Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science & Information Technology

A Comparative Study on Document Categorization Using Apriori Algorithm and Naive Bayse Classifier

Dissertation

Submitted to

Central Department of Computer Science & Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
For the Master's Degree in Computer Science & Information Technology

By
Sudan Maharjan
April, 2018

Supervisor
Asst. Prof. Sarbin Sayami

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science & Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...
Sudan Maharjan

Date:... ... ...

Supervisor's Recommendation

I hereby recommend that this dissertation prepared under my supervision by Mr.Sudan Maharjan entitled "**A Comparative Study on Document Categorization Using Apriori Algorithm and Naive Bayse Classifier** " in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

... ...
Asst. Prof. Sarbin Sayami
Central Department of Computer Science and Information Technology
Kirtipur, Nepal

Date: ... ...

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science & Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

... ...
Asst. Prof Nawaraj Poudel
Central Department of Computer Science
And Information Technology
Tribhuvan University
Kathmandu, Nepal
(Head)

... ...
Asst. Prof Sarbin Sayami
Central Department of Computer Science
And Information Technology
Tribhuvan University
Kathmandu, Nepal
(Supervisor)

... ...
(External Examiner)

Date: ... ...

... ...
(Internal Examiner)

Date: ... ...

ACKNOWLEDGEMENTS

I would first like to thank my respected thesis supervisor **Asst. Prof Sarbin Sayami**, Central Department of Computer Science and Information Technology, Tribhuvan University, Kathmandu Nepal. The door to Asst. Prof. Sayami office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

Secondly, I would like to thank my respected teacher Asst. Prof. Mr. Nawaraj Poudel, Head of Computer Science & IT Department Kirtipur, TU (Kathmandu, Nepal) for his guidance and encouragement. I would also like to express my gratitude to respected teachers Prof. Dr. Shashidhar Ram Joshi, Prof. Dr. Subarna Shakya, Prof. Sudarshan Karanjeet, Mr. Min Bahadur Khati, Mr. Bishnu Gautam, Mr. Jagdish Bhatt, Mr. Bikash Balami, Mr. Dhiraj Pandey, Mr. Arjun Singh Saud, Mrs. Lalita Sthapit and others staffs of CDCSIT for granting me broad knowledge and inspirations within the time of period of two years.

I would also like to thank my all close friends for the stimulating discussions and for all the fun during the period of two years.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Sudan Maharjan

ABSTRACT

Automatic document classification is the process for classifying electronic text document into specific category based on its contents. This dissertation work is about document classification and this dissertation will help in arranging electronic documents automatically. Document classification has many applications in computer science, information science, newspaper classification, library science etc. Document classification can be used in spam filtering, news article classification, pornography classification, indexing of documents, routing of emails etc.
The problem of automated document classification can be solved in supervised, unsupervised or semi-supervised machine learning technique. This dissertation work is based on both unsupervised and supervised machine learning technique where Apriori Algorithm is related to unsupervised machine learning and the Navie Bayes Classifier itself is supervised machine learning. The overall work of training and testing is based on three different classes of documents: Graphics, Guns and Sports. The system performance is measured on the basis of accuracy and F1 measure where Apriori Algorithm performed better than Naïve Bayes.

Keywords:
Automatic document classification, Supervised machine learning, unsupervised machine learning, semi-supervised machine learning, Apriori Algorithm, Naïve Bayes Classifier

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CM	Confusion Matrix
DT	Decision Tree
FE	Feature Extraction
GA	Genetic Algorithm
NB	Naive Bayes
NN	Neural Network
TC	Text Classification

Chapter 1

INTRODUCTION

1.1 Introduction

Document categorization is the process of automatically categorizing documents to one or more predefined categories. Document categorization was developed in early 60's. But it was widely known in early 90's [1]. And now a days, it is getting most challenging and widely researched area as numerous text documents are increasing in electronic form.

There are numerous text documents available in electronic form. More and more are becoming available every day [2]. In such a case, document categorization can be a challenging job. For this, automatic document categorization can be the best solution and without data mining techniques as such; is very much difficult to categorize the document into specific categories in less time and economically.

Basically there are two stages involved in document categorization. Training stage and testing stage. In training stage, the classifier is generated. For this, documents are preprocessed and are trained by a learning algorithm. In testing stage, a validation of classifier is performed. Many algorithms have been developed for automatic document classification. The most common techniques used for this purpose include Apriori Algorithm, NB Classifier, GA, Decision Tree and so on.

1.2 Applications of Document Categorization

Document categorization can be used in wide range. It has many applications.

1.2.1 Boost up the Internet Search

We can classify web pages into different categories to speed up the internet search, which is very useful for some search engines like Yahoo.

1.2.2 Span Filtering

Document categorization can be applied to spam filtering. A spam filter is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. It detects spam email messages by looking at the message header and content.

1.2.3 News Article Classification

It is also used for news agencies to classify articles into several categories like sports, politics, medical and etc by document categorization methods.

A news or media company will typically get hundreds and thousands of submissions every day. In order to efficiently handle such vast flow of information, there is a need of an automatic text classification system, which would categorize each document by topics.

1.2.4 Web Page Prediction

Text classification can be used to predict web page the user is likely to click on. Each hyperlink text description is treated as a miniature document.

Also a text categorization system could be used to naively predict the next page for a fast look-ahead caching system.

1.2.5 Pornography Classification

The exponential increase of information in internet has raised the issue of information security. Pornography web content is one of the biggest harmful resources that pollute the mind of children and teenagers and document categorization is the best solution of it.

1.3 Motivation

Document categorization is the one of the most challenging and widely researched area as numerous text documents are increasing in electronic form. As text documents are increasing day by day, it is more difficult to categorize each document into appropriate categories. Documents can be categorized into its appropriate categories either manually or automatically. Manual document categorization is suitable if the numbers of documents are less. But if the numbers of documents are higher, automatic document categorization should be done. In automatic document categorization, there has been lots

of research done. Though many researches has been carried out with respect to Naïve Bayes (NB)[3][4], K - Nearest Neighbor (KNN) [5], Apriori Algorithm, GA etc; no significant study has been conducted on comparing NB algorithm and Apriori algorithm for document categorization. Therefore, this dissertation provides insights between these two algorithms in terms of accuracy and F1-Measure.

1.4 Objectives

- To categorize documents into specific category like Graphics, Guns Sports using Apriori Algorithm and NB classifier.
- To perform cross fold validation and compare the result of Apriori Algorithm and NB classifier.

