related SSE minimization algorithms that are independently proposed in different disciplines.

It is discussed by Berkhin (Berkhin, 2002) that there are two versions of k-means algorithm known. The first version is the batch version and is also known as Forgy's algorithm (Forgy, 1965). It consists of the following two-step major iterations:

(1) Reassign all the documents to their nearest centroids

(2) Recompute centroids of newly assembled groups

Before the iterations start, firstly k documents are selected as the initial centroids.

Iterations continue until a stopping criterion such as no reassignments occur is achieved.

Initially, k documents from the corpus are selected randomly as the initial centroids. Then, iteratively documents are assigned to their nearest centroid and centroids are updated incrementally, i.e., after each assignment of a document to its nearest centroid. Iterations stop, when no reassignments of documents occur.

The centroid vector c of cluster C of documents is define as follows(Arzucan, 2002):

$$\mathbf{c} = \frac{\sum_{d \in C} \mathbf{d}}{|C|}$$

……equation 8

So, c is obtained by averaging the weights of the terms of the documents in C. Analogously, the similarity between a document d and a centroid vector c by cosine similarity measure defined as (Arzucan, 2002)

$$cos(\mathbf{d}, \mathbf{c}) = \frac{\mathbf{d} \bullet \mathbf{c}}{\|\mathbf{d}\| \|\mathbf{c}\|}$$

……equation 9

Note that although documents are of unit length, centroid vectors are not necessarily of unit length.

Bisecting K-Means

Although bisecting k-means is actually a divisive clustering algorithm that achieves a hierarchy of clusters by repeatedly applying the basic k-means algorithm, the researcher discuss it in this section as it is a variant of k-means.

In each step of bisecting k-means a cluster is selected to be split and it is split into two by applying basic k-means for k = 2. The largest cluster, that is the cluster containing the maximum number of documents, or the cluster with the least overall similarity can be chosen to be split.

Hierarchical Clustering Techniques

Hierarchical clustering algorithms produce a cluster hierarchy named a dendrogram (Berkhin, 2002). These algorithms can be categorized as divisive (top-down) and agglomerative (bottom-up) (Jain, 1999) (Berkhin, 2002). We discuss these approaches in the following sub-sections.

Divisive Hierarchical Clustering

Divisive algorithms start with one cluster of all documents and at each iteration split the most appropriate cluster until a stopping criterion such as a requested number k of clusters is achieved.

A method to implement a divisive hierarchical algorithm is described by Kaufman and Rousseeuw. In this technique in each step the cluster with the largest diameter is split, i.e. the cluster containing the most distant pair of documents. As we use document similarity instead of distance as a proximity measure, the cluster to be split is the one containing the least similar pair of documents. Within this cluster the document with the least average similarity to the other documents is removed to form a new singleton cluster. The algorithm proceeds by iteratively assigning the documents in the cluster being split to the new cluster if they have greater average similarity to the documents in the new cluster (Kang, 2003).

Agglomerative Hierarchical Clustering

Agglomerative clustering algorithms start with each document in a separate cluster and at each iteration merge the most similar clusters until the stopping criterion is met. They are mainly

categorized as single-link, complete-link and average-link depending on the method they define inter-cluster similarity.

Single-link The single-link method defines the similarity of two clusters C_i and C_j as the similarity of the two most similar documents $d_i \in C_i$ and $d_j \in C_j$ (Arzucan, 2002):

$$similarity_{single-link}(C_i, C_j) = \max_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})|$$

Complete-link The complete-link method defines the similarity of two clusters

C_i and C_j as the similarity of the two least similar documents $d_i \in C_i$ and $d_j \in C_j$ (Arzucan, 2002):

$$similarity_{complete-link}(C_i, C_j) = \min_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})| \quad \text{.....equation 10}$$

Average-link The average-link method defines the similarity of two clusters

C_i and C_j as the average of the pairwise similarities of the documents from each cluster (Arzucan, 2002):

$$similarity_{average-link}(C_i, C_j) = \frac{\sum_{\mathbf{d_i} \in C_i, \mathbf{d_j} \in C_j} |cos(\mathbf{d_i}, \mathbf{d_j})|}{n_i n_j} \quad \text{.....equation 11}$$

where n_i and n_j are sizes of clusters C_i and C_j respectively.

2.2.3. Semi-supervised learning

Semi-supervised learning algorithms can be broadly classified based on the role the available side information plays in providing the solution to supervised or unsupervised learning.

2.2.3.1. Semi-supervised classification

While semi-supervised classification is a relatively new research area, the idea of using unlabeled samples to augment labeled examples for prediction was conceived several decades ago.

The initial work in semi-supervised learning is attributed to Scudders for his work on "selflearning". An earlier work by Robbins and Monro on sequential learning can also be viewed

as related to semi-supervised learning. Vapnik's Overall Risk Minimization (ORM) principle advocates minimizing the risk over the labeled training data as well as the unlabelled data, as opposed to the Empirical Risk Minimization, and resulted in transductive Support Vector Machines (Pavan, 2010).

Given a set of labeled data, a decision boundary may be learned using any of the supervised learning methods. When a large number of unlabeled data is provided in addition to the labeled data, the true structure of each class is revealed through the distribution of the unlabeled data. The unlabeled data defines a "natural region" for each class, and the region is labeled by the labeled data. The task now is no longer just limited to separating the labeled data, but to separate the regions to which the labeled data belong. The definition of this "region" constitutes some of the fundamental assumptions in semi-supervised learning (Kang 2003).

Existing semi-supervised classification algorithms may be classified into two categories based on their underlying assumptions. An algorithm is said to satisfy the manifold assumption if it utilizes the fact that the data lie on a low-dimensional manifold in the input space. Usually, the underlying geometry of the data is captured by representing the data as a graph, with samples as the vertices, and the pairwise similarities between the samples as edge-weights. Several graph based algorithms such as Label propagation, Markov random walks, Graph cut algorithms, Spectral graph transducer, and Low density separation are based on this assumption. The second assumption is called the cluster assumption. It states that the data samples with high similarity between them, must share the same label. This may be equivalently expressed as a condition that the decision boundary between the classes must pass through low density regions. This assumption allows the unlabeled data to regularize the decision boundary, which in turn influences the choice of the classification models. Many successful semi-supervised algorithms like TSVM and Semi-supervised SVM follow this approach (Pavan, 2010). These algorithms assume a model for the decision boundary, resulting in an inductive classifier.

Bootstrapping Classifiers from Unlabeled data

One of the first uses of unlabeled data was to bootstrap an existing supervised learner using unlabeled data iteratively. The unlabeled data is labeled using a supervised learner trained on the labeled data, and the training set is augmented by the most confident labeled samples.

This process is repeated until all the unlabeled data have been processed. This is popularly known as "Self-training", which was first proposed by Scudders (Scudder, 1965). Yarowsky (Yarowsky, 1995) applied self-learning to the "word sense" disambiguation problem. Rosenberg et al. (Rosenberg, 2005) applied self-training for object detection.

Several classifiers proposed later follow the bootstrapping architecture similar to that of self-training, but with a more robust and well-guided selection procedure for the unlabeled samples for inclusion in the training data. Semi-supervised generative models using EM (Dempster, 1977), for instance, the Semi-supervised Naive Bayes (Nigam, 2000), is a "soft" version of self-training. Many ensemble classification methods, in particular, those following the semi-supervised boosting approach (Bennet, 2002), (Mallapragada, 2009) use specific selection procedures for the unlabeled data, and use a weighted combination of classifiers instead of choosing the final classifier.

Margin based classifiers

The success of margin based methods in supervised classification motivated a significant amount of research in their extension to semi-supervised learning. The key idea of margin based semi-supervised classifiers is to model the change in the definition of margin in the presence of unlabeled data. Margin based classifiers are usually extensions of Support Vector Machines (SVM). An SVM minimizes the empirical error on the training set, along with a regularization term that attempts to select the classifier with maximum margin.

Vapnik (Yang, 1999) first formulated this problem and proposed a branch and bound algorithm.

A Mixed Integer Programming based solution is presented in (Mallapragada, 2009), which is called Semi-supervised SVM or S^3VM. Fung and Mangasarian (Pavan, 2010) proposed a successive linear approximation to the min (.) function in the loss function, and proposed VS^3VM. None of these methods are applicable to real datasets (even small size datasets) owing to their high computational complexity.

Transductive SVM (TSVM) is one of the early attempts to develop a practically usable algorithm for semi-supervised SVM. TSVM provides an approximate solution to the combinatorial optimization problem of semi-supervised SVM by first labeling the unlabeled data with an SVM

trained on the labeled data, followed by switching the individual labels of unlabeled data such that the objective function is minimized. Gradient descent was used in (Pavan, 2010) to minimize the same objective function, while defining an appropriate subgradient for the min(.) function. This approach was called ∇TSVM, and its performance is shown to be comparable to that of the other optimization schemes discussed above.

Graph Connectivity

Graph theory has been known to be powerful tool for modeling unsupervised learning (clustering) problems since its inception to relatively recent Normalized Cuts and Spectral clustering (Ng, 2002), and shown to perform well in practice (Brandes, 2003). Graph based methods represent the data as a weighted graph, where the nodes in the graph represent the data points, and the edge weights represent the similarity between the corresponding pair of data points. The success of graph based algorithms in unsupervised learning motivates its use in semi-supervised learning (SSL) problems.

The edge weight between a pair of samples is set to ∞ if they share the same label, to ensure that they remain in the same partition after partitioning the graph. Szummer and Jakkola (Szummer , 2001) and Zhu and Ghaharamani (Zhu, 2002) model the graph as a discrete Markov random field, where the normalized weight of each edge represents the probability of a label (state) jumping from one data point to the other. The solution is modeled as the probability of a label (from a labeled data point) reaching an unlabeled data point in a finite number of steps. Zhu et al., (Zhu, 2003) relax the Markov random field with a discrete state space (labels) to a Gaussian random field with continuous state space, thereby achieving an approximate solution with lower computational requirements.

Most graph based semi-supervised learning methods are non-parametric and transductive in nature, and can be shown as solutions to the discrete Green's function, defined using the discrete Graph Laplacian (Yang, 1999).

2.2.3.2. Semi-supervised clustering

Clustering aims to identify groups of data such that the points within each group are more similar to each other than the points between different groups. Clustering problem is ill-posed, and hence multiple solutions exist that can be considered equally valid and acceptable. Semi-supervised

clustering utilizes any additional information, called side information, which is available to disambiguate between the solutions. The side information is usually present in the form of instance level pairwise constraints (Wagstaff , 2000). Pair wise constraints are of two types – must-link constraints and cannot-link constraints. Given a pair of points, must link constraints require the clustering algorithm to assign the same label to the points. On the other hand, cannot-link constraints require the clustering algorithm to assign different labels to the points.

Penalizing Constraints

One of the earliest constrained clustering algorithms was developed by Wagstaff and Cardie (Wagstaff, 2000), (Wagstaff, 2001), called the COP K-means algorithm. The cluster assignment step of Kmeans algorithm was modified with an additional check for constraint violations. However, when constraints are noisy or inconsistent, it is possible that there are some points that are not assigned to any cluster. This was mitigated in an approach by Basu et. al. (Basu, 2004) which penalizes constraint violations instead of imposing them in a hard manner. A constrained clustering problem is modeled using a Hidden Markov Random Field (HMRF) which is defined over the data and the labels, with labels as the hidden states that generate the data points. The constraints are imposed on the values of the hidden states. Inference is carried out by an algorithm similar to that of K-means which penalizes the constraint violations.

Generative models are very popular in clustering. Gaussian mixture model (GMM) is one of the well-known models used for clustering (Dempster, 1977), (Figueiredo, 2002). Shental et al. (Shental, 2004) incorporated pairwise constraints into the GMMs. To achieve this, groups of points connected by must-link constraints are defined as chunklets and each chunklet is treated as a single point for clustering purposes. Zhao and Miller (Zhao, 2005) proposed an extension to GMM which penalizes constraint violations. A method to automatically estimate the number of clusters in the data using the constraint information was proposed. Lu and Leen (Lu, 2005) incorporate the constraints into the prior over all possible clustering.

In many approaches that enforce constraints in a hard manner (including those that penalize them), non-smooth solutions are obtained. A solution is called non-smooth when a data point takes a cluster label that is different from all of its surrounding neighbors. As noted in (Law, 2005), it is possible that the hypothesis that fits the constraints well may not fit the data well.

Therefore, a tradeoff between satisfying the constraints and fit to the data is required. Lange et al. (Lange, 2005) alleviate this problem by involving all the data points into a constraint through a smooth label.

Adapting the Similarity

Several semi-supervised clustering methods operate by directly modifying the entries of the pair-wise similarity matrix that are involved in constraints. All these algorithms reduce the distance between data points connected by must-link constraints and increase the distance between those connected by must-not link by a small value. Spectral Learning algorithm by Kamvar et al. (Kamvar, 2003) modifies the normalized affinity matrix by replacing the values corresponding to must-link constraints by 1 and must-not link constraints by 0. The specific normalization they use ensures that the resulting matrix is positive definite. The remaining steps of the algorithm are the same as the Spectral clustering algorithm by Ng et al. (Ng, 2002). Klien et. al. (Klien, 2002) modified the dissimilarity metric by replacing the entries participating in must-link constraints with 0 and replaced the entries participating in cannot-link constraints by maximum pairwise distance incremented by 1. This is followed by a complete link clustering on the modified similarity matrix. Kulis et al. (Kulis, 2007) propose a generalization of Spectral Learning via semi-supervised extensions to the popular normalized cut (Shi, 2000), ratio cut and ratio association (Hagen,1992). To ensure positive definiteness of the similarity matrix, they simply add an arbitrary positive quantity to the diagonal.