1.5 Problem Definition

The problem of document categorization is to determine the class of input document according to its content. The main task of document categorization is to categorize the document into predefined categories like jobs, sports, entertainment, politics, news etc. The system performs document classification on the basis of collections of important words in document. The classification task is carried out with Naive Bayesian and Apriori algorithm approach. Beside the text classification problem, it has further sub problems like tokenization, removing stop words, word stemming etc.

1.6 Outline of the Document

This Thesis is organized into six chapters including the introduction chapter. Introduction chapter simply contains introduction about the document categorization with objective of this dissertation work and problem definition.

The remaining part of the document is organized as follows:

Chapter 2 describes necessary background information and related work of document classification. It contains literature review section which verifies reviews the related topics. Literature review includes summary of definitions of NB categorization, GA, Apriori Algorithm, research on single document as well as in multi document.

Chapter 3 describes in detail the system model and the theoretical approaches for automated document classification problem. It includes tokenization, document preprocessing, stemming and classification methods.

Chapter 4 describes the implementation details of the system. All the methods described in the Chapter 3 are implemented for system evaluation.

Chapter 5 includes experimentation results and analysis of the systems.

Chapter 6 concludes the system performance and future directions.

Chapter 2

LITERATURE REVIEW AND BACKGROUND

2.1 Data Mining

The world is getting digital. Competition in market is getting higher and higher. In such case, a marketing manager working for a grocery store is not satisfied with just a list of all items sold, but wants a clear picture of what customers have purchased in the past as well as predictions of their future purchases. Data mining thus evolved to meet these increasing information demands [6].

Data mining emerged in the late 1980s, made great progress during the Information Age and in the 1990s [7]. Data Mining is defined as the process of extracting previously unknown, useful information from databases. It is also known as knowledge discovery i.e detecting something new from large scale or information processing [8]. In recent years data mining not only attracted business organizations, but also has been widely used in the information technology industry. Data mining is playing an important role in real world applications due to the availability of large amounts of data, and need to turn that data in to useful information.

2.2 Techniques in Data Mining

Data mining is the automated extraction of patterns representing knowledge implicitly stored in large databases, data warehouses and other massive information repositories. Some of the techniques adopted in data mining as given below:

2.2.1 Association Rule

In this technique, interesting association between attributes that are contained in a database are discovered which are based on the frequency counts of the number of items occur in the event (i.e. a combination of items), association rule tells if item X is a part of the event, then what is the percentage of item Y is also the part of event.

2.2.2 Clustering

Clustering is a technique used to discover appropriate groupings of the elements for a set of data. It is undirected knowledge discovery or unsupervised learning i.e, there is no target field and relationship among the data is identified by bottom-up approach.

2.2.3 Decision Trees

In this Decision Trees [9] technique, classification is performed by constructing a tree based training instance with leaves having class labels. The tree is traversed for each test instance to find a leaf, and the class of the leaf is predicted class. DT is a directed knowledge discovery in the sense that there is a specific field to predict the value.

2.2.4 Neural Networks

NN [10] is often represented as a layered set of interconnected processors. These processor nodes are frequently referred as neurons so as to indicate a relationship with the neurons of the brain. Each node has a weight connection to several other nodes in adjacent layers, each individual nodes take the received from connected nodes and use the weights together to compute output values.

2.2.5 Classification and Prediction

Classification is the technique in which set of documents are classified in the predefined category. Prediction is the process of predicting categorical class labels, constructing a model based on the training set and class labels in a classifying attribute.

2.3 Machine Learning

There are many well known data mining tasks, categorization is one among them on which this thesis concentrates. Categorization is a supervised machine learning technique [Margaret, 6].

“Optimizing a performance criterion using example data and past experience”, said by E. Alpaydin [11], gives an easy but faithful description about machine learning. In machine learning, data plays an indispensable role, and the learning algorithm is used to discover and learn knowledge or properties from the data. The quality or quantity of the dataset will affect the learning and prediction performance.

In general there are three different types of machine learning techniques. They are:

1. Supervised learning.
2. Unsupervised learning.
3. Semi-supervised learning

2.3.1 Supervised Learning

Supervised learning is a machine learning technique that learns from training data set. A training data set consists of input objects, and categories to which they belong. Assigning categories to input objects is carried out manually by an expert. Given an unknown object, supervised learning technique must be able to predict an appropriate category based on prior training.

2.3.2 Unsupervised Learning

In unsupervised classification, the set of possible classes is not known. After classification, name is assigned to that class. It is called clustering, where the classification is done entirely without reference to external information. It uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that data are organized by itself presented to the network and detects their emergent collective properties. The lack of direction for the learning algorithm in unsupervised learning can sometime be advantageous, since it lets the algorithm to look back for patterns that have not been previously considered [12].

2.3.3 Semi-Supervised Learning

In this classification, parts of the documents are labeled by the external mechanism. It learns with a small set of labeled examples and a large set of unlabeled examples i.e. learning with positive and unlabeled examples. Semi supervised learning methodology can deliver high performance of classification by utilizing unlabeled data [13].

2.4 Automatic Text Classification

Automatic text classification has always been an important application and research topic since the inception of digital documents. Today, TC is a necessity due to the very large amount of text documents that have to deal with daily. Dealing with unstructured text,

handling large number of attributes, examining success of preprocessing techniques , dealing with missing meta data and choice of a suitable machine learning technique for training a text classifier are major concerns of automatic text classification.

In general, TC includes topic based text classification and text genre-based classification. Topic-based text categorization classifies documents according to their topics [14]. Texts can also be written in many genres, for instance: scientific articles, news reports, movie reviews, and advertisements.

2.5 Automatic Text Classification Techniques

2.5.1 Categorization

Categorization is one of the most popular and familiar data mining techniques. Given a database D = {t_1, t_2, …, t n } of objects and a set of categories, C = { C_1, C2, …, Cn}, the problem of categorization is to define a mapping f: D—>C where each item ti is assigned to one category. A category C_j, contains only those objects mapped to it; that is,

$C_j = \{ t_i \mid f(t_i) = C_j , 1 \leq i \leq n \text{ and } t_i \in D \}$ [1].

Categorization can also be defined as "the task of learning a target function f that maps each object set x to one of the predefined class labels y" as shown in Fig 2.1. [1]

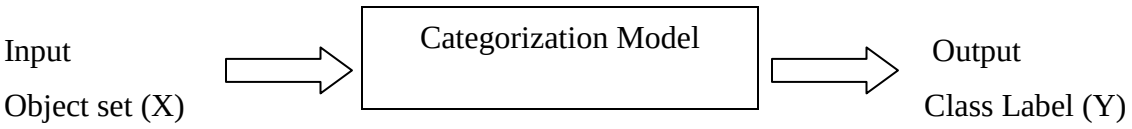


Fig 2.1 Categorization Mapping Input Object Set x to Class Label y.

2.5.2 Genetic Algorithm

The basic principles of GA [16] were first proposed by Holland in 1970 [17]. GA is inspired by the mechanism of natural election, a biological process in which stronger individuals are likely be the winners in a competing environment [17]. The strength of individuals is measured in terms of fitness of individuals where fitness is the positive value which is used to reflect the degree of goodness of the chromosomes for solving the problem. Throughout a genetic evolution, a fitter chromosome has the tendency to yield

good-quality offspring, which means a better solution to the problem. In GA, the solutions are called individuals or chromosomes [16].

The chromosome should in some way contain information about solution which it represents. The most used way of encoding is a binary string. The chromosome then will look like in Fig. 2.2:

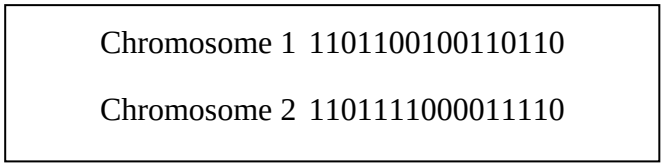


Fig. 2.2 Chromosomes in Binary

Operations in GA are:

2.5.2.1 Selection

To generate good offspring, a best parent selection mechanism is necessary. The goodness of each individual depends on its fitness [16].

2.5.2.2 Crossover

After selecting two best chromosomes, a crossover site along the bit strings is randomly chosen. The value of the two strings is exchanged up to this point. If S1 = 000000 and S2 = 111111 are two chromosomes and crossover point is 2 then S12 = 001111 and S21 = 110000 are two offspring generated after the crossover. By recombining portions of good individuals, the process is likely to create even better individuals.

2.5.2.3 Mutation

After the crossover, the offspring are muted. In mutation, the binary bit of the offspring is just altered in order to maintain diversity within the population and inhibit premature convergence.

2.5.3 Association Rule Mining

Association rule mining is a data mining task that discovers relationships among items in a transactional database [18]. Association rule miming finds all rules in the database that satisfy some minimum support and minimum confidence threshold [19]. It is described

as: Let I= { i_1, i_2,…i_m}, be a set of items. Let D, the task relevant data, be a set of database transactions where each transaction T is a set of items such that T $\subseteq$ I. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if A $\subseteq$ T. An association rule is an implication of the form A=> B, where A $\subset$ I, B $\subset$ I and A$\cap$ B = NULL [18].

Following key parameters are used to generate valuable rules:

- Support (s)

Support (s) of an association rule is the ratio (in percent) of the records that contain X$\cup$Y to the total number of records in the database: support(X => Y) = Prob { X $\cup$ Y }

- Confidence (c)

For a given number of records, confidence (c) is the ratio of the number of records that contain X$\cup$Y to the number of records that contain X.

Confidence(X => Y) = Prob {Y| X} = (support (X $\cup$ Y)) / (support (X))

- Strong Association Rules

Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called strong rules.

There are two main steps to process association rule mining:

- Find all frequent item sets

Every item set occurs at least more than the min_support value.

- Generate strong association rules from the frequent item sets

Generated rules must satisfy both min_support value and min_confidence value. When the support and confidence are greater than or equal to the pre-defined threshold minimum support and minimum confidence, the association rule is considered to be a valid rule [20].

2.5.4 Apriori Algorithm

Apriori invented by Rakesh Agarwal and Ramakrishnan Srikant in 1994 is a well known algorithm in data mining [1][21].Apriori employes an iterative approach known as a level-wise search [22][21], where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found. This set is denoted L_1. L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the database. An important

property called Apriori property, based on the observation is that, if an itemset I is not frequent, that is, P(I) < min_sup then if an item A is added to the itemset I, the resulting itemset (i.e., I ∪ A) cannot occur more frequently than I. Therefore, I ∪ A is not frequent either, that is, P(I ∪ A) < min_sup. In Apriori Algorithm, two-step process is followed, consisting of join and prune actions.

2.5.4.1 The Join Step

To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted by C_k. [23]

2.5.4.2 The Prune Step

Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset [23]. Hence if any (k-1)-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.

2.5.4.3 Algorithm

Input: Database, D;
minimum support threshold, min_sup.
Output: L, frequent itemsets in D.
Method:

(1) L_1=find_frequent_1-itemsets (D);
(2) **for** (k=2; Lk-1 ≠ Φ; k++){
(3) C_k = apriori_gen (L_{k-1}, min_sup);
(4) **for each** transaction t € D { // scan D for counts
(5) C_t = subset (C_k, t); // get the subsets of t that are candidates
(6) **for each** candidate c €Ct
(7) c.count++;
(8) }
(9) L_k = {c € Ck | c.count ≥ min_sup}
(10) }
(11) **return** L = $\cup_k L_k$;

procedure apriori_gen(L_{k-1}: frequent (k-1)-itemsets; min_sup: minimum support threshold)

(1) **for each** itemset l1 € L_{k-1}

(2) **for each** itemset l_2 € □ ł$_{k-1}$

(3) **if** ($l_1[1]=l_2[1]$). ($l_1[2]=l_2$ [2]) ...($l_1[k-2]=l_2[k-2]$).($l_1[k-1]<l_2$ [k-1])

then {

(4) c = l_1 | l_2; // join step: generate candidates

(5) **if** has_infrequent_subset (c,L_{k-1}) **then**

(6) **delete** c; // prune step: remove unfruitful candidate

(7) **else add** c to C_k;

(8) }

(9) **return** C_k;

procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}: frequent (k-1)-itemsets);

// use prior knowledge

(1) **for each** (k-1)-subset s of c

(2) **if** s ∉ L_{k-1} **then**

(3) **return** TRUE;

(4) **return** FALSE;

2.5.5 Naive Bayes Categorization

NB categorization is a simple probabilistic Bayesian categorization. It is based on Bayes' theorem of posterior probability [15]. It assumes that the effect of an attribute value on a given category is independent of the values of other attributes. This assumption is called conditional independence which was introduced to simplify complex computations involved, hence the name "naive" [1][2]. It exhibits high accuracy and speed when applied to large databases, and its performance is comparable with decision trees and neural networks [1][2].