The specific values of increments had chosen in the above algorithms impacts the performance of the clustering algorithm. In order to apply spectral algorithms, we need the pairwise similarity matrix to be positive semi-definite. Arbitrary changes (especially decrements) to the similarity matrix may not retain its positive semi-definiteness. Some methods avoid using spectral algorithms, while some update the similarity matrix carefully to retain the essential properties. The similarity adaptation methods are adhoc in nature, and are superseded by the similarity learning approaches presented in the next section.

Learning the Similarity

The performance of a clustering algorithm depends primarily on the similarity metric defined between the samples. It is usually difficult to design a similarity metric that suits all the clustering scenarios. For this reason, attempts have been made to directly learn the similarity metric from the data using the side information. Similarity metric learning is not a new problem, and has been considered before in both unsupervised dimensionality reduction methods (LLE (Roweis , 2000), ISOMAP (Silva, 2003) and supervised methods like Fisher Linear Discriminate (Cohen, 1996), Large Margin Distance Metric Learning (Weinberger,2006) and Neighborhood Component Analysis (Goldberger, 2005). Only those methods that learn the distance metric in a semi-supervised setting, i.e., using pairwise constraints and unlabeled data are reviewed here.

Once a similarity metric is learned, standard classification algorithms may later be applied with the learned similarity metric. The distance metric learning problem can be posed in its generality as follows: learn a function $f : X \times X \rightarrow R$ such that the distance between points linked by must-link constraints is smaller than that between the points linked by must-not link constraints overall. The distance function is usually parametrized in its quadratic form, i.e $fA(xi, xj) = xTi Axj$, where A is the unknown parameter to be estimated from the constraints.

Xing et al. (kang, 2003) formulated distance metric learning as a constrained optimization problem, where A is estimated such that the sum of distances between points connected by must-link constraints is minimized, while constraining the sum of distances between points connected by must-not link to be greater than a fixed constant. Bar-Hillel et al. (Bar-Hillel, 2005) proposed Relevant Component Analysis (RCA), which estimates a global transformation of the feature space by reducing the weights of irrelevant features such that the groups of data points linked by must-link constraints (called chunklets) are closer to each other. A modified version of the constrained K-means algorithm that learns a parametrized distance function is presented in (Bilenko, 2004).

Yang et al. (Yang, 2006) learn a local distance metric by using an alternating optimization scheme that iteratively selects the local constraints, and fits the distance metric to the constraints.

They parametrize the kernel similarity matrix in terms of the eigenvalues of the top few eigenvectors of the pairwise similarity matrix computed using the RBF kernel. Hoi et al. (Hoi,

2007) present a non-parametric distance metric learning algorithm that addresses the limitations of quadratic distance functions used by almost all the other approaches. Lee et al. (Lee, Jin, 2008) proposed an efficient distance metric learning algorithm and applied it to a content based image retrieval task showing significant performance gains.

There has been a recent surge in the interest in online learning algorithms due to the large volume of datasets that need to be processed. Shalev-shwartz et al. (Shalev-Shwartz, 2004) present an online distance metric learning algorithm called POLA, that learns a quadratic distance function (parametrized by the covariance matrix) from pairwise constraints. A batch version of the algorithm is obtained by multiple epochs of the online algorithm on the training data. Davis et al. (Davis, 2007) present online and batch versions of an algorithm that searches for the parameterized covariance matrix A that satisfes the constraints maximally. Additionally, a log-determinant regularizer is added to prevent A from moving too far away from the initial similarity metric A0.

2.3. Document Preprocessing and Representation

In order to cluster or classify text documents by applying machine learning techniques, documents should first be preprocessed. In the preprocessing step, the documents should be transformed into a representation suitable for applying the learning algorithms. The most widely used method for document representation is the vector space model introduced by Salton et. al. In this model, each document is represented as a vector d. Each dimension in the vector d stands for a distinct term in the term space of the document collection (Arzucan, 2002).

A term in the document collection can stand for a distinct single-word, a stemmed word or a phrase. Phrases consist of multiple words such as “data mining” or “mobile phone” and constitute a different context than when used separately. Phrases can be extracted by using statistical or Natural Language Processing (NLP) techniques. By statistical methods phrases can be extracted by considering the frequently appearing sequences of words in the document collection (Cohen, 1996). A research on extracting phrases by using NLP techniques for text categorization is discussed by Fuernkranz et al. (Fuernkranz, 1998).

In vector space representation, defining terms as distinct single words is referred to as “bag of words” representation. Some researchers state that using phrases rather than single words to

define terms produce more accurate classification results (Cohen, 1996); whereas others argue that using single words as terms does not produce worse results (Dumais, 1998)(Sahami, M., 1998). As “bag of words” representation is the most frequently used method for defining terms and it is computationally more efficient than the phrase representation.

One challenge emerging when terms are defined as single words is that the feature space becomes very high dimensional. In addition, words which are in the same context such as biology and biologist are defined as different terms. So, in order to define words that are in the same context with the same term and consequently to reduce dimensionality the researcher have decided to define the terms as stemmed words. To stem the words, the researcher has chosen to use Porter’s Stemming Algorithm (Porter, 1980), which is the most commonly used algorithm for word stemming in English.

Preprocessing and document representation phase, which is implemented in python, consists of the following steps:

- Tokenization
- Removing stop words
- Stemming
- Term weighting
- Dimensionality reduction

These steps will be described briefly in the following sections.

Tokenization

Tokenization is the process of breaking down of documents into individual tokens.

In the tokenization process irrelevant and noisy features for the classification process such as punctuation marks and any irrelevant characters removed from documents in the collection. This is because these features are not relevant to represent the content of documents and they have no contribution in discriminating one document or category from the other. (Arzucan, 2002).

Removing Stop words

There are words, such as pronouns, prepositions and conjunctions that are used to provide structure in the language rather than content. These words, which are encountered very frequently and carry no useful information about the content and thus the category of documents, are called stopwords. Removing stopwords from the documents is very common in information retrieval. In this paper stope words are eliminated from the documents, which will lead to a drastic reduction in the dimensionality of the feature space(Arzucan, 2002).

Stemming

In order to define words that are in the same context with the same term and consequently to reduce dimensionality. Porter's stemming Algorithm which is the most commonly used algorithm for word stemming in English. For instance, we reduce the similar terms computer", "computers", and "computing" to the word stem "compute". Implementation of Porter's Stemming Algorithm in python is developed. This algorithm is embedded to the preprocessing system.

After stemming, terms that are shorter than two characters are also removed as they do not carry much information about the content of a document.

Term Weighting

We represent each document vector d as

d=(w1, w2,....., wn)

Where wi is the weight of i^{th} term of document d. There are various term weighting approaches most of which are based on the following observations (Arzucan, 2002):

- ✓ The relevance of a word to the topic of a document is proportional to the number of times it appears in the document.
- ✓ The discriminating power of a word between documents is less, if it appears in most of the documents in the document collection.

A comparative study of different term weighting approaches in automatic text retrieval is presented by Salton and Buckley (Arzucan, 2002). The term weighting approach applied in the

study and some other standard term weighting functions are discussed in the following subsections. In the study terms are defined as follows:

tfi as the raw frequency of term i in document d;

N as the total number of documents in the document corpus;

ni as the number of documents in the corpus where term i appears; and M as the number of terms in the document collection (after stopword removal and stemming is performed).

Boolean Weighting

Boolean weighting is the simplest method for term weighting. In this approach, the weight of a term is assigned to be 1 if the term appears in the document and it is assigned to be 0 if the term does not appear in the document(Arzucan, 2002).

$$w_i = \begin{cases} 1 & \text{if } tf_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Term Frequency (TF) Weighting

Term frequency weighting is also a simple method for term weighting. In this method, the weight of a term in a document is equal to the number of times the term appears in the document, i.e. to the raw frequency of the term in the document (Arzucan, 2002).

wi = tfi

Term Frequency * Inverse Document Frequency (TF*IDF) Weighting

Boolean weighting and term frequency weighting do not consider the frequency of the term throughout all the documents in the document corpus. TF*IDF weighting is the most common method used for term weighting that takes into account this property.

In this approach, the weight of term i in document d is assigned proportionally to the number of times the term appears in the document, and in inverse proportion to the number of documents in the corpus in which the term appears (Arzucan, 2002).

wi = tfi * log(N/ni) ……equation 12

TF*IDF weighting approach weights the frequency of a term in a document with a factor that discounts its importance if it appears in most of the documents, as in this case the term is assumed to have little discriminating power.

TF£IDF Weighting With Length Normalization

In this approach, to account for documents of different lengths each document vector is normalized so that it is of unit length.

$wi = tf_i * \log(N/n_i)$ ……equation 13

Salton and Buckley discuss that TF£IDF weighting with length normalization generally performs better than the other techniques (Arzucan, 2002). Therefore, we applied this weighting approach in our study.

Dimensionality Reduction

There are various methods applied for dimensionality reduction in document categorization.

Some common examples are Information Gain (IG), Mutual Information (MI), Chi-Square Statistic, Term Strength (TS), and Document Frequency (DF) Thresholding. The study discuss these techniques briefly in the following subsections.

Information Gain (IG)

Information gain measures the number of bits of information gained for category prediction when the presence or absence of a term in a document is known. When the set of possible categories is fc1; c2; …; cmg, the IG for each unique term t is calculated as follows (Joachims):

$$IG(t) = -\sum_{i=1}^{m} P(c_i)\cdot \log P(c_i) + P(t)\cdot \sum_{i=1}^{m} P(c_i|t)\cdot \log P(c_i|t) + P(\bar{t})\cdot \sum_{i=1}^{m} P(c_i|\bar{t})\cdot \log P(c_i|\bar{t})$$ ..equation 14

As seen from Equation, IG calculates the decrease in entropy when the feature is given vs. absent. $P(c_i)$ is the prior probability of category c_i. It can be estimated from the fraction of documents in the training set belonging to category ci. P(t) is the prior probability of term t. It can be estimated from the fraction of documents in the training set in which term t is present. Likewise, P(t) can be estimated from the fraction of documents in the training set in which term t

is absent. Terms whose IGs are less than some predetermined threshold are removed from the feature space.

Mutual Information (MI)

Mutual information is a technique frequently used in statistical language modeling of word associations and related applications. MI between term t and category c is defined to be (Arzucan, 2002):

$$MI(t,c) = \log \frac{P(t \wedge c)}{P(t) \times P(c)} \quad \text{.....equation 15}$$

It is estimated by using (Arzucan, 2002):

$$MI(t,c) \approx \log \frac{A \times N}{(A+R) \times (A+B)} \quad \text{.....equation 16}$$

Here, A is the number of times t and c co-occur, B is the number of times t occurs without c, R is the number of times c occurs without t, and N is the total number of documents. When t and c are independent MI (t; c) is equal to zero.

We can write equation in the following equivalent form:

$$MI(t,c) = \log P(t|c) - \log P(t) \quad \text{.....equation 17}$$

It is seen from equation for terms that have an equal conditional probability, rare terms will have a higher MI value than common terms. So, MI technique has the drawback that MI values are not comparable among terms with large frequency gaps.

Category specific MI scores for a term t can be combined into a global MI score for that term in the following two ways(Arzucan, 2002):

$$MI_{avg}(t) = \sum_{i=1}^{m} P(c_i) \times MI(t,c_i) \quad \text{.....equation 18}$$

or

$$MI_{max}(t) = \max_{i=1}^{m}\{MI(t, c_i)\} \quad (2.10)$$ ……equation 19

Terms that have lower MI values than a predetermined threshold are eliminated.

Term Strength (TS)

Term strength method, estimates term importance based on how commonly a term is likely to appear in closely related documents (Yang, 1999). The first step in this method is to use a training set of documents to find document pairs which have a similarity larger than a predetermined threshold. In the next step TS is calculated based on the estimated conditional probability that a term appears in the second document given that it appears in the first one. Suppose, x and y are any pair of distinct but related documents. Then the TS of term t is defined to be (Yang, 1999):

$$TS(t) = P(t \in y | t \in x)$$ ……equation 20

Unlike IG, MI, and X^2 statistic, TS is an unsupervised dimensionality reduction technique where document categories are not used. It is based on document clustering and assumes that documents with many shared words are related and the terms that are heavily shared among these related documents are relatively informative.

Document Frequency Thresholding (DF)

Document frequency (DF) of a term is the number of documents that term appears. In this technique, the document frequency of each unique term is computed and terms whose document frequencies are less than a predetermined threshold are eliminated. The basic assumption behind this technique is that rare terms are either non-informative for document categorization or they do not have much weight in global performance. This technique can also lead to improvement in categorization accuracy in case rare terms are noise terms. However, DF is usually not used for aggressive term elimination because there is another widely accepted assumption in information retrieval that low-DF terms are distinctive and thus relatively informative and for this reason should not be removed aggressively (Yang, 1999).

A comparative study of feature selection in text categorization is presented by Yang and Pedersen (Yang, Y). It has been reported that IG and X^2 statistic performed the best. However,

DF, the simplest and most efficient method in terms of computational complexity, performed similar to IG and X^2 statistics. It has been suggested that DF can be reliably used instead of IG and X^2 statistics when computation performances of the latter two are too expensive.

Another point to consider is that IG, MI and X^2 statistics are supervised techniques and use information about term-category associations. As our main focus is on unsupervised techniques for document organization, these methods are not suitable to be applied in our study. To reduce the dimensionality of the data, we apply DF Thresh holding. We define the document frequency threshold as 1 and hence remove the terms that appear in only one document.

Document Similarity Measure

To use a clustering or classification algorithm, a similarity measure between two documents must be defined. Cosine similarity measure is the most widely used similarity measure to calculate the similarity of two documents. This measure is defined as (Steinbach, 1999):

$$cos(\mathbf{d_1}, \mathbf{d_2}) = \frac{\mathbf{d_1} \bullet \mathbf{d_2}}{\|\mathbf{d_1}\|\|\mathbf{d_2}\|}$$ ……equation 21

That is, it is the dot product of d1 and d2 divided by the lengths of d1 and d2.