2.5.5.1 Bayes Theorem

Let X be a data sample whose category is unknown. Let H be some hypothesis say data sample X belongs to a specified category C. For categorization problems one need to

determine P(H | X) the probability that the hypothesis H holds given the observed data sample X. Bayes theorem is given by:

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)}$$

Where P (H | X) is the posterior probability of H conditioned on X. For example, consider a data sample consisting of fruits described by their color and shape. Suppose that X is red and round, and that H is the hypothesis that X is an apple. Then P (H | X) implies that X is an apple given that, it is observed to be red and round. P (H) is the prior probability of H i.e. regardless of what the data sample looks; it is the probability that the given sample is apple. Posterior probability is based on information such as background knowledge rather than the prior probability which is independent of data sample X.

In the same way, P (X | H) is the posterior probability of X conditioned on H i.e. probability that X is red and round, and it is true that X is an apple. P (X) is the prior probability of X. It is the probability that a data sample from the set of fruits is red and round.

Given a large data sample, it would be difficult to calculate above probabilities. To overcome this difficulty, conditional independency was introduced.

2.5.5.2 Step wise Representation of NB Categorization

1. Initially each data sample is represented as a vector, $X = (x_1, x_2, , x_n)$ which are measurements made on the sample from n attributes, respectively, $A_1, A_2, , An$.

2. Suppose that there are m categories, C_1, C_2, C_m. If an unknown data sample X is given, then the categorization model will predict the correct category for X based on highest posterior probability, conditioned on X. NB categorization will assign unknown sample X to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \text{ for } 1 \leq j \leq m,\ j \neq i. \text{ Where,}$$

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)} \qquad \text{(By Bayes Theorem)}$$

3. As P(X) is constant for all classes, only P (X | C_i) P (C_i) need to be calculated. If the prior probabilities of categories are not known, then it can be assumed that all are equally likely i.e. P (C_1) = P (C_2)= …….= P (C_m).

Prior probabilities of categories can be calculated by P (C_j)= s_j / s , where s_j is the number of training samples of class C_j and s is the total number of training samples.

4. It is extremely expensive to compute P (X | C_i) for data sets with many attributes. In order to reduce this computation NB categorization assumes conditional independence. By this assumption values of the attributes are conditionally independent of one another given the category of the sample. There are no dependence relationships among the attributes. Thus,

P (X | C_i) = $\prod^{n}_{k=i}$ P (x_k | C_i).

5. If an unknown sample X is given then the NB categorization computes the value of P (X | C_i) P (C_i) for each category. Unknown sample X is then assigned to the category C_i if and only if

P (C_i | X)P (C_i) > P (C_j | X) P (C_j) for $1 \leq j \leq m, j \neq i$.

In other words categorization model maps sample X with the category C_i having maximum P (C_i | X)P (C_i) value.

Chapter 3

METHODOLOGY

3.1 Document Classification System Architecture

Document classification is divided mainly into five different sub-systems. They are data acquisition, pre processing, feature extraction, dimensionality reduction and document classification. Fig. 3.1 shows the Flow Chart of Automated Text Classification System.

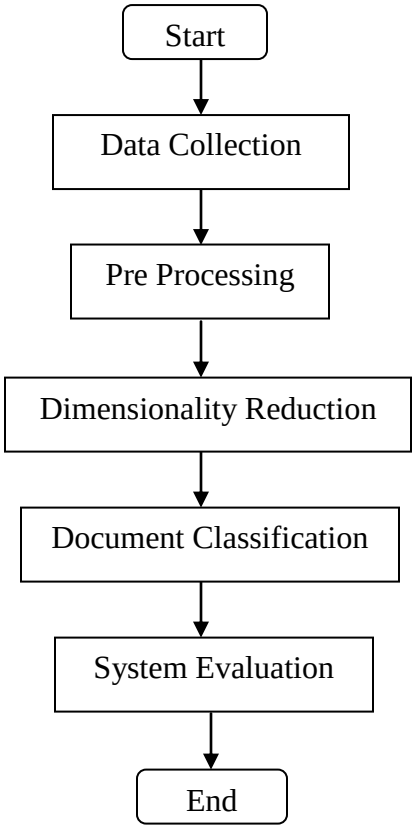


Fig. 3.1: Flow Chart of Automated Text Classification System

3.2 Data Collection

The data was collected from Enron Corpus. The Enron Corpus is a large database of over 600,000 emails generated by 158 employees. Three categories Graphics, Guns and Sports with total 165 documents were extracted to use in this dissertation. Each category contained equal number of documents i.e. 55.

3.3 Document Representation

Following Fig 3.2 is the format for documents.

```
Newsgroups: rec.sport.hockey
Path:
cantaloupe.srv.cs.cmu.edu!rochester!udel!wupost!cs.utexas.edu!utnut!
torn!pulp.cs.laurentian.ca!maynard
From: maynard@ramsey.cs.laurentian.ca (Roger Maynard)
Subject: Re: Superstars and attendance (was Teemu Selanne, was +/-
leaders)
Message-ID: <1993Apr5.194401.28354@ramsey.cs.laurentian.ca>
Organization: Dept. of Computer Science, Laurentian University,
Sudbury, ON
References: <1993Apr5.182124.17415@ists.ists.ca>
Distribution: na
Date: Mon, 5 Apr 1993 19:44:01 GMT
Lines: 62
In   <1993Apr5.182124.17415@ists.ists.ca>   dchhabra@stpl.ists.ca
(Deepak Chhabra) writes:
>Dean J. Falcione (posting from jrmst+8@pitt.edu) writes:
>>But I think the reason is Lemieux
>>had a 168 point season and was the first non-Gretzky to win the
Hart and
>>Ross since 1980.  People turned out to watch him play.
>I will grant that a star like Mario will draw fans, even if the
team sucks.
>But this is short term only; I still do not think the attendance
increase
>will last, unless the team is a winning/competitive/improving/butt-
kicking
>one.  Pittsburgh was still getting better, so people continued to
support
>them.  If they suddenly dropped to, say, 50 points, you'd have knee
surgery
>for some of the people jumping off the bandwagon.
--
Roger Maynard
maynard@ramsey.cs.laurentian.ca
```

Fig 3.2: A Screenshot of Category Sports

3.4 Document Preprocessing

Preprocessing method plays a very important role in text mining techniques and applications. It is the first step in the text mining process. In preprocessing, the whole document was chopped it up into pieces called tokens and unwanted and set of non-content-bearing functional words were removed.

Data pre-processing reduces the size of the input text documents significantly. It involves activities like tokenization, natural language specific stop-word elimination [24][25][26] and stemming [25] [27]. Document preprocessing mainly contain three steps. They are:

i. Tokenization

ii. Stop Word Removal

iii. Stemming

These are described briefly below.

3.5 Tokenization

Tokenization is the process of breaking parsed text into pieces, called tokens [18]. Every word in text is a token. During this phase text was lowercased and punctuations were removed. For example consider the sentence "Good players come up young, most players who come up young will be good." from a document that belong to category Sports tokenized as shown in Table 3.1.