Related works

There are a lot of works done on classification of Amharic document. From these the following works are included:

Zelalem Sentayehu has worked on supervised Amharic news text classification in 2001. The overall result of his research has showed that statical technique can be used to analyze Amharic news items and classify automatically in to predefind classes. After training the classifier classified 273 out of 321 news items correctly (Zelalem, 2001).

Surafel Teklu has worked on supervised Amharic news text classification in 2003. The objective of the researcher 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 (Surafel, 2003).

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 (Surafel, 2003).

Yohannes has also worked on supervised Amharic text classification in 2007. Because of the high dimensionality of the source data, classifier algorithms that are suitable for high-dimensional data the researcher used, Decision Tree and Support Vector Machine (SVM) for the

research experiment. The researcher also used the open source Weka package for the automatic classification of the preprocessed data. Out of the many classifier algorithms available in Weka, the Logic Model Tree (LMT) and the Library of SVM (LibSVM) classifiers were used for performance testing (Yohannes, 2007).

Both LMT and LibSVM classifier showed good classification accuracy correctly classifying 79.72% and 81.15% of the test instance into the 15 news categories, respectively. However, the computational cost of the automatic classification was very high - taking several hours in high capacity computers. The classification performance measures indicate the need for additional works in developing tools and methods for mining Amharic data. (Yohannes, 2007)

Lakechew yayeh has done on unsupervised Amharic text news classification in 2011 and the researcher used k-means, bisecting k-means and average link clustering algorithms. The performances of document clustering algorithms: k-means, bisecting k-means and average link were compared for the 4, 7 and 10 clustering solutions using entropy, purity and overall similarity evaluation metrics over the different pre-defined data sets. The performances of k-means and bisecting k-means are similar in terms of the overall similarity measure in all number of clusters and they produced similar clustering solutions. However, the results of the findings indicate that the bisecting k-means produced better clustering solutions consistently according to the entropy and purity evaluation measures.

The results also shows that k-means and bisecting k-means clustering algorithms consistently produced clusters that are most similar to pre-defined classes at different data sets. Moreover, both k-means and bisecting k-means clustering algorithms produced clusters relatively with similar cluster size (number of documents), while the agglomerative hierarchical clustering algorithms generally produced clusters that are not similar to pre-defined classes and clusters with unbalanced cluster size (number of documents).

Agglomerative hierarchical clustering algorithms produced low quality results as compared to the k-means and the bisecting k-means clustering algorithms. Among the agglomerative clustering algorithms, the average link achieved the best performance as compared to single link and complete link in all evaluation measures.

In this study, the potential application of unsupervised Learning techniques for the classification of Amharic text documents was explored.

The effect of the number of clusters and the size of documents used on the performance and efficiency of clustering algorithms was tested and compared using different data sets.

Moreover, the performances of these clustering algorithms were also tested at increasing number of clusters using the same data set. The agreement between the number of predefined classes and the number of clusters discovered by the agglomerative clustering algorithm was also tested for 10 clusters over the whole document collection.

Based on the experiments done in this thesis, the following concluding remarks were made.

As the number of clusters and documents increase, the clustering solutions produced by k-means and bisecting k-means become more internally cohesive and externally isolated. However, the clustering results do not match better with the pre-defined classes and requires relatively high computational requirements.

Moreover; the purity values of single link, complete link and average link decrease.

According to the results obtained, it was difficult to determine the entropy and the overall similarity values of the three agglomerative approaches at increasing number of clusters and documents.

All the clustering algorithms: k-means, bisecting k-means, single link, complete link and average link achieved better clustering quality as the number of clusters increases with the same data set. The clustering solutions became more internally cohesive, externally isolated and match better with the pre-defined classes (Lakechew, 2011)

CHAPTER THREE

THE AMHARIC LANGUAGE AND ITS WRITING SYSTEM

3.1. The Amharic Language

The name Amharic (አማርኛ - amarəññña) comes from the district of Amhara (አማራ) in northern Ethiopia, which is thought to be the historic centre of the language. Amharic is a Semitic language and the national language of Ethiopia (ኢትዮጵያ). The majority of the 25 million or so speakers of Amharic can be found in Ethiopia, but there are also speakers in a number of other countries, particularly Eritrea (ኤርትራ), Canada, the USA and Sweden.Amharic is the working language of the Federal Government of Ethiopia and is spoken and written as a first or second language in many parts of the country (Yohannes, 2007).

Amharic, like other languages that use the Ethiopic script (Gurage, Harari, Tigre, and Tigrniya), use characters derived mainly from Geez.

The Ethiopic script was first displayed on a computer around 1986. At the time the challenge in the computer representation of the script was developing a software package that can handle character design, keyboard layout and printer set-up. The work by ESTC started an enthusiastic rush to develop Ethiopic software by different IT companies and teams of individuals which led to the problem of lack of standardization. Now a day there are more than 35 Ethiopic software products available, each with its own character set, encoding system, typeface names and keyboard layout.

The recent development of the introduction of the Ethiopic range with the Unicode standard could help in standardizing the different incompatible software products.

3.2. The Amharic writing system

The writing system of Amharic is taken from Geez (Bender, 1976; Aklilu, 1984) that in turn evolved out of Sabaean Language the descendent of South Semitic Script. It was brought to highlands of Ethiopia by immigrants from South Arabia in the first century A.D (Bender, 1976).

Geez, which remained the ecclesiastical and literary expression in Ethiopia until the 16th century, gradually gave way to Amharic that was used both in spoken and writing in the royal courts. It began to be used for literary purposes at the beginning of the 19th century as the administrative state changed its way of communication from oral to written one(Surafel, 2003).

Up to 350 A.D Geez scripts have no vowel indications. Latter, however, vocalized consonant signs had come into being by undergoing a variety of changes in the structure of the consonantal symbols. The structural changes added six additional forms to each basic consonant increasing the total number of symbols to 182(26x7). Since then, vowels became an integral part of Ethiopic writing (Surafel, 2003).

By the time Geez was replaced by Amharic, in addition to the 26 symbols that were used in the Geez language, it added symbols by deriving them from the already existing Geez alphabets.

ሸ From ሰ

ቸ From ተ

ኘ From ነ

ዠ From ዘ

ጀ From ደ

ጨ From ጠ

ኸ From ከ

This increased the total number of fundamental characters used in Amharic writing system to 34; out of which 33 are core characters and 1 is a special character (Million, 2000).

3.3. The Amharic Characters (ፊደል)

In Amharic writing system there are a total of 231 characters, 33 of the characters are the ‘core’ characters and one is ‘special’ character. Each character has seven different forms called orders that reflect the seven vowel sounds (e, u, i, a, e, i, o); one basic form and six non – basic forms representing syllable combinations consisting of a consonant and vowel. It is shown in appendix 1

There exists other character in addition to the 231 core characters that are indicated in appendix 3. The syllables with the vowel transliterated as (i) are pronounced (ə), except in final position when the vowel is not pronounced.

Characteristics of the Amharic Character

Amharic writing system is often called syllabary rather than an alphabet because the seven orders of Amharic characters indicated above represent syllable combination consisting of consonant and following vowel. The non basic forms (vocalization) are derived from the basic forms (consonants) by attaching small appendages (diacritic marks) to the right, left, top, or bottom in more or less regular modification. Some are formed by adding strokes, others by adding loops or other forms of differentiation to each core character. The writing system is difficult and vulnerable to various problems; it is difficult to automate information retrieval system for Amharic language. These writing problems have a negative effect on the performance of different machine learning approaches in text classification and text clustering. Some of the problems are discussed in the following sections.

Formation of Compound Nouns

Bender stated that compound nouns are sometimes written as two separate words (Bender, 1976). For example, ብርድ-ልብስ which means "blanket" may be written as ብርድ ልብስ or ብርድልብስ and; ክፍለከተማ as ክፍለ-ከተማ which means "sub city". This happened to be inconsistent in Amharic texts and should be considered in automatic classification (Surafel, 2003).

Character Redundancy

Out of 275 Amharic characters 231 are actually necessary to represent Amharic because the other characters are redundant, i.e., by using only one character from a group of characters with the same sound (Yohannes, 2007).

Spelling variations of a word would unnecessarily increase the number of words representing a document which could reduce the efficiency and accuracy. Amharic document processing for feature selection should therefore normalize word variants (spelling differences) caused by inconsistent usage of redundant characters.(Yohannes, 2007)

During the pre-processing stage of Amharic documents for this research, the different forms of a character that have the same sound are changed to one common form.

Consonants	Other symbols with the same sound
ሀ(hä)	ሐ፣ኀ፣ሃ፣ሓandኃ
ሰ(sä)	ሠ
አ(ä)	ዐ፣ኣ andዓ
ጸ(tsä)	ፀ

Table 1 shows a sample of redundant characters where more than one symbol is used for a given sound.

Inconsistency of Abbreviations

To write Amharic words in abbreviation people use different symbols. Forward slash ("/") and period (".") are the most common symbols used to write words in shorter form. For example the short form of the word ፍርድ ቤት can be written as "ፍ/ቤት", "ፍ.ቤት" or "ፍ-ቤት" which result in an inconsistency of abbreviating Amharic words. These different representations of the same word create high dimensional vector space and it has a negative effect on the performance of learning algorithms.(Lakechew, 2010)

Variations due to Pronunciations

The usage of foreign language words in Amharic is also found to be another source of word spelling variations. Most of the time different writers use different spellings in the writings of words adapted from foreign languages. This writing problem also has a negative effect on the performance of different machine learning approaches in text classification and text clustering

For example, the word ላብራቶሪ (laboratory) is found to have different Amharic spellings like ላቭራቶሪ፣ ላቦራቶሪ in the source data.(Yohannes, 2007)

Other Cases of Word Variations:

Usage of different affixing and suffixing style for same word causes word spelling variations. In most cases different writers use different affix and suffix spellings in the writings of words. For example difference in suffixing would result in the two writings ኢትዮጵያዊ and ኢትዮጵያአዊto refer

to human intellect while difference in prefixing would give the two writings ጥዳት and ፅዳት to mean ‘sanitary’ (Yohannes, 2007).

Punctuation

In Amharic language words are separated by two dots (: ሁለት ነጥብ), however, blank spaces are generally used. 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 from foreign languages such as (?, !, “ , ”, ‘, /, \, etc.). According to Beletu (Beletu, 1982) there are about 17 punctuation marks used in Amharic language. However, the existing Amharic software does not make use some of them. It is shown in appendix 4

Numerals

According to Bender et al., Amharic number 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. Numbering starts from one and has single characters for numbers one to ten, more than one character for multiples of ten (twenty to ninety), hundred, and thousand. There is no symbol for zero in the Amharic script. Ethiopic numbers are used mostly in writing dates and page numbers in text (Bender, 1976). Amharic number characters are indicated in appendix 2

3.4. Computerizing the Amharic Script

The ASCII code does not recognize Amharic scripts and thus cannot assign numeric codes to the scripts. For ease of preprocessing and compatibility reasons, the Amharic text was transliterated into an ASCII representation using SERA.

A SERA is a scheme for transliterating Amharic characters. The fundamentals of SERA are discussed in Daniel (1996). SERA is a convention for transliteration of Amharic characters (Fidel) script into Latin script that insures the integrity of the format and content of the original document, and that can be fully transportable across all computer mediums.

CHAPTER FOUR

4. METHODOLOGY

Introduction

The machine learning approach to text categorization is to automatically build the classifiers by learning the concept descriptions of the categories. One type of machine learning, applied to text categorization, is "supervised learning". This requires a set of pre-labeled (pre-categorized) training documents for generating classifiers. In contrast, "unsupervised learning" refers to the task of automatically identifying a set of categories from a set of unlabeled documents and grouping these unlabeled documents under these identified categories (Merkl , 1998). This task is typically called document clustering. Semi-supervised classification algorithms train a classifier given both labeled and unlabeled data. The goal is to label only the unlabeled data available during training

In order to cluster or classify text documents by applying machine learning techniques, documents should first be preprocessed. In the preprocessing step, the documents should be transformed into a representation suitable for applying the learning algorithms. The most widely used method for document representation is the vector space model introduced by Salton et. al.(Salton, 1975).

In this model, each document is represented as a vector d. Each dimension in the vector d stands for a distinct term in the term space of the document collection.

A term in the document collection can stand for a distinct single-word, a stemmed word or a phrase. Phrases consist of multiple words such as "data mining" or "mobile phone" and constitute a different context than when used separately. Phrases can be extracted by using statistical or Natural Language Processing (NLP) techniques. By statistical methods phrases can be extracted by considering the frequently appearing sequences of words in the document collection (Cohen, 1996). A research on extracting phrases by using NLP techniques for text categorization is discussed by Fuernkranz et al. (Fuernkranz, 1998).

In vector space representation, defining terms as distinct single words is referred to as "bag of words" representation. Some researchers state that using phrases rather than single words to define terms produce more accurate classification results; whereas others argue that using single words as terms does not produce worse results (Sahami, 1998). As "bag of words" representation

is the most frequently used method for defining terms and it is computationally more efficient than the phrase representation.

The next phase of the approach used is clustering the documents based on their similarity. Next to this phase classify the clusters in to their predefined categories. In this stage of the KDT process, experimentations were conducted using the most commonly used semi-supervised machine learning algorithms and finally the outputs produced were evaluated using different evaluation metrics.

4.1. Architecture of Amharic Text News classification

Amharic news classification using semi-supervised has its own architecture. The architecture is composed of five components. These are document collection, document preprocessing and representation, clustering, classification and evaluating the results. The architecture of Amharic document classification system is described in Figure 1.

Some of the documents are collected from ENA manually and then preprocessing of documents was held.

In the document preprocessing stage transliteration, tokenization, normalization, stop words and numbers removal, stemming and dimension reduction were done.