Table 3.1: Tokenization Table

Good
Players
Come
Up
Young
Most
Players
Who
Come
Up
Young
Will
Be
Good

3.6 Stop Word Removal

Stop words are from articles (a, the), conjunctions (and, but), interjections (oh, but), prepositions (in, over), pronouns (he, it), and forms of the "to be" verb (is, are) [28] that does not help in deciding whether a document belongs to a category or not. They only make the text look heavier and less important for analysts. Stop words are high-frequency words of a language which rarely contribute to useful information in terms of document relevance and appear frequently in the text but provide less meaning in identifying the important content of the document. So to reduce the dimensionality, stop words were removed from the datasets. For this, list of stop words were collected and generated the stop words dictionary. The words from datasets that were found in the dictionary were removed. This helped in reducing the dimensionality of term space.

3.7 Stemming

Stemming is the process of reducing terms to their stems of root variant. For example "computer", "computing", "compute" is reduced to "comput" and "engineering", "engineered", "engineer" is reduced to "engine" [1]. In this dissertation, datasets contained different morphological variants of same root word. So to have efficient document categorization, these words were reduced to its single root word. Stemming process also helped to gain efficiency and effectiveness in categorization.

3.8 Build Inverted Index

Inverted Index is an index data structure storing a mapping from content, such as terms to its locations in a set of documents

3.9 Categorization using Apriori Algorithm

3.9.1 Generate Frequent Itemsets

Frequent itemsets were generated using Apriori Algorithm. Minimum support threshold is defined as 10 for Apriori Algorithm i.e. if itemset occurs in at least 10 documents then only it is considered as frequent itemset. The functioning of Apriori Algorithm is already explained in literature review.

3.9.2 Dimensionality Reduction

A major difficulty of text categorization problem using Apriori Algorithm is the high dimensionality of feature space i.e. total number of terms considered. Even for a moderate-sized text collection there are hundreds of thousands of unique terms [29]. So document frequency thresholding was used for reducing vocabulary in the collection as it is the simplest dimensionality reduction technique. For this, minimum threshold was set to three and the maximum threshold was set to hundred. Itemsets with document frequency less than three or greater than hundred were removed from the list.

3.9.3 Itemsets Categorization Method

Each frequent itemset were categorized into specific category using following formula:

$$W_{\pi j} = D\pi_j \cap DC_i / DC_i$$

where i = 1, 2, 3, 4, 5 categories, π_J is the itemset, C_i is the category and $W_{\pi J}$ is the weight factor of frequent itemset. Itemset π_J is mapped with category C_i based on the maximum value of $W\pi_J$.

The above formula was used to determine which itemsets fall into which categories.

3.9.4 Document Categorization Method

Finally, by determining the sum of weight factors for all itemsets of a given category, the document was categorized into particular category. Test document is associated with only that category which has maximum weight factor. The weight factor is calculated by

$$W_{cj} = \sum_{i=1}^{Cj} Wf_{\pi i}$$

Where $(\pi_i \in C_j) \wedge (\pi_i \subseteq D)$, for all j = 1, 2, 3, 4, 5 categories.

D is the set of significant terms obtained from the new test document. $Wf_{\pi j}$ is the weight factor of frequent itemsets.

3.10 Categorization using Naïve Bayes Classifier

3.10.1 Create Vector Table

Vector creation is an important factor for the NB classification system [30]. In vector table, the transaction database was represented in binary form of 0's and 1's. Each row

corresponds to a document id and each column corresponds to an item. If an item exists in a document then it is represented as '1' otherwise '0'. Example of Vector Table is shown in Table 3.2.

Table 3.2 Vector Table

TID	Word Attributes				
	W1	**W2**	**W3**	**W4**	**W5**
D1	1	1	0	1	0
D2	1	0	1	0	1
D3	0	1	1	1	0
D4	1	0	0	1	1

3.10.2 Create Occurrence Table

After creating vector table, occurrence table was created where occurrence of each item was counted. Each row corresponds to an item and each column corresponds to category with frequency of each item in that category. Example of Occurrence Table is shown in Table 3.3.

Table 3.3 Occurrence Table

Word Attributes	**Graphics**	**Sports**	**Guns**
W1	3	2	3
W2	2	2	4
W3	4	3	3
W4	2	4	2

3.10.3 Running the Naïve Bayes System

After building the word occurrence table successfully, NB system was used to classify a document to its specific category.

3.11 System Evaluation Measures

The correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples

that do not belong to the class (true negatives), examples that either were incorrectly assigned to the class (false positives) and examples that were not recognized as class examples (false negatives).

These four counts constitute a CM [31].

Measures for multi-class classification based on a generalization of the measures of binary classification for many classes Ci are given below. Where, tpi represent true positive for class Ci, fpi represent false positive for class Ci, fni represent false negative for class Ci, tni represent true negative for class Ci, and _ represent micro averaging.

3.11.1 Average System Accuracy

Average system accuracy evaluates the average per-class effectiveness of a classification system.

$$\text{Average Accuracy} = \sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_I + fn_i + fp_i + tn_i}$$

3.11.2 System Error

System error is the average per-class classification error of the system.

$$\text{Error Rate} = \sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_I + fn_i + fp_i + tn_i}$$

3.11.3 Precision

Precision (also called positive predictive value) is the number of correctly classified positive examples divided by the number of examples labeled by the system as positive.

Micro precision is the agreement of the data class labels with those of classifiers if calculated from sums of per-test decisions.

$$\text{Precision}_{\mu} = \sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}$$

3.11.4 Recall

Recall (also called sensitivity) is the number of correctly classified positive examples divided by the number of positive examples in the test dataset.

Micro recall is the effectiveness of a classifier to identify class labels if calculated from sums of per-test decisions.

$$\text{Recall}_{\mu} = \sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}$$

3.11.5 F1- Measure

The harmonic mean of precision and recall is called the F1-Measure. It is defined as [29]

$$\text{F1-Measure} = \frac{2}{1/\text{precision} + 1/\text{recall}}$$

Chapter 4

IMPLEMENTATION TOOLS AND TECHNIQUES

4.1 Implementation Tools and Techniques

All the algorithms of purposed prediction system were implemented in Microsoft Visual Studio 10 with .net framework 3.0 version installed on an Intel(R) Core(TM) i5 CPU M 480 @ 2.67 GHz, 2.67 GHz processor. The Computer has total main memory of 4 Gigabyte and 64-bit Operating system, x64-based processor and Microsoft Windows7 ultimate operating system installed in it.