Once all these document preprocessing and representation activities were done, the datasets were prepared in an appropriate format and given to the learning algorithms. The learning algorithms process this dataset and group them into the appropriate clusters and classify to its category and finally the performances of those classification algorithms were evaluated using different classification evaluation metrics. The details of each phase are discussed in the following sections.

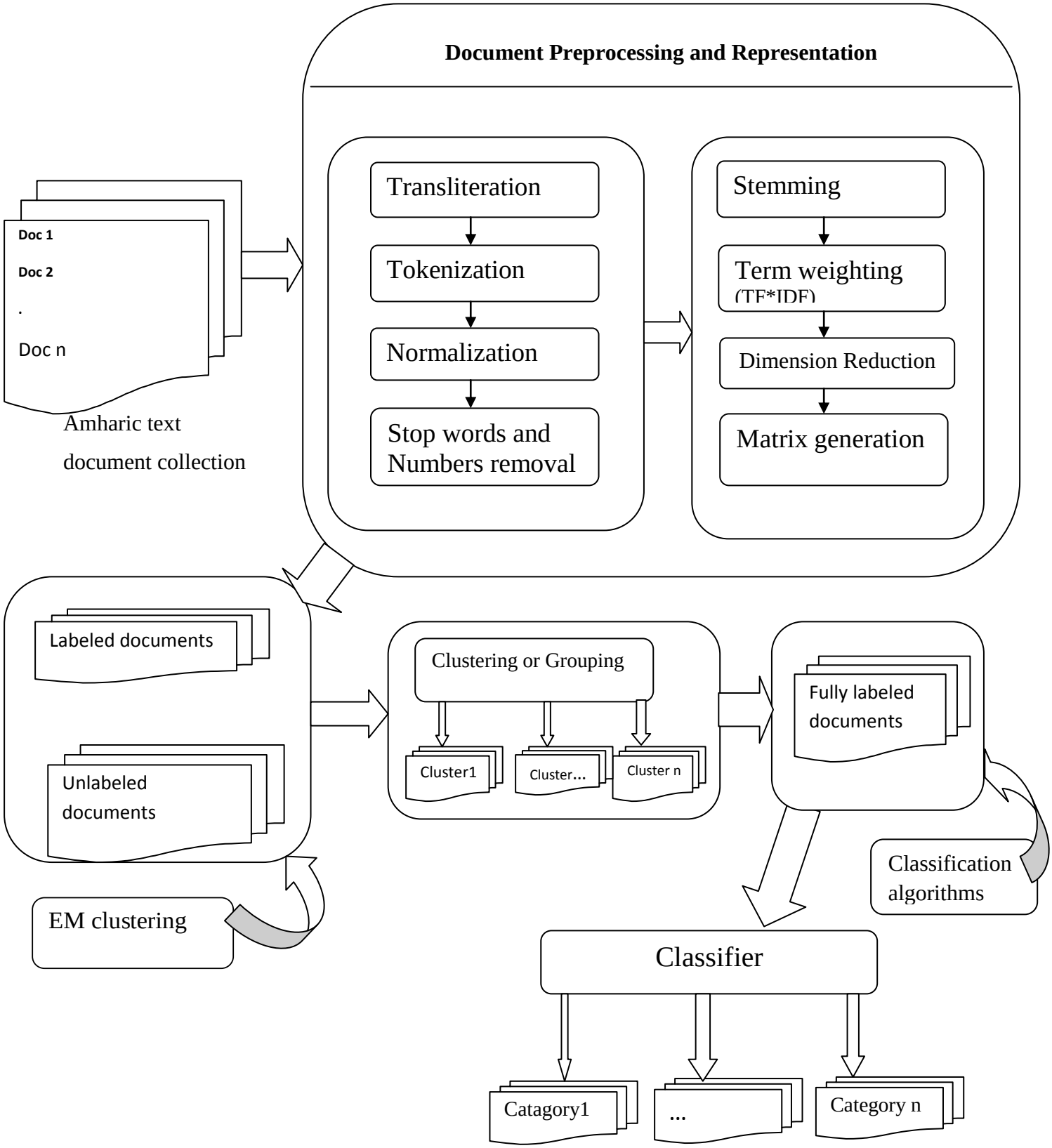


Figure 1Semi-Supervised Amharic text classification Architecture

4.2. Document Collection

The document data set that was used for the experiments were Amharic text News which was collected from ENA and used by other previous researchers. Even though, classification of news items is done manually, ENA uses software called ENASoft to make the management of news items easy. Once the classification task is done manually, ENASoft is used to dispatch news items into different Media such as Ethiopian Radio and Television, Addis Zemen, Sheger FM and others. The total number of categories collected and considered in this study are 10; with a total 3,154 Amharic news items or documents.

4.3. Document Preprocessing

Document preprocessing is important to improve the accuracy, efficiency, and scalability of the classification process. In order to get better experiment results, language dependent document 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 includes a removal of non-informative words or characters from the text. It is the first step in the preparation of documents to present them in a format suitable for classification.

The process of tokenization, normalization and stemming is language-dependent and in this thesis the different characteristics or features of the Amharic language were considered in the development of the algorithms. The document preprocessing task was implemented using python programming language (Python 3.1). The document preprocessing activities done in this thesis are presented in the following subsections.

4.4. Amharic Document Transliteration

For ease of use and compatibility purposes, the Amharic documents originally written using the Amharic script fidel were transliterated to an ASCII representation using a file conversion utility called g2 command. g2 command was made available to us through Daniel Yacob of the

Ge'ez Frontier Foundation (Daniel). Both document preprocessing activities and the experiments were done using the transliterated form in order to simplify spelling normalization of Amharic characters and to make it compatible with the classification tool used for the experiments.

Tokenization

Tokenization is the process of breaking down of documents into individual tokens.

In the tokenization process irrelevant and noisy features for the classification process such as punctuation marks and any non Amharic characters were removed from documents in the collection using python. This is because these features are not relevant to represent the content of documents and they have no contribution in discriminating one document or category from the other.

```
Read document file
Read punctuations list
Read unnecessary characters list
For each token in file
        If token ends with punctuation then
                Remove punctuation from file
        End if
        If token is in characters list
                Remove token from file
        End if
End for
```

Figure 2document tokenization algorithm

Normalization

In the Amharic writing system characters with the same sound have different symbols. These different symbols must be considered as similar because they do not have effect on meaning. As a result, in this study, all different symbols of the same sound were converted to one common form. In order to exploit this equivalence Lakechew's (Lakechew, 2011) algorithm was used. Thus, for example, if the character was one of ሐ፣ሓ፣ኀ፣ኻ ኃ or ኧ (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.

Read document file
Read characters list
For each token in file
    If token is ሐ or ሓ or ኀ or ኻ ኃ or ኧ *then*
        token is ሀ
    elseif ሠ
        token is ሰ
    elseif ዐ
        token is አ
    End if
End for

Figure 3 Normalization algorithm

Stemming

Natural language texts are characterized by variations in word forms. The most common ways of creating word variant are suffixing and prefixing. In general, word variants may be caused by factors including grammar requirements, national or local usage, transliteration, abbreviation, and spelling errors. Stemming might be used to normalize word variants by removing affixes through identification of word-stems from full words.

```
Read document file
Read exception list
Read prefix list
Read suffix list
Assign the first 1, 2, 3,… character(s) of the token to prefix
Assign the last 1, 2, 3,… character(s) of the token to suffix
For each token in file
        If token is not in exception list and prefix is in prefix list
                Remove prefix from token
        End if
        If token is not in exception list and suffix is in suffix list
                Remove suffix from token
        End if
End for
```

Figure 4 Stemming algorithm

In this study, tools for removal of common prefixes and suffixes, correction variations due to transliteration, correcting common spelling variation, and normalizing different forms of words are adapted from Nega Alemayehu(Nega, 2002).

Stop Word Removal

In case of irrelevant attributes in the dataset, attribute subset selection can be used to find a reduced set of attributes while keeping the original data class distribution as much as possible (Yohannes, 2007).

After a document is processed and its features identified, different techniques are used to select the features that adequately represent the document for the purpose of text classification.

Removal of stop words is one method of feature selection. Stop words are sometimes defined as function words. Function words have important role in grammar but carry little meaning, and, therefore, do not contribute much to categorization (Yu, 2005).

The stop words are of two kinds: those which are common to Amharic language text and those Amharic news items. 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 are usually found at the end of a sentence. Hence, news specific common words of this type were used as a stopword list. Both types of common words were used by Lakechew (Lakechew, 2011) 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 stop words from each document during the indexing process.

```
Read document file
Read stop word list
For each token in file
        If token is in stop word list then
                Remove token from file
        End if
        If token is number then
                Remove token from file
        End if
End for
```

Figure 5 Stop word removals

Compound words and abbreviations expansion

Compound words and abbreviations have different ways writing styles leading to inconsistency in writing. This different and inconsistent representation of compound words and abbreviations was solved by expanding all the short forms into their expanded form.

Concatenation of Compound Words

There are different representations of compound words in Amharic writing which result in an increase in the dimension of the vector space. Hence, to solve this problem, algorithm 8 was used to convert the expanded form into a single common standard form after creating a list that contained such type of words.

```
Read document file
Read compound words list
For each token in file
        If token is in list then
                Concatenate token with the next token
        End if
End for
```

Figure 6 Concatenation of compound words

Term Weighting

For the purpose of classification and clustering, a document can be considered as a collection of key words. These key words are often called features or attributes of the document. All terms or words with in a document are not relevant equally to represent the contents of the document. Term weighing is used to weight representative terms that describe and summarize document content based on the importance of terms with in a document. Hence, in order to define the importance of a word within Amharic text documents, a vector representation was used, where for each word a numerical importance value is stored using the TF*IDF term weighting approach.

Dimension Reduction

Document representation using bag of words create a problem in that the feature space becomes very high dimensional which imposes a big challenge on the performance of clustering algorithms. The computational complexity of any operations with such feature vectors will be proportional to the size of the feature vector (Yang, 1997).

In addition, it has been shown that some specific words in specific languages only add noise to the data and removing them from the feature vector actually improves classification performance (Yang, 1997).

Feature selection not only reduces the high dimensionality of the feature space, but also provides better data understanding, which improves the classification and clustering result (Sebastiani, 2002). Hence it is important to reduce the size of the feature vector by selecting only relevant terms that leads to better clustering performance.

There are various methods applied for dimensionality reduction in document categorization.

Some common examples are Information Gain (IG), Mutual Information (MI), Chi-Square Statistic, Term Strength (TS), and Document Frequency (DF) thresh holding.

4.5. Document classification and Evaluation

A number of different techniques are used to reliably estimate the accuracy of classifiers. The techniques include Naivesbayes, Hyperpipes and RBF network for classification and EM for clustering purpose. In the experimental part of this study cross-validation technique is used.

Cross-validation

The cross-validation method can be generalized into two as k-fold cross-validation and stratified cross-validation.

In k-fold cross-validation the initial data are randomly partitioned into mutually exclusive subsets (folds), D_1, D_2, …, D_k each of approximately equal size. In iteration i, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the model, which will continue for all K iterations. Classification accuracy is estimated by dividing the overall number of correct classification from the iterations by the total number of instances (documents) in the initial data (Arzucan, 2002).

In stratified cross-validation, the folds are stratified so that the class distribution of the documents in each fold is approximately the same as that in the initial data.

Generally in practice 10-fold cross-validation is employed for estimating accuracy due to its relatively low bias and variance. The 10-fold cross-validation is used for all experiments in this research.

Evaluation of a classifier can be conducted by measuring its efficiency and its effectiveness. Efficiency is typically measured by using the elapsed processor time and it refers to the ability of a classifier to run fast. Efficiency of a classifier can usually be measured on two dimensions: learning efficiency (i.e., the time a machine learning algorithm takes to generate a classifier from a set of training examples) and categorization efficiency (i.e., the time the classifier takes to assign appropriate categories to a new document). Because of the unstable nature of parameters on which the evaluation depends, efficiency is rarely used as the singular performance measure in text categorization. However, efficiency is important for the practical application of the system.

A much more common evaluation method for text categorization systems is effectiveness: this refers to the ability to take the right decisions on the categorization of new incoming documents. There are several commonly used performance measures of effectiveness. However, there is no agreement on one single measure for use in all applications. Indeed, the type of measure that is preferable depends on the characteristics of the test data set and on the user's interests. The absence of one optimal measure of effectiveness makes it very difficult to compare the relative effectiveness of classifiers (Arzucan, 2002).

In the next section, the study will discuss various performance measures of effectiveness that have been widely used for the evaluation of text categorization systems.

4.6. Performance Measures of Effectiveness

While a number of different conventional performance measures are available for the effectiveness evaluation for text categorization, the definition of almost all measures is based on the same 2×2 contingency table model that is constructed as shown in following Table 2

In this table, 'YES' and 'NO' represent a binary decision given to each document dj under category ci. Each entry in the table indicates the number of documents of the specified type:

- TP_i: the numbers of true positive documents that the system predicted was YES, and were in fact in the category c_i.
- FP_i: the number of false positive documents that the system predicted were YES, but actually were not in the category c_i.
- FN_i: the numbers of false negative documents that the system predicted were NO, but were in fact in the category c_i.
- TN_i: the numbers of true negative documents that the system predicted were NO, and actually were not in the category ci.

Here, note that the larger TP_i and TN_i values are (or the smaller FP_i and FN_i values are), the more effective c_i is.

category c_i		label by human expert	
		YES is correct	**NO** is correct
label by the system	predicted **YES**	TP_i	FP_i
	predicted **NO**	FN_i	TN_i

Table 2 effectiveness evaluation for text categorization

Given such a two-way contingency table, most conventional performance measures compute a single value from the four values in the table. The standard performance measures for classic information retrieval research are recall and precision that has been also frequently adopted for the evaluation for the text categorization.