4.2 Dot net Framework

The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework consists of the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. We can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, while also enforcing strict type safety and other forms of code accuracy that promote robustness. In fact, the concept of code management is a fundamental principle of the runtime. The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

4.3 Programming Language and IDE

4.3.1 C# Programming Language

C# is type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. We can use C# to create Windows client applications, web application and other various applications. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many

other tools to make it easier to develop applications based on the C# language and the .NET Framework.

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time.

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure, an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, when the C# program is executed, the assembly is loaded into the CLR, Then, and the CLR performs Just In Time compilation to convert the IL code to native machine instructions.

4.3.2 Visual Studio IDE

Microsoft Visual Studio is an IDE from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services

Visual Studio includes a code editor supporting IntelliSense (the code completion component). It support integrated debugger and Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhances the functionality at almost every level—including adding support for source-control systems (like Subversion). Visual Studio supports different programming languages and allows the code editor and debugger It support visual C#, VB.net (visual basic.net),F#,J#. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. The Visual Studio 2010 IDE was redesigned which, according to Microsoft, clears the UI organization and "reduces clutter and complexity. The new IDE better supports multiple document windows The IDE shell has been rewritten using the Windows Presentation Foundation .The new multi-paradigm ML-variant F# forms part of Visual Studio 2010. Visual Studio 2010 comes with .NET Framework 4 and support developing applications targeting Windows. It supports IBM DB2 and Oracle databases, in addition to

Microsoft SQL Server. It has integrated support for developing Microsoft Silverlight applications; including an interactive designer. Visual Studio 2010 offers several tools to make parallel programming simpler: Visual Studio 2010 includes tools for debugging. The new tools allow the visualization of parallel Tasks and their runtime stacks. Tools for profiling parallel applications can be used for visualization of thread wait-times and thread migrations across processor cores.

The Visual Studio 2010 code editor now highlights references; whenever a symbol is selected; all other usages of the symbol are highlighted. It also offers a Quick Search feature to incrementally search across all symbols in C++, C# and VB.NET projects. Quick Search supports substring matches and camel Case searches. The Call Hierarchy feature allows the developer to see all the methods that are called from a current method as well as the methods that call the current one. Visual Studio supports a consume-first mode which developers can opt into. In this mode, IntelliSense does not auto-complete identifiers; this allows the developer to use undefined identifiers (like variable or method names) and define those later. Visual Studio 2010 can also help in this by automatically defining them, if it can infer their types from usage. Current versions of Visual Studio have a known bug which makes IntelliSense unusable for projects using pure C (not C++).

Chapter 5

EXPERIMENTATIONS AND RESULTS

5.1 Cross Validation

Cross validation is the method of estimating expected prediction error. It helps selecting the best fit model. In this dissertation, 3 cross fold validation was carried out. First experiment contained 90.90% training datasets and 9.09% testing datasets i.e. 150 datasets for training purpose and 15 datasets for testing purpose. Similarly, second experiment contained 83.30% training datasets and 16.96% testing datasets i.e. 137 datasets for training and 28 datasets for testing purpose. And third experiment contained 80% training datasets and 20% testing datasets i.e. 132 datasets for training and 33 datasets for testing purpose.

5.2 Training and Testing Datasets

In this dissertation, data were collected for three different categories from Enron Corpus. Some of the collected datasets for three categories are presented below.

5.2.1 Sports

Sports class of text documents contains information about mainly two games. They are: hockey, baseball. Data sample for category Sports is presented in Fig. 5.1.

i. I disagree. McNall has demonstrated with Gretzky that a star brings out the crowds whether or not the team is expected to do well. Very few fans real-istically expect the Kings to do well this year although I do) and yet they still go out to see Gretzky. This is the marketing strategy - selling the game by selling the stars - that is employed by Baseball and, notably, the NBA and this is the attitude that the new Bettman/McNall leadership is bringing to the league.

ii. >In article <1993Apr14.015415.10176@mprgate.mpr.ca>, tasallot@galaxy.mpr.ca >(Mathew Tasalloti) says:>>chances this year), but it seems to me like Washington is the ONLY team that can stop the Penguins from winning their next Stanley Cup. > Really? I think both the Islanders and Devils would have a better chance >at the Penguins than the Capitals, IMO.

iii. In article <ekdfc.14.0@ttacs1.ttu.edu>, ekdfc@ttacs1.ttu.edu (David Coons) writes: >In article <1993Apr4.221228.17577@bsu-ucs> 00ecgillespi@leo.bsuvc.bsu.edu >writes:>>I AM DOING A POSTITION PAPER ON THE DESIGNATED HITTER RULE. ANY INFORMATION >>OR EVEN OPINIONS WOULD BE GREATLY APPRECITATED. 00ECGILLESPIE "MAGIC" >Should be rescinded. The rules say baseball is a game between two teams of >nine players each. Let's keep it that way. Last weeks Sports Illustrated has a couple of big articles on the designated

iv. In article <7862@blue.cis.pitt.edu> genetic+@pitt.edu (David M. Tate) writes: >mss@netcom.com (Mark Singer) said:>>I meant that one should not let the exception make the rule. >It's not an exception. Good players come up young; most players who come >up young will be good. This has always been the rule.

v. Being a baseball fan and a fan of the above mentioned band I was wondering if anyone could clue me in on whether the Dead (or members of) sang the national anthem at todays Giant opener? I would imagine that it is a bit too early for anyone to know, but an answer would be greatly appreciated.

Fig. 5.1: Sport Sample

5.2.2 Graphics

Graphics class of text documents contains information about graphics design, colors, graphic cards, display etc. Data sample for category Graphics is shown in Fig. 5.2.

i. I wrote something about making color modifications quickly >>>with 8bit quantized images and only at the saving the image to file >>>process we have to make the modifications to the 24bit image. >>>This makes sense, because the main use of XV is only viewing images.

ii. I've been away for a couple of weeks and have become out of touch with the latest information on the Diamond Viper Card. Does anyone know if Diamond has come out with any Vesa Driver updates lately? Also, I was wondering what the latest Windows Driver version is up to now.

iii. : I've only had the computer for about 21 months. Is that a reasonable life : cycle for a LCD display? My Toshiba T1100+ LCD (CGA, 1986) died in 11 months. Replaced under the 12 month warranty, fortunately. When it died, it died instantly and completely.

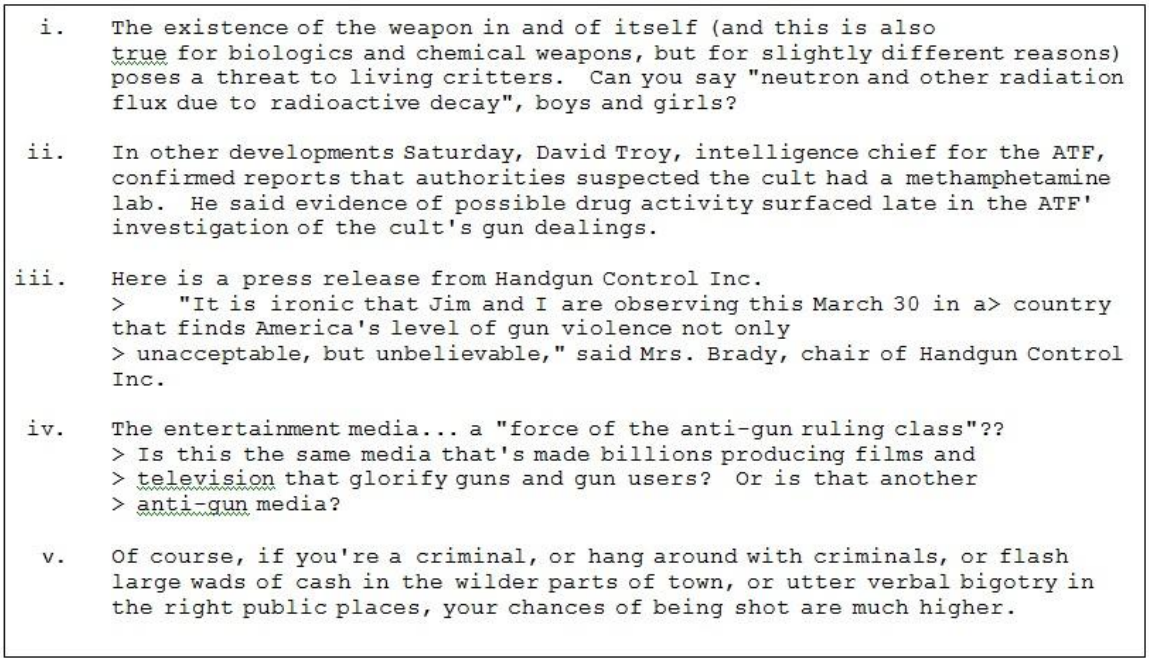
iv. I am trying to find a program which can run under the environment ULTRIX/X11R4 to plot surfaces and contour plots from a set of {x,y,z}. I would really appreciate any hint on the name of such a plotting program and where to find it.

v. I am happy to announce the first public release of the bit program, an INTERACTIVE, FULL COLOR image viewer and editor based on SGI GL. Besides typical touchup tasks, such as crop, rotate, smooth, etc, bit offers some unique features not available in similar programs, such as text and vector support and the separation of text and image.

Fig. 5.2: Graphics Sample

5.2.3 Guns

Guns class of text documents contains information about different weapons, crimes related with guns. Data sample for category Guns is shown in Fig. 5.3.



i. The existence of the weapon in and of itself (and this is also true for biologics and chemical weapons, but for slightly different reasons) poses a threat to living critters. Can you say "neutron and other radiation flux due to radioactive decay", boys and girls?

ii. In other developments Saturday, David Troy, intelligence chief for the ATF, confirmed reports that authorities suspected the cult had a methamphetamine lab. He said evidence of possible drug activity surfaced late in the ATF' investigation of the cult's gun dealings.

iii. Here is a press release from Handgun Control Inc.
> "It is ironic that Jim and I are observing this March 30 in a> country that finds America's level of gun violence not only
> unacceptable, but unbelievable," said Mrs. Brady, chair of Handgun Control Inc.

iv. The entertainment media... a "force of the anti-gun ruling class"??
> Is this the same media that's made billions producing films and
> television that glorify guns and gun users? Or is that another
> anti-gun media?

v. Of course, if you're a criminal, or hang around with criminals, or flash large wads of cash in the wilder parts of town, or utter verbal bigotry in the right public places, your chances of being shot are much higher.

Fig. 5.3: Gun Sample

5.3 Training Datasets:

Three datasets; Training Dataset1, Training Dataset2 and Training Dataset3 are used for training the system. Statistics for all datasets are presented below.

5.3.1 Training Dataset1

Statistics about Training Dataset1 are given in the Table 5.1. Here total 150 documents were used in training purpose where each category contained 50 documents.

Table 5.1: Training Dataset1

Categories	No. of samples
Sports	50
Graphics	50
Guns	50
Total	150

5.3.2 Training Dataset2

Statistics about Training Dataset2 are given in the Table 5.2. For 2nd training, Training Dataset2 contained total 137 documents where category Sports contained total 47 documents, category Graphics contained total 45 documents and category Guns contained total 45 documents.

Table 5.2: Training Dataset2

Categories	No. of samples
Sports	47
Graphics	45
Guns	45
Total	137

5.3.3 Training Dataset3

Statistics about Training Dataset3 are given in the Table 5.3. In Training Dataset3, total 132 documents were used for training purpose. Here category Sports contained total 45 documents, category Graphics contained total 44 documents and category Guns contained total 43 documents.

Table 5.3: Training Dataset3

Categories	No. of samples
Sports	45
Graphics	44
Guns	43
Total	132

5.4 Testing Datasets

For testing also, three testing datasets; Testing Dataset1, Testing Dataset2 and Testing Dataset3 were used in the system. Statistics for all datasets are presented below.

5.4.1 Testing Dataset1

Statistics about Testing Dataset1 are given in the Table 5.4. In Testing Dataset1, total 15 documents were used for testing purpose where each category contained total 5 documents.

Table 5.4: Testing Dataset1

Categories	No. of samples
Sports	5
Graphics	5
Guns	5
Total	15

5.4.2 Testing Dataset 2

Statistics about Testing Dataset2 are given in the Table 5.5. Testing Dataset2 contained total 28 documents for testing purpose where category Sports contained total 8 documents, category Graphics contained total 10 documents and category Guns contained total 10 documents.

Table 5.5: Testing Dataset2

Categories	No. of samples
Sports	8
Graphics	10
Guns	10
Total	28

5.4.3 Testing Dataset 3

Statistics about Testing Dataset3 are given in the Table 5.6. Testing Dataset3 contained total 33 documents for testing purpose where category Sports contained total 10 documents, category Graphics contained total 11 documents and category Guns contained total 12 documents.