These measures are computed as follows.

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad \text{if } TP_i + FN_i > 0$$

……equation 22

$$Precision = \frac{TP_i}{TP_i + FP_i} \quad \text{if } TP_i + FP_i > 0$$

……equation 23

Recall measures the proportion of documents that are predicted to be YES and correct, against all documents that are actually correct. While, the precision is the proportion of documents which are both predicted to be YES and are actually correct, against all documents that are predicted YES. In general, the higher precision is, the lower recall becomes, and vice versa. For example, we can achieve very high precision by rarely predicting 'YES' (i.e., by setting a very high threshold value) or very high recall by rarely predicting 'NO' (i.e., by setting very low threshold value). For this reason, they are seldom used alone as a sole measure of effectiveness. Instead, it is common in the literature to show two associated values of recall and precision at each level.

Other performance measures that are purely based on the contingency table are accuracy and error. They are defined as follows:

$$Accuracy = \frac{TP_i +}{|D|} \quad \text{where } |D| = TP_i + FP_i + FN_i + TN_i > 0$$

……equation 24

Accuracy and error are also used for performance measures in text categorization. The accuracy and error are defined as the proportion of documents that are correctly predicted and the proportion of documents that are wrongly predicted, respectively. Both measures, in common, have |D| which is the total number documents in their denominator.

CHAPTER FIVE

EXPERIMENT AND PERFORMANCE EVALUATION

Introduction

This chapter discusses the results obtained from the experiment. The experiments are performed based on the concepts discussed in the previous chapters.

The experiments were done using three document classification and one clustering algorithms: Naivesbay's, Hyperpipes and RBF Network classification algorithm and EM clustering algorithms. The results obtained from these classifications and clustering algorithms are discussed and a comparison of these classification algorithms was done to select the best classification solution among the algorithms.

5.1. Experimentations setup for supervised

For supervised experiment the researcher used the same data with semi-supervised, shown in Table 3, a total of 10 classes and 3154 documents were used in the experimentation process. The all documents are labeled to its pre-defined classes with the corresponding provided by ENA.

To test the performances of Naives bay's, Hyperpipes and RBF Network classification algorithms at increasing number of classes and documents, the different pre-defined number of classes and the corresponding pre-classified documents were used to conduct the experiments. The 10 categories were divided into three and the experiments were done on 4, 7 and 10 number of classes using 1250, 2200 and 3154 documents respectively as shown in Table 3. The first experiment was don on four classes: 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected. The second experiment was performed on seven classes: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', and maheberawiguday. The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Experiments	No.	List of Classes used	Number of documents	Algorithms used
On four classes	1	1. Economy 2. politica 3. sport 4. tena	292 301 335 322 Total 1250	1. Naives baye's 2. Hyperpipes 3. RBF Network
On seven classes	2	1. Economy 2. politica 3. sport 4. tena 5. bahelnaturism 6. science 7. maheberawiguday	292 301 335 322 335 301 314 Total 2200	1.Naives baye's 2.Hyperpipes 3.RBF Network
On ten classes	3	1. Economy 2. politica 3. sport 4. tena 5. bahelnaturism 6. science 7. maheberawiguday 8. Tmhert 9. Heg 10. Adega	292 301 335 322 335 301 314 297 319 338 Total 3154	1.Naives baye's 2.Hyperpipes 3.RBF Network

Table 3 Experimentations setup

5.1.1. Naïve Bays Test

As explained in chapter two Naïve Bays is one of the simple algorithms of machine learning. A naive Bayes classifier could be defined as an independent feature model deals with a simple probabilistic classifier based on applying Bays' theorem with strong independence assumptions. The test results for the naïve Bays classifier is discussed in the following sections.

Experiment on four classes

Four classes 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected; where 1250 news items were used. 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. The following confusion matrix details are for the four classes:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         956               76.48 %
Incorrectly Classified Instances       294               23.52 %

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 209  38  19  26 |   a = economy
  91 194  29  22 |   b = politica
   6   2 264  12 |   c = sport
  14   5  30 289 |   d = tena
```

Figure 7 confusion matrix for four classes using Naivebays

The first row indicates that 209 documents are classified correctly as the category 'economy';83 documents from this category are misclassified as other category.38 as 'politica'; 19 as 'sport' and 26 as 'tena'. The second row indicates 91 documents from the category 'politica' are classified incorrectly to the category 'economy'; 194 documents are classified correctly; 29 documents classified incorrectly to the category 'sport' and 22 documents are classified incorrectly to the category 'tena'. The third row indicates 6 documents from the category 'sport' are classified incorrectly to the category 'economy'; 2 documents from the category 'sport' are classified incorrectly to the category 'politica'; 264 documents are classified correctly; and 12 documents classified incorrectly to the category 'tena'. In the same manner, for the fourth row, category 'tena', 14 documents classified incorrectly as a category 'economy', 5 documents are

classified incorrectly to the category ‘politica’ 30 document is classified incorrectly to the category ‘sport’ and 289 documents classified correctly in the category.

As we can see from the above experiment result the algorithm classified 76.48% of the documents correctly and 23.52 % of the document incorrectly. That is correctly classified news items are 956 out of 1251. The highest confusion (91) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on Seven classes

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1330	60.4545 %
Incorrectly Classified Instances	870	39.5455 %

=== Confusion Matrix ===

```
   a   b   c   d   e   f   g   <-- classified as
 214  33  19  33  28  41   6 |   a = science
  39  90  52  49  29  44   6 |   b = economy
  24  38 195  17  27  21  22 |   c = politica
   2   2   0 187  17  16   0 |   d = tena
  16  17   9  34 210  28   7 |   e = bahelnaturism
  28  24  11  47  36 182   6 |   f = maheberawiguday
   4   4   5   3   9  17 252 |   g = sport
```

Figure 8 confusion matrix for four classes using Naivebays

As we can see from the above experiment result the algorithm classified 60.4545 % of the documents correctly and 39.5455 % of the document incorrectly. The highest confusion (49) happened between tena and economy followed by tena and maheberawiguday (47). This shows that these classes have a lot in common.

Experiment on ten classes

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        2223               69.7084 %
Incorrectly Classified Instances       966               30.2916 %
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 284   3   1   2   1   0   3   5   2   4 |   a = politica
  11 168  32  36  26  23  34  17  16  11 |   b = economy
   4  17 235   6  23  19  17  27  32  10 |   c = heg
   3  23  14 168  10  18  18  13  16   5 |   d = science
   0   4  10   1 264  11   7   3  10   0 |   e = tena
   1  12  30  11  31 138  13  21  18   5 |   f = maheberawiguday
   2   4  11   4  13   3 283   3   2   0 |   g = tmhert
   6   7  26   3   8  14  11 266   9   8 |   h = bahelnaturism
   2   4  24  11  18  17   6  14 198   4 |   i = adega
   2   2   8   2   4   5   1  12   6 219 |   j = sport
```

Figure 9 confusion matrix for ten classes using Naivebays

From the above experiment result we can see that the algorithm classified 69.7084 % of the documents correctly and 30.2916 % of the document incorrectly. From the confusion matrix the highest confusion (36) happened between science and economy followed by tmhert and economy (34). This shows that these classes are more related.

5.1.2. Hyperpipes

HyperPipes is a very simple algorithm that constructs a "hyperpipe" for every class in the data set; each hyperpipe contains each attribute-value found in the examples from the class it was built to cover. An example is classified by finding which hyperpipes covers it the best. Extremely simple algorithm, but has the advantage of being extremely fast, and works quite well when you have lots of attributes.

Experiment on four classes

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances         925               74
%
Incorrectly Classified Instances       325               26
%
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 202  56  11  23 |   a = economy
  97 204  12  23 |   b = politica
  16  10 249   9 |   c = sport
  25  18  25 270 |   d = tena
```

Figure 10 confusion matrix for four classes using hyperpipe

As we can see from the above experiment result the algorithm classified 74% of the documents correctly and 26% of the document incorrectly. From the confusion matrix the highest confusion (97) happened between politica and economy. This shows that these classes are more related.

Experiment on seven classes

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1171               53.2273 %
Incorrectly Classified Instances      1029               46.7727 %

=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 251  28  33  10  13  36   3 |   a = science
  85  86  55  19  22  38   4 |   b = economy
  52  74 154   5  19  18  22 |   c = politica
  27  22   3 126  25  21   0 |   d = tena
  41  29  13  19 184  31   4 |   e = bahelnaturism
  75    43    20    24    37  133      2  |      f  =
maheberawiguday
   9   9  10   1  11  17 237 |   g = sport
```

Figure 11 confusion matrix for seven classes using hyperpipe

As it is shown in the above the algorithm classified 53.2273% of the documents correctly and 46.7727 % of the document incorrectly. As we can see from confusion matrix the highest confusion (85) happened between politica and economy followed by maheberawiguday and economy (75). This indicates that these classes are more related each other.

Experiment on ten classes

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1484	46.535 %
Incorrectly Classified Instances	1705	53.465 %

=== Confusion Matrix ===

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 128  67  31  10   5   9  11  19   7  18 |   a = politica
  65 158  38  26  22  10  35  14   5   1 |   b = economy
  15  46 193  11  26  19  23  34  20   3 |   c = heg
   9  53  27 117  11  10  26  18  15   2 |   d = science
   3  17  28   7 203  17  21   2  12   0 |   e = tena
   7  25  56  17  34  81  22  25  11   2 |   f = maheberawiguday
   6  50  38   9  34  19 137  19  12   1 |   g = tmhert
  18  17  59  22  16  22  22 162  15   5 |   h = bahelnaturism
  12  16  60  14  27  18  13  16 122   0 |   i = adega
  16   5  19   0   4   7   5  14   8 183 |   j = sport
```

Figure 12 confusion matrix for ten classes using hyperpipe

As we can see from the above experiment result the algorithm classified 46.535% of the documents correctly and 53.465 % of the document incorrectly. From the confusion matrix the highest confusion (65) happened between politica and economy followed by heg and adega (60). This shows that these classes have a lot in common.

5.1.3. RBF network

Radial basis function (RBF) networks have a static Gaussian function as the nonlinearity for the hidden layer processing elements. The Gaussian function responds only to a small region of the input space where the Gaussian is centered. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions. The simulation starts with the training of an unsupervised layer. Its function is to derive the Gaussian centers and the widths from the input data. These centers are encoded within the weights of the unsupervised layer using competitive learning. During the unsupervised learning, the widths of the Gaussians are

computed based on the centers of their neighbors. The output of this layer is derived from the input data weighted by a Gaussian mixture (Stig-Erland, 2007).

The advantage of the radial basis function network is that it finds the input to output map using local approximators. Usually the supervised segment is simply a linear combination of the approximators. Since linear combiners have few weights, these networks train extremely fast and require fewer training samples.

Experiment on four classes

Time taken to build model: 1.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	902	72.16 %
Incorrectly Classified Instances	348	27.84 %

=== Confusion Matrix ===

```
   a   b   c   d   <-- classified as
 206  39  20  27 |   a = economy
  93 170  46  27 |   b = politica
   7   3 247  27 |   c = sport
  23   6  30 279 |   d = tena
```

Figure 13 confusion matrix for four classes using RBF Network

As it is depicted in the above the algorithm classified 72.16% of the documents correctly and 27.84 % of the document incorrectly. As we can see from the confusion matrix the highest confusion (93) happened between politica and economy. This shows that these classes are more related.

Experiment on seven classes

Time taken to build model: 1.77 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1243               56.5    %
Incorrectly Classified Instances       957               43.5    %
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 203  34  19  47  10  58   3 |   a = science
  35  88  48  64  12  59   3 |   b = economy
  20  42 189  26  16  36  15 |   c = politica
   3   2   0 183   7  29   0 |   d = tena
  16  13  11  47 140  93   1 |   e = bahelnaturism
  23  21  10  48  18 213   1 |   f = maheberawiguday
   3   6  11  15   4  28 227 |   g = sport
```

Figure 14 confusion matrix for seven classes using RBF Network

As we can see from the above experiment result the algorithm classified 56.5% of the documents correctly and 43.5 % of the document incorrectly. From the confusion matrix the highest confusion (64) happened between politica and economy followed by maheberawiguday and economy (59). This shows that these classes have a lot in common.

Experiment on ten classes

Time taken to build model: 48.89 seconds

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1890               59.2662 %
Incorrectly Classified Instances      1299               40.7338 %
=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 288   2   4   2   1   1   0   3   2   2 |   a = politica
  11 133  54  49  26  72  10   8  11   0 |   b = economy
   4  15 224  11  19  73   8   9  26   1 |   c = heg
   4  30  28 155  13  34   6   6  11   1 |   d = science
   0  18  17   4 243  16   1   3   8   0 |   e = tena
   1  29  47  11  29 142   6   3  11   1 |   f = maheberawiguday
   3  18  61  14  27  10 187   5   0   0 |   g = tmhert
   4  15  31   7  14  54  40 181   9   3 |   h = bahelnaturism
   3   5  49  11  20  31   4   9 163   3 |   i = adega
   3  14  21   4   6  24   0   3  12 174 |   j = sport
```

Figure 15 confusion matrix for ten classes using RBF Network

As we can see from the above experiment result the algorithm classified 59.2662 % of the documents correctly and 40.7338 % of the document incorrectly. The confusion matrix shows the highest confusion (54) happened between politica and economy followed by heg and adega (54). This shows that these classes have a lot in common.

Number of classes	Naivebays accuracy(%)	Hyperpipes accuracy(%)	RBF Network accuracy (%)
Four	76.48	74	72.16
Seven	60.4545	53.2273	56.5
Ten	69.7084	46.535	59.2662

Table 4 Comparision of algorithms at different class level

As shown in Table and figure above, Naivebays achieved the highest performance in terms of accuracy.

5.2. Experimentations setup for semi-supervised learning

As shown in Table 3, a total of 10 classes and 3154 documents were used in the experimentation process. The list of pre-defined classes with the corresponding documents is already provided by ENA. These pre-classified documents were used as a centroid before clustering.

To test the performances of Naives bay's, Hyperpipes and RBF Network classification algorithms at increasing number of classes and documents, the different pre-defined number of classes and the corresponding pre-classified documents were used to conduct the experiments. The predefined classes were arranged in the manner that each classes are not related. That means classes that have great similarity have smaller probability to be selected at the same time. The 10 categories were divided into three and the experiments were done on 4, 7 and 10 number of classes using 1250, 2200 and 3154 documents respectively as shown in Table 3.

5.2.1. Naïve Bayes Test

Experiment on four categories

Four classes 'economy', 'politica', 'sport' and 'tena' that contain relatively equal number of news items were selected; where 1250 news items were used. 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. The following confusion matrix details are for the four classes:

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1043	83.44 %
Incorrectly Classified Instances	207	16.56 %
Total Number of Instances	1250	

=== Confusion Matrix ===

```
  a   b   c   d   <-- classified as
237  39   4  12 |   a = economy
 81 187  13  20 |   b = politica
  7   5 307  16 |   c = sport
  6   4   0 312 |   d = tena
```

Figure 16 confusion matrix for four classes using Naivesbays

The first row indicates that 237 documents are classified correctly as the category ‘economy’;55 documents from this category are misclassified as other category.39 as ‘politica’; 4 as ‘sport’ and 12 as ‘tena’. The second row indicates 81 documents from the category ‘politica’ are classified incorrectly to the category ‘economy’; 187 documents are classified correctly; 13 documents classified incorrectly to the category ‘sport’ and 20 documents are classified incorrectly to the category ‘tena’. The third row indicates 7 documents from the category ‘sport’ are classified incorrectly to the category ‘economy’; 5 documents from the category ‘sport’ are classified incorrectly to the category ‘politica’; 307 documents are classified correctly; and 16 documents classified incorrectly to the category ‘tena’. In the same manner, for the fourth row, category ‘tena’, 6 documents classified incorrectly as a category ‘economy’, 4 documents are classified incorrectly to the category ‘politica’ 0 document is classified incorrectly to the category ‘sport’ or there is no economy document that are predicted as politica and 312 documents classified correctly in the category. So, correctly classified news items are 1,052 out of 1251and the average accuracy is 83.44 percent. The highest confusion (81) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on Seven Categories

The second experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, and maheberawiguday.

=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        1576               71.6364 %
Incorrectly Classified Instances       624               28.3636 %
Total Number of Instances             2200
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 203  27   3  13  12  14  20 |   a = economy
  58 167  11  10  20  17  18 |   b = politica
   1   3 296   6  15   2  12 |   c = sport
   4   4   0 280   2   9  23 |   d = tena
   9   2   6  21 261   6  30 |   e = bahelnaturism
  19   6   2  23  15 205  31 |   f = science
  16      9      3    69    31    22  164  |      g  =
maheberawiguday
```

Figure 17 Confusion matrix for seven classes using Naivesbays

As we can see from the above experiment result the algorithm classified 71.6364% of the documents correctly and 28.3636 % of the document incorrectly. From the confusion matrix we can see that the highest confusion (69) is happened between tena and maheberawiguday followed by ecoinomy and politica(58). This shows that tena and maheberawiguday have a lot in common.

Experiment on ten Categories

The third experiment was performed on ten categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, ‘maheberawiguday’, ‘tmhert’,’heg’ and ‘adega’.

Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===

```
Correctly Classified Instances        2032               64.4261 %
Incorrectly Classified Instances      1122               35.5739 %
Total Number of Instances             3154
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 197  26   1   9  13   8  12  10   6  10 |   a = economy
  53 153   8   5  13  14  13   9  26   7 |   b = politica
   1   3 288   5  10   2  11   6   5   4 |   c = sport
   4   1   0 263   1   9  12  18   3  11 |   d = tena
   5   0   5  17 241   4  23  16  13  11 |   e = bahelnaturism
  12   5   0  11  11 179  18  47   6  12 |   f = science
  16   7   3  49  18  18 124  38  17  24 |   g = maheberawiguday
   2   3   0  31   6  17   9 222   3   4 |   h = tmhert
  10  15   2  17  29   7  34  29 146  30 |   i = heg
   3  14   1  23  20  13  20  14  11 219 |   j = adega
```

Figure 18 confusion matrix for ten classes using Naivesbays

As we can see from the above experiment result the algorithm classified 2032 (64.4261%) of the document correctly and 1122(35.5739%) of the documents incorrectly. As it is shown in the confusion matrix the highest confusion (53) is happened between ecoinomy and politica followed by tena and maheberawiguday). This shows that ecoinomy and politica have a lot in common.

Number of classes	Accuracy performance achieved
4	83.44 %
7	71.6364 %
10	64.4261 %

Table 5 accuracy performances achieved at different levels of class using Naivebays algorithm

From the above table, the highest accuracy is 83.44% and the lowest is 64.4261 % when the number of class is four and ten respectively. The seven class category yields an accuracy of 71.6364 %. From this we can deduce that when the number of class increase the accuracy goes in a reverse way. This is due to the increase of similarity between classes.

5.2.2. Hyper Pipes test

Experiment on four categories

The first experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, and ‘tena’.

HyperPipes classifier

Time taken to build model: 0.01 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1035	82.8 %
Incorrectly Classified Instances	215	17.2 %
Total Number of Instances	1250	

=== Confusion Matrix ===

```
   a   b   c   d   <-- classified as
 242  38   1  11 |   a = economy
  95 176  17  13 |   b = politica
   6   4 320   5 |   c = sport
  16   9   0 297 |   d = tena
```

Figure 19 confusion matrix for four classes using Hyperpipes

As we can see from the confusion matrix details 1035 news items out of 1250 are correctly classified and the average percent accuracy is 82.8%. The highest confusion (95) happened between politica and economy. This shows that these classes have a lot in common.

Experiment on seven categories

The second experiment was performed on seven categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', and 'maheberawiguday'.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1529	69.5 %
Incorrectly Classified Instances	671	30.5 %
Total Number of Instances	2200	

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 211  34   4   7   8  16  12 |   a = economy
  70 166  15  12  17   7  14 |   b = politica
   4   3 304   2  11   2   9 |   c = sport
   9   5   0 261   7  10  30 |   d = tena
   9   9  10  23 259   9  16 |   e = bahelnaturism
  26  11   2  27  18 188  29 |   f = science
  25  13   7  62  37  30 140 |   g = maheberawiguday
```

Figure 20 confusion matrix for seven classes using HyperPipes

As shown from the above confusion matrix details 1529 news items out of 2,200 are correctly classified and the average percent accuracy is 69.5 percent. As we can see from the confusion matrix the highest confusion (70) happened between politica and economy followed by tena and maheberawiguday (62). This shows that these classes have a lot in common.

Experiment on ten categories

The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

HyperPipes classifier

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1944	61.636 %
Incorrectly Classified Instances	1210	38.364 %
Total Number of Instances	3154	

```
=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 196  32   3   6  11  13  12   7   6   6 |   a = economy
  69 151   9   6  16   9  10   8  18   5 |   b = politica
   2   3 298   2  12   4   8   4   1   1 |   c = sport
   9   4   0 238   3   8  23  21   2  14 |   d = tena
  10   5  10  19 248   6  19   8   6   4 |   e = bahelnaturism
  21   6   1  18   8 181  22  29   7   8 |   f = science
  16   9   7  48  27  20 125  24  21  17 |   g = maheberawiguday
  12   3   0  43  12  26  23 170   4   4 |   h = tmhert
  13  31   4  23  22  10  35  17 142  22 |   i = heg
  10  11   4  27  20  15  29   5  22 195 |   j = adega
```

Figure 21 confusion matrix for ten classes using Hyperpipes

As shown from the above figure confusion matrix details 1,944 news items out of 3,154 are correctly classified and the average percent accuracy is 61.636 percent. From the confusion matrix we can see that the highest confusion (69) happened between politica and economy followed by tmhert and tena(48). This shows that these classes have a lot in common each other.

Number of classes	Accuracy performance achieved
4	82.8 %
7	69.5%
10	61.636%

Table 6 Accuracy performance achieved at different levels of class using Hyperpipe algorithm

From the above table, the highest accuracy is 82.8% and the lowest is 61.636 % when the number of class is four and ten respectively. The seven class category yields an accuracy of 69.5%. From this we can conclude that when the number of class increase the accuracy goes in a reverse way.

5.2.3. Radial basis function network (RBF Network) Test

Experiment on four categories

The first experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, and ‘tena’.

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Radial basis function network

Time taken to build model: 0.22 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	1030	82.4 %
Incorrectly Classified Instances	220	17.6 %
Total Number of Instances	1250	

```
=== Confusion Matrix ===
   a   b   c   d   <-- classified as
 246  28   5  13 |   a = economy
  93 167  28  13 |   b = politica
   2   4 318  11 |   c = sport
  13   8   2 299 |   d = tena
```

Figure 22 confusion matrix for four classes using RBF Network

Correctly classified news items are 1,030 out of 1251and the average accuracy is 82.4%. The confusion matrix shows that the highest confusion (93) happened between politica and economy. This shows that these classes are more related.

Experiment on seven categories

The second experiment was performed on seven categories: ‘economy’, ‘politica’, ‘sport’, ‘tena’, ‘bahelnaturism’ ‘science’, and ‘maheberawiguday’.

Time taken to build model: 2.2 seconds

=== Stratified cross-validation ===

=== Summary ===

```
Correctly Classified Instances        1501               68.2273 %
Incorrectly Classified Instances       699               31.7727 %
Total Number of Instances             2200

=== Confusion Matrix ===
   a   b   c   d   e   f   g   <-- classified as
 206  28  12   8  30   4   5 |   a = economy
  62 157  14  16  37  13   1 |   b = politica
  17   7 193  12  65   1   6 |   c = science
  12   6   7 213  77  11   9 |   d = bahelnaturism
  17   8  19  15 220   7  28 |   e = maheberawiguday
   3   4   1  10  28 288   4 |   f = sport
   6   7  15   6  61   0 224 |   g = tena
```

Figure 23 confusion matrix for seven classes using RBF Network

As shown from the above confusion matrix details 1501 news items out of 2,200 are correctly classified and the average percent accuracy is 69.5 percent. Confusion matrix of the experimental result indicates that the highest confusion (62) happened between politica and economy followed by tena and maheberawiguday (61). This shows that these classes are more related.

Experiment on ten categories

The third experiment was performed on ten categories: 'economy', 'politica', 'sport', 'tena', 'bahelnaturism' 'science', 'maheberawiguday', 'tmhert','heg' and 'adega'.

Time taken to build model: 10.09 seconds

=== Stratified cross-validation ===

=== Summary ===

```
Correctly Classified Instances        1673               53.0438 %
Incorrectly Classified Instances      1481               46.9562 %
Total Number of Instances             3154

=== Confusion Matrix ===
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 179  34   2  14   8   4  29  14   7   1 |   a = economy
  49 143   6  13   7   8  34  13  25   3 |   b = politica
   3   6 274   4  12   2  18   9   5   2 |   c = sport
   7   4   0 227   2   0  29  40   4   9 |   d = tena
   5   4   6  18 204   1  63  26   6   2 |   e = bahelnaturism
  21  15   4  21  18  53  45 107  13   4 |   f = science
  10   9   3  38  20   4 143  60  19   8 |   g = maheberawiguday
   3   4   0  30   5  26  20 203   5   1 |   h = tmhert
  13  19   1  15  24   7  92  23 114  11 |   i = heg
   6  22   1  38  18  10  48  29  33 133 |   j = adega
```

Figure 24 confusion matrix for ten classes using RBF Network

As shown from the above confusion matrix details 1673 news items out of 3154 are correctly classified and the average percent accuracy is 69.5%. From the confusion matrix the highest confusion (92) happened between heg and maheberawiguday followed by politica and economy (49). This shows that these classes have a lot in common.

Number of classes	Accuracy performance achieved
4	82.4 %
7	68.2273%
10	53.0438%

Table 7 accuracy performances achieved at different levels of class using RBF Network algorithm

From the above table, the highest accuracy is 82.4% and the lowest is 53.0438% when the number of class is four and ten respectively. The seven class category yields an accuracy of 68.2273%. From this we can say that when the number of classes increases the accuracy goes in a reverse way.

The above all experimental confusion matrix shows that politca and economy have a lot in common. This means these classes are more related followed by tena and maheberawiguday.

DISSCUSSIONS

The classification algorithms used in the experiments presented in this paper were implemented from WEKA package. These algorithms were chosen to include diverse set of paradigms, while high computational efficiency. Based on the above experimental results, we can clearly see that the highest accuracy is 83.44 % and the lowest is 82.4% when the number of class is four. The other algorithm yields an average accuracy of 82.8%. In fact, the highest accuracy belongs to the NaiveBayes classifier, followed by Hyperpipes and Radial basis function with a percentage of 82.8% and 82.4%.

5.3. Comparison of classification Algorithms

In this research different classification algorithms were used. The performances of each document classification algorithms: Naivebayes, Hyperpipe and RBF network were compared using their accuracy. Table below shows the comparison of the classification results obtained by Naivebayes, Hyperpipe and RBF network for the 4, 7 and 10 classes.

Number of classes	nave	Hyperpipe	RBF network
Four	83.44	82.8	82.4
Seven	71.636	69.5	68.2273
Ten	64.4261	61.636	53.0438

Table 8 performance evaluation at different class stages

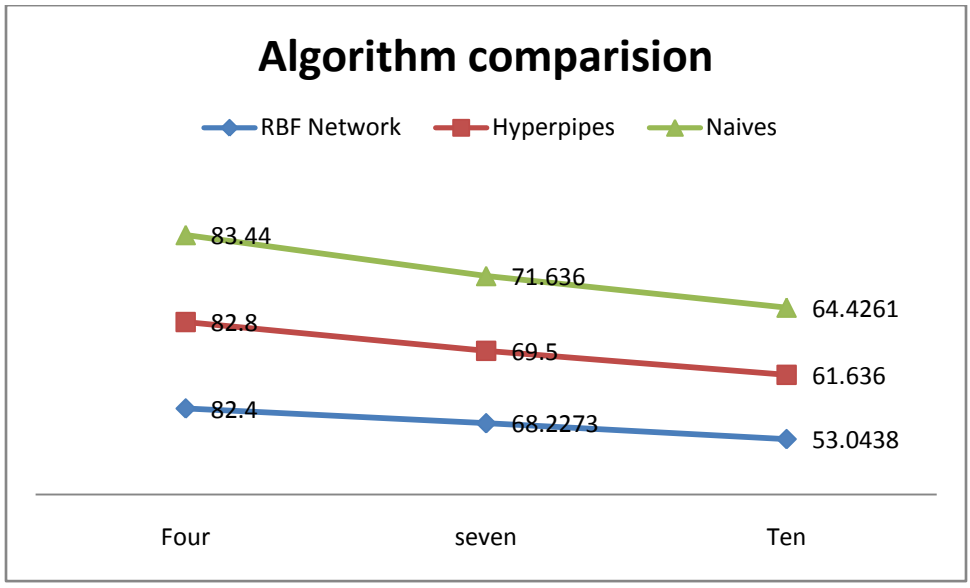


Figure 25 performance evaluation different classification algorithm and different class levels

As shown in Table and figure above, the Naivebays achieved the highest performance in terms of accuracy in all number of classes. However, the accuracy of all algorithms decreases as the number of categories increases.

In this simple experiment, from Figures, we can say Naivebays classifier requires the shortest time which is around 0.02 seconds compared to the others when class level is 10. Radial basis function (RBF) networks algorithm requires the longest model building which is 10.09 seconds.

5.1. Comparison of Supervised and Semi-supervised classification

Machine learning approaches	Number of classes	Naves	Hyperpipe	RBF network
Supervised	Four	76.48	74	72.16
	Seven	60.4545	53.2273	56.5
	Ten	69.7084	46.535	59.2662
Semi-Supervised	Four	83.44	82.8	82.4
	Seven	71.6364	69.5	68.2273
	Ten	55.4209	57.2618	51.9056

Table 9 performance comparison of semi-supervised and supervised performance

As shown experimental results above, all the results of semi-supervised classification approach is quite greater than supervised machine learning approaches. Therefore, this indicates that semi-supervised text classification is significantly better than supervised text classification.

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

In this study, the potential application of semi-supervised Learning approach for the classification of Amharic news documents was explored and is both feasible and important. The effect of the number of classes and the size of documents used on the performance and efficiency of classification algorithms was tested and compared using different data sets. Moreover, the performances of these classification algorithms were also tested at increasing number of classes using the same data set.

The study shows that semi-supervised approach for Amharic news classification significantly improve predictive accuracy over different classes, the semi-supervised approach was more successful (significantly better in for cases) than supervised and unsupervised approaches.

In this study, three classification algorithms including Naives bays, Hyperpipes and RBF Network are applied to Amharic news dataset.

Based on the experiments done in this study, the following concluding remarks were made.

- As the number of classes and documents increase, the accuracy produced by different classifiers, Naives bay's, Hyperpipes and RBF Network, become decreased and requires relatively high computational requirements. Moreover, it is learnt that considering categories with equal number of news items increases the performance of the classifiers.
- The best result obtained by the Naives bay's, Hyperpipes and RBF Network, classifiers is on four categories data (83.44% 82.8% and 82.4%) and the least performance is shown on the 10 categories data (64.4261% , 61.636, and 53.0438%) respectively. Compared to Hyperpipes and RBF Network Naives bay's classifier methods obtain a good result. Naives bay's shows that it can provide better results with larger training set. This paper indicated that naïve Bayes classifier is more applicable to Amharic news articles than Hyperpipes and RBF Network classifiers.
- All the classification algorithms: Naives bay's, Hyperpipes and RBF Network achieved better classification accuracy in semi-supervised than supervised. Therefore we can say that applying semi-supervised text classification better than supervised approach.

6.2. Recommendations

This study shows the potential application of semi-supervised machine learning techniques to the analysis of textual Amharic documents is both feasible and crucial. However, recommendations for further research are forwarded to improve the performance of document classification and to explore all algorithms and applications of semi-supervised document classification especially for local languages. Thus, the recommendations forwarded are organized as follows.

- ✓ The Naivesbayes, Hyperpipes and RBF Network classifiers used in this research have shown good accuracy. Therefore, there is a need to look for other classifiers with less processing cost and better accuracy.
- ✓ 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.
- ✓ As to the researcher's knowledge, there is no standard corpus open for researchers to apply different machine learning approaches. Researchers can devote much time on their work and explore more if standard corpus is prepared for Amharic classification experiments like 'Reuters-21578' for English.
- ✓ The bag of words representation approach which describes each document with its most significant terms was used in this thesis. However, future researchers may consider different document representation approaches such as phrase based and ontology based representations to select index or representative terms.
- ✓ Feature researchers can also compare the performance of semi-supervised learning approaches with the unsupervised approaches and two step approach using the same evaluation methods and document collections.
- ✓ Currently, few researches were conducted on automatic Amharic news classification and the results of the researches are promising. However, ENA still uses manual classification of News. So, it is better for the agency to review the different research works and to start the implementation of automatic classification of news.
- ✓ There is also a mismatch between the news categories provide by ENA and the clusters discovered by automatic clustering algorithms. Hence, it is better for the agency to revisit it news categories based on these findings.

- ✓ A number of researches were done on Amharic text document classification. However, as to the knowledge of the researcher, all the previous studies were conducted using Amharic text news item only. Future researchers can also explore document classification techniques to various real world problems such as classification and clustering of research papers and e- mail messages. Moreover, document classification and clustering techniques can also be extended to other local languages if huge collection of documents is available.

REFERENCES

Andreetto M., Zelnik Manor L., & Perona P. (2007). Non-parametric probabilistic image segmentation, in Proceedings of the International Conference on Computer Vision, pp.1–8.

Arzucan O. (2002), supervised and unsupervised machine learning techniques for text document categorization, Bo?gazi¸ci University.

Bar-Hillel, Hertz T., Shental N., & Weinshall D. (2005). Learning a mahalanobis metric from equivalence constraints, Journal of Machine Learning Research, vol. 6, pp. 937–965,.

Basu S., Bilenko M., & Mooney R. J. (2004). A probabilistic framework for semi-supervised clustering, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68.

Beletu R. (1982). A Graphemic Analysis of the Writing System of Amharic. Paper for the

Requirement of the Degree of bachelor of Art in Linguistics. Addis Ababa University.

Bennet K., Demiriz A., & Maclin R. (2002). Exploiting unlabeled data in ensemble methods, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 289–296.

Berkhin, P., (2002) Survey of clustering data mining techniques, Research paper, Accrue Software, http://www.accrue.com/products/researchpapers.html.

Bilenko M., Basu S., & Mooney R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering, in Proceedings of the International Conference on Machine Learning, vol. 69.

Bishop C. (2005). Neural Networks for Pattern Recognition. Oxford Univ.

Bishop C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Blei D. M. & Jordan M. I. (2004). Hierarchical topic models and the nested Chinese restaurant process, in Advances in Neural Information Processing Systems.

Blei D. M., Ng A. Y., & Jordan M. I. (2003). Latent dirichlet allocation, Journal of Machine Learning Research, vol. 3, pp. 993–1022.

Bradley C. L. (2002). 'Comparing supervised and unsupervised category learning': Psychonomic Bulletin & Review. University of Texas, Austin, Texas

Brandes U., Gaertler M., & Wagner D. (2003). Experiments on graph clustering algorithms, in Proceedings of the 11th European Symposiom of Algorithms, pp. 568– 579.

Brasethvik T. & Gulla J. A., (2001) Natural Language Analysis for Semantic Document Modeling. Data & Knowledge Engineering, 38, pages 45-62.

Breiman L. (1996). Bagging predictors, Machine Learning, vol. 24, no. 2, pp. 123–140.

Breiman L. (2001). Random forests, Machine Learning, vol. 45, no. 1, pp. 5–32,.

Breiman L. ,(1996). Bagging predictorsMachine Learning, vol. 24, no. 2, pp. 123–140.

Breiman L., Friedman J., Olshen R., & Stone C. (1984). Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks.

C. Apte, F. Damerau, & Weiss S. M. (1994). Automated Learning of Decision Rules for Text Categorization. ACM Transactions of Information Systems, 12(3), pages 233-251

Charniak E., (1997) Statistical Techniques for Natural Language Parsing. AI Magazine, 18(4), pages 33-44,

Cohen W. W. (1995). Fast effective rule induction, in Proc. of the 12th International Conference on Machine Learning, A. Prieditis and S. Russell, Eds. Tahoe City, CA: Morgan Kaufmann, July 9–12, pp. 115–123.

Cohen, W. W. & Singer Y. (1996). Context-sensitive learning methods for text categorization, Proceedings of the 19th Annual ACM SIGIR Conference.

Cohen, W. W. & Singer Y. (1996). Context-sensitive learning methods for text categorization. Proceedings of the 19th Annual ACM SIGIR Conference.

Comaniciu D. & Meer P. (2002). Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603–619.

Davis J., Kulis B., Jain P., Sra S., & Dhillon I.(2007). Information-theoretic metric learning, in Proceedings of the International Conference on Machine Learning, pp. 209– 216.

Demiroz G. & Guvenir H. A.(1997). Classification by voting feature intervals, in European Conference on Machine Learning, pp. 85–92.

Dempster P., Laird N. M., & Rubin D. B., (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: Series B, vol. 39, pp. 1–38.

Dempster, Laird N. M., & Rubin D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society: Series B, vol. 39, pp. 1–38.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). 'A Probabilistic Theory of Pattern Recognition'. Springer Verlag.

Duda, Hart P. E., & Stork D. G. (2000). Pattern Classification. Wiley-Inter science Publication.

Dumais, S., Platt T., Heckermann D., & Sahami M. (1998). Inductive learning algorithms and representations for text categorization, Proceedings of the Seventh International Conference on Information and Knowledge Management.

Figueiredo M. & Jain A. (2002). Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 381–396,.

Forgy, E., (1965) Cluster analysis of multivariate data: Efficiency versus interpretability of classification, Biometrics, Vol. 21, pp. 768–780.

Frank E. & Witten I. H. (1998). Generating accurate rule sets without global optimization, in ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 144–151.

Frank E., Wang Y., Inglis S., Holmes G., & Witten I. H. (1998). Using model trees for

classification, Machine Learning, vol. 32, no. 1, pp. 63–76.

Freund Y. & Mason L. (1999). The alternating decision tree learning algorithm, in Proc. 16th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA , pp. 124–133.

Freund Y. and Schapire R. E. (1996). Experiments with a new boosting algorithm in Proceedings of the International Conference on Machine Learning, pp. 148–156.

Freund Y. and Schapire R. E. (1998) Large margin classification using the perceptron algorithm, in Computational Learing Theory, pp. 209– 217.

Fuernkranz, J., Mitchell T., & Riloff E. (1998). A case study in using linguistic phrases for text categorization on the www, Sahami, M., editor, In Learning for Text Categorization: Papers from the 1998 AAAI Workshop (Technical Report WS-98-05).

Geberhiwot A. B. (2011). A two step approach for Tigrigna text categorization. (Master's Thesis. Department of Information Science, Addis Ababa University, Ethiopia).

Ghani R., Jones R., & Rosenberg C. (Eds.) (2003). 'ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining', Washington, DC.

Goldberger J., Roweis S., Hinton G., & Salakhutdinov R. (2005). Neighbourhood components analysis, in Advances in Neural Information Processing Systems, vol. 17, pp. 513–520.

Hagen L. & Kahng A., (1992). New spectral methods for ratio cut partitioning and clustering, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 11, no. 9, pp. 1074–1085.

Ham F. & Park S., (2002). A Robust Neural Network Classifier for Infrasound Events Using Multiple Array data. IEEE International Joint Conference NN, vol. 3, 2615-2619.

Hartigan, J., (1975). Clustering Algorithms, John Wiley & Sons, New York, NY,.

Heller K. & Ghahramani Z. (2005). Bayesian hierarchical clustering, in Proceedings of the

International Conference on Machine Learning, vol. 22, p. 297.

Hirotoshi T., (2002). 'Text Categorization using Machine Learning'. Dissertation Thesis. Department of Information Science, Nara.

Hoi S., Jin R., & Lyu M. (2007). Learning nonparametric kernel matrices from pairwise constraints, in Proceedings of the International Conference on Machine Learning, pp. 361–368.

Holmes G., Hall M., & Frank E. (1999). Generating rule sets from model trees, in AI '99: Proceedings of the 12th Australian Joint Conference on Artificial Intelligence. London, UK: Springer-Verlag, pp. 1–12.

Jaakkola T., & Haussler, D. (1999). 'Exploiting generative models in discriminative classifiers'. In Advances in Neural Information Processing Systems 11, pp. 487–493.

Jain K. & Dubes R. C. (1988). Algorithms for Clustering Data. Prentice Hall,.

Jain K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666.

Jain K., Murty M. N., & Flynn P. J. (1999). Data clustering: A review, ACM Computing Surveys, vol. 31, no. 3, pp. 264–323.

Jain, A. K., Murty M. N., & Flynn P. J. (1999). Data clustering: A review, ACM Computing Surveys, Vol. 31, No. 3, pp. 264–323.

Joachims T. (1998). Text categorization with support vector machines: Learning with many relevant features. European Conference on Machine Learning (ECML).

Kamvar S., Klein D., & Manning C. (2003). Spectral learning, in International Joint Conference On Artificial Intelligence, vol. 18, pp. 561–566, Citeseer.

Kang H. L., (2003) Text Categorization with a Small Number of Labeled Training Examples, A

Thesis Presented, School of Information Technologies University of Sydney, Sydney

Klien, Kamvar S. D., & Manning C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering, in Proceedings of the International Conference on Machine Learning.

Kohavi R. (1995). The power of decision tables, in ECML '95: Proceedings of the 8th European Conference on Machine Learning. London, UK: Springer-Verlag, pp. 174–189.

Kohavi R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 202–207.

Lakechew Y.(2011) Unsupervised Amharic news classification. (Master's Thesis, Department of Information Science, Addis Ababa University, Ethiopia).

Landwehr N., Hall M., & Frank E. (2005). Logistic model trees, Machine Learning, vol. 59, no. 1-2, pp. 161–205.

Lange T., Law M. H., Jain A. K., & Buhmann J. (2005). Learning with constrained and unlabelled data, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 730–737.

Law M. H. C. (2006). Dimensionality Reduction and Side Information. (PhD thesis, Michigan State University).

Lee J., Jin R., & Jain A. (2008). Rank-based distance metric learning: An application to image retrieval, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Liszka L. & Holmström M.(1999). Extraction of a deterministic component from ROSAT X-ray data using a wavelet transform and the principal component analysis. Astron. Astrophys. Suppl. Ser. , 140, 125-134.

Lu Z. & Leen T. (2005). Semi-supervised learning with penalized probabilistic clustering,in Advances in Neural Information Processing Systems, p. 849856.

MacQueen J. B. (1967). Some methods for classification and analysis of multivariate observations, in Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1. Berkeley, University of California Press, pp. 281–297.

Mallapragada P. K., Jin R., Jain A. K., & Liu Y. (2009) SemiBoost: boosting for semi-supervised learning., IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 11, p. 2000.

McCallum, A. & K. Nigam. (1998). A comparison of event models for naive bayes text classification, AAAI-98 Workshop on Learning for Text Categorization.

McLachlan G. L. & Basford K. E. (1987). Mixture Models: Inference and Applications to Clustering. Marcel Dekker.

McLachlan G. L. & Peel D. (2000). Finite Mixture Models. Wiley,.

Merkl D., (1998) Text Classification with Self-Organizing Maps: Some Lessons Learned. Neurocomputing, 21:1-3, pages 61-77.

Merkl. (1998). Text Classification with Self-Organizing Maps: Some Lessons Learned. Neurocomputing, 21:1-3, pages 61-77.

MitchellT. M. , (1997), Machine Learning, McGraw Hill.

Nega A. & Peter W. (2002).Stemming of Amharic Words for Information Retrieval. Literary Linguistic Computing Vol. 17, No.1,.

Ng,M., Jordan, & Weiss Y. (2002). On spectral clustering: Analysis and an algorithm, in Advances in Neural Information Processing Systems, vol. 2, pp. 849–856.

Nigam K., McCallum A. K., Thrun S., & Mitchell T. M., (2000). Text classification from labeled and unlabeled documents using EM, Machine Learning, vol. 39, no. 2/3, pp. 103–134.

Olivier C., Bernhard S. & Alexander Z.(2006). Semi-Supervised Learning. Cambridge, Massachusetts London, England

P. Tan, Steinbach M., & Kumar V. (2005). Introduction to Data Mining. Pearson Addison Wesley Boston.

Pavan Kumar Mallapragada, (2010), Some contributions to semi-supervised learning, a dissertation, Computer Science, Michigan State University

Porter, M. F.(1980). An algorithm for suffix stripping, Program, Vol. 14, pp. 130–137.

Quinlan J. R. (1986). Induction of decision trees. Machine Learning, vol. 1, no. 1, pp. 81–106.

Quinlan J. R.(1992) Learning with Continuous Classes. In 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348.

Quinlan R.(1993).C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.

Rosenberg, Hebert M., & Schneiderman H. (2005). Semi-supervised self-training of object detection models, in Proceedings of the Workshop on Applications of Computer Vision, vol. 1, pp. 29–36.

Rosenblatt F. (1988). The perceptron: A probalistic model for information storage and orginazation in the brain, Psychological Review, vol. 65, pp. 386–407, 1958, (Reprinted in Neuro-computing).

Roweis S. & Saul L. (2000). Nonlinear dimensionality reduction by locally linear embedding, Science, vol. 290, no. 5500, pp. 2323–2326.

Sahami, M. (1998). Using Machine Learning to Improve Information Access, Ph.D. thesis, Stanford University.

Sahami, M., Dumais S., Heckerman D., & Horvitz E. (1998). A bayesian approach to filtering junk e-mail. Sahami, M., editor, Proceedings of AAAI-98 Workshop on Learning for

Text Categorization.

Salton G., Yang C., & Wong A. (1975). A vector-space model for automatic indexing, Communications of the ACM, Vol. 18, No. 11, pp. 613–620.

Schmitter, E. D. (2006). Characterisation and Classification of Natural Transients, Transactions on Engineering, Computing and Technology, vol. 13.

Scudder H. J. (1965). Probability of error of some adaptive pattern-recognition machines, IEEE Transactions on Information Theory, vol. 11, pp. 363–371.

Sebastiani F., (2002) Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1), pages 1-4.

Shalev-Shwartz S., Singer Y., & Ng A. (2004). Online and batch learning of pseudometrics, in Proceedings of the International Conference on Machine Learning,.

Shental N., Bar-Hillel A., Hertz T., & Weinshall D. (2004). Computing gaussian mixture models with EM using equivalence constraints, in Advances in Neural Information Processing Systems.

Shi J. & J. Malik, (2000). Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888–905.

Silva V. De & Tenenbaum J. (2003) Global versus local methods in nonlinear dimensionality reduction, in Advances in Neural Information Processing Systems, pp. 721– 728.

Steinbach, M., G. Karypis, & V. Kumar, (1999) A comparison of document clustering techniques, KDD Workshop on Text Mining.

Stig-Erland H. (2007). Solving Classification Problems through Automatic Programming, (Master Thesis, Department of Computer Science, Østfold University College, Halden, Norway)

Szummer M. & Jaakkola T. (2001). Partially labeled classification with Markov random walks,

in Advances in Neural Information Processing Systems, pp. 945–952.

Teh Y., Jordan M., Beal M., & Blei D. (2006). Hierarchical dirichlet processes, Journal of the American Statistical Association, vol. 101, no. 476, pp. 1566–1581.

Vapnik V. (1998). The Nature of Statistical Learning Theory. Wiley-Interscience.

Wagstaff K. & Cardie C. (2000). Clustering with instance-level constraints, in Proceedings of the International Conference on Machine Learning, pp. 1103–1110.

Wagstaff K., Cardie C., Rogers S., & Schroedl S., (2001). Constrained k-means clustering with background knowledge, in Proceedings of the International Conference on Machine Learning, pp. 577–584.

Wang D. (1993). Pattern Recognition: Neural Networks in Perspective. Ohio State University.

Weinberger K., Blitzer J., & Saul L. (2006). Distance metric learning for large margin nearest neighbor classification, in Advances in Neural Information Processing Systems, vol. 18, p. 1473.

Witten I. H. & Frank E. (2005). Data mining: Practical Machine Learning Tools and Techniques (2nd Ed.), Morgan Kaufmann Publishers, San Mateo, CA.

Yang L., Jin R., Sukthankar R., & Liu Y. (2006). An efficient algorithm for local distance metric learning, in Proceedings of the National Conference on Artificial Intelligence, p. 543.

Yang, Y. & Liu X. (1999). A re-examination of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval , Berkeley, US.

Yang, Y. (1999) . An evaluation of statistical approaches to text categorization. Journal of Information Retrieval , Vol. 1, No. 1–2, pp. 69–90.

Yarowsky D. (1995). Unsupervised word sense disambiguation rivalling supervised methods, in Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, p. 189-196.

Yohannes A.(2007). Automatic Amharic text classification. (Master's Thesis, department of information science Addis Ababa university, Ethiopia).

Yu. (2005). Comparative Literary Style Mining between Native and Non-native English Writers. Graduate School of Library and Information Science, University of SIllinois.

Zelalem S. (2001). Automatic classifications of Amharic news items. (Master's Thesis , department of information science Addis Ababa university,Ethiopia).

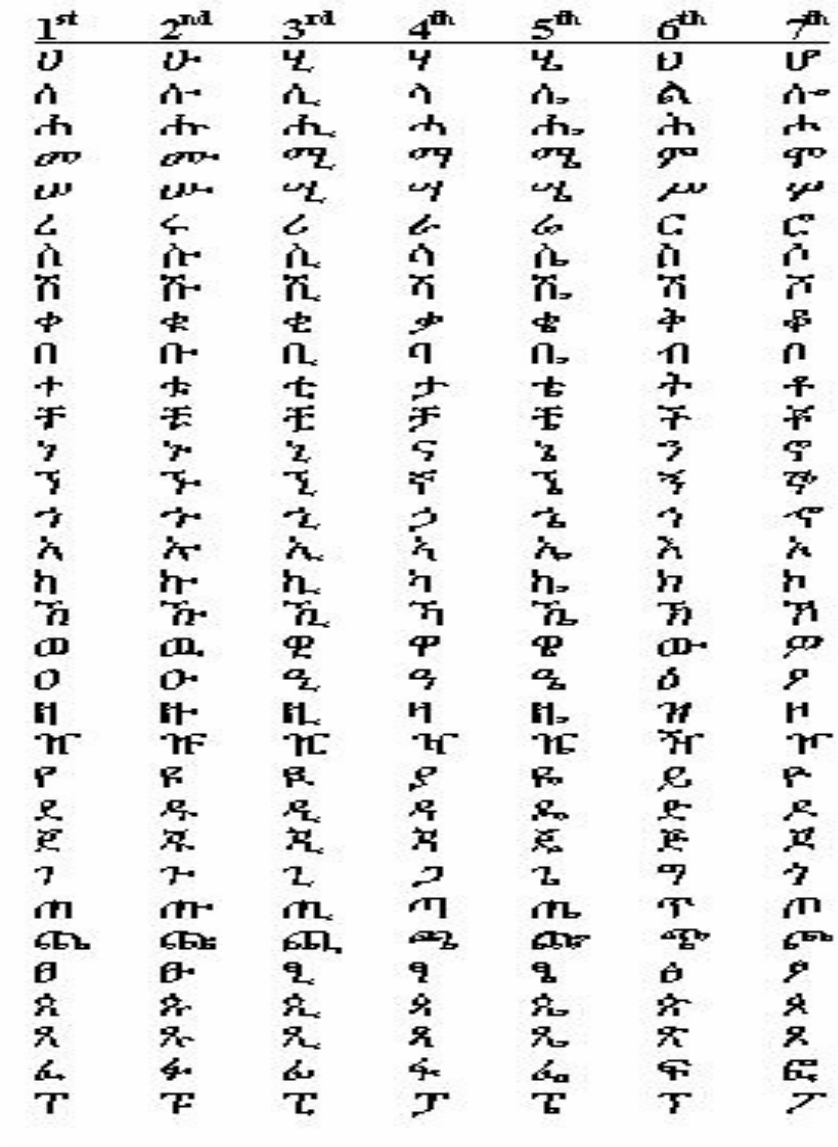
Zhao Q. & Miller D. (2005). Mixture modeling with pairwise, instance-level class constraints, Neural computation, vol. 17, no. 11, pp. 2482–2507.

Zhu & Ghahramani Z. (2002). Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107, Carnegie Mellon University

Zhu X., Ghahramani Z., & Lafferty J. (2003) Semi-supervised learning using gaussian fields and harmonic functions, in Proceedings of the International Conference on Machine Learning, pp. 912–919.

APPENDIXES

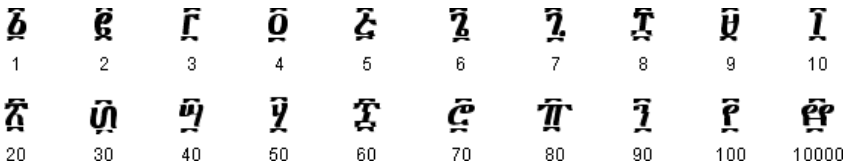
Appendix 1 Amharic alphabets



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

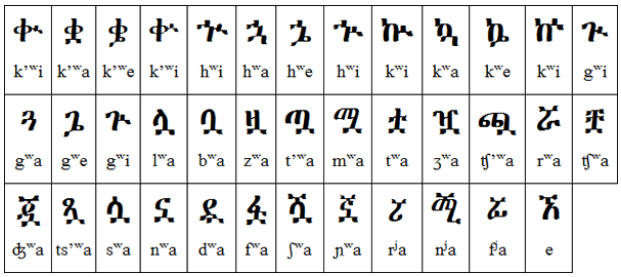
Shows Amharic characters

Appendix 2 Amharic number characters



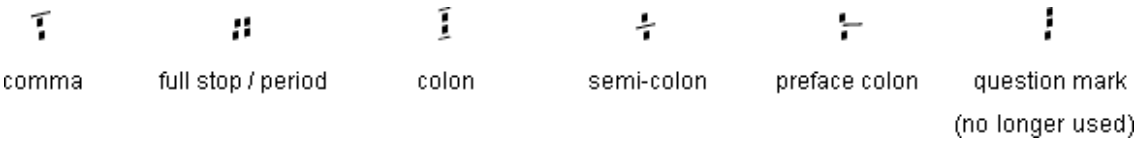
Shows Amharic number characters

Appendix 3 special Amharic characters



Shows special Amharic characters

Appendix 4 Amharic punctuation marks



Shows Amharic punctuation marks