Table 5.6: Testing Dataset3

Categories	No. of samples
Sports	10
Graphics	11
Guns	12
Total	33

5.5 Data Dictionaries

5.5.1 Stop Word Dictionary

Stop words are common words from articles (a, the), conjunctions (and, but), interjections (oh, but), prepositions (in, over), pronouns (he, it), and forms of the "to be" verb (is, are). These are less informative and un-useful words. So they were removed from the documents in pre-processing stage. In stop word dictionary, common and un-useful words were listed and excluded all those words from documents which were matched with the words that are listed in stop word dictionary. Some of the stop words from stop word dictionary are listed in Fig. 5.4.

a, about, above, across, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, am, among, amongst, amount, an, and, another, any, anyhow, anyone, anything, anyway, anywhere, are, around, as, at, back, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, bill, both, bottom, but, by, call, can, cannot, cant, co, computer, con, could, couldnt, cry, de, describe, detail, do, done, down, due, during, each, eg, eight, either, eleven, else, elsewhere, empty, enough, etc, even, ever, every, everyone, everything, everywhere, except, few, fifteen, fify, fill, find, fire, first, five, for, former, formerly, forty, found, four, from, front, full, further, get, give, go, had, has, have, he, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, him, himself, his, how, however, hundred, i, ie, if, in, inc, indeed, interest, into, is, it, its, itself, keep, last, latter, latterly, least, less, ltd, made, many, may, me, meanwhile, might, mill, mine, more, moreover, most, mostly, move, much, must, my, myself, name, namely, neither, never, nevertheless, next, nine, no, nobody, none, nor, not, nothing, now, nowhere, of, off, often, on, once, one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own, part, per, perhaps, please, put, rather, re, same, see, seem, seemed, seeming, seems, serious, several, she, should, show, side, since, sincere, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such, system, take, ten, than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thick, thin, third, this, those, though, three, through, throughout, thru, thus, to, together, too, top, toward, towards, twelve, twenty, two, un, under, until, up, upon, us, very, via, was, we, well, were, what, whatever, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, will, with, within, without, would, yet, you, your, yours, yourself, yourselves

Fig.5.4: Stop Word Dictionary

5.5.2 Delimiter Dictionary

The delimiters that don't carry the special meanings were removed in the pre-processing stage. Some of the delimiters are given in Fig 5.5.

```
',', ';', '.', '?', '_', '-', '!', '@', '*', '\\', '/', '%', '(', ')', '[',
']', '{', '}', '|', '~', '\n
```

Fig. 5.5: Delimiters

5.6 Experimentation Results

According to collected datasets, system was trained and tested. This section evaluates the results and the analysis of the outcomes. Various performance matrices were evaluated. In three different experiments, the system was trained and tested.

5.6.1 Experiment 1

Table 5.7 Confusion Matrix for Apriori Algorithm (Experiment 1)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	5	0	0
Guns	0	5	0
Sports	1	1	3

Table 5.8 Confusion Matrix for Naïve Bayes (Experiment 1)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	5	0	0
Guns	1	4	0
Sports	1	1	3

In Experiment 1, 90 % of the total documents were used for training and only 10 % of total documents were used for testing. Table 5.7 and Table 5.8 show the CM for Apriori Algorithm and for NB Classifier respectively.

From Experiment 1, the document categorization from Apriori Algorithm performed better than document categorization from NB classifier in the basis of accuracy and F1-Measure. The performed output of this first experiment is shown in Table 5.9.

Table 5.9 Experiment Result (Experiment 1)

Algorithm	Avg. Sys. Acc.(%)	Error(%)	Precision(%)	Recall(%)	F1-Measure(%)
Apriori Algorithm	86.66	13.34	88.88	86.66	87.75
Naïve Bayes	80	20	83.80	80	81.56

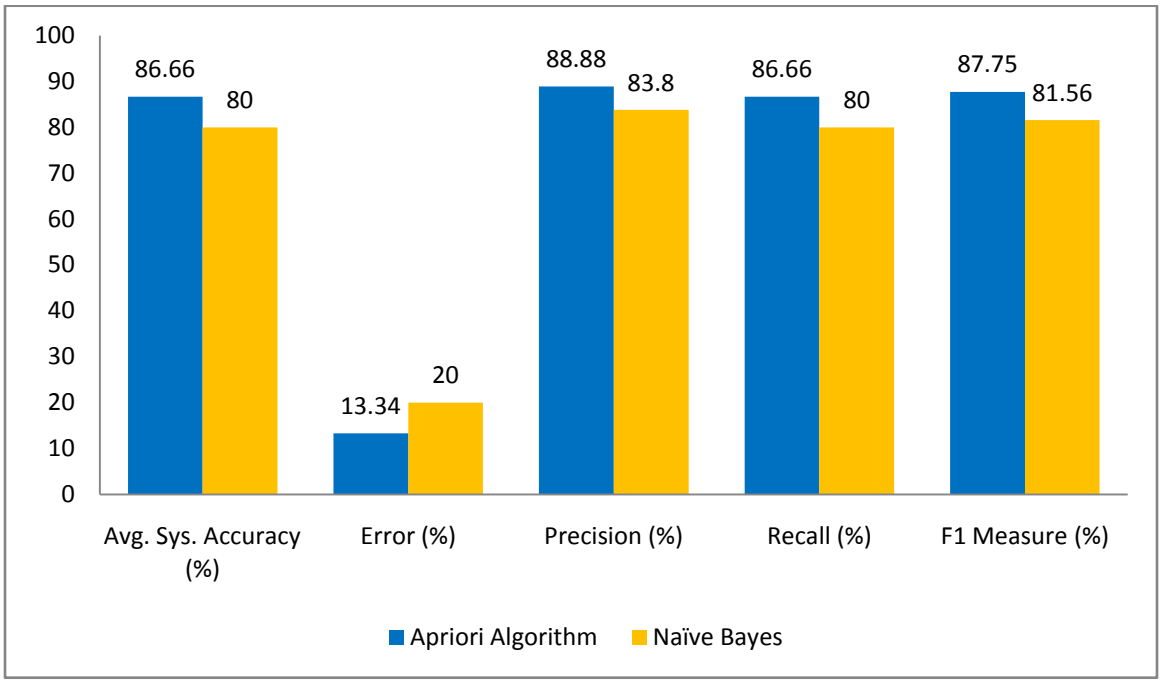


Fig: 5.6: Bar Chart of Experiment 1

Table 5.9 and the Fig. 5.6 show the result of Experiment 1 in tabular and chart form respectively. From the result, it is clear that Apriori Algorithm has high system Accuracy and F1-Measure than that of NB classifier. Apriori Algorithm system has system Accuracy rate of 86.66%, system Error rate of 13.34%, Precision ate of 88.88%, Recall rate of 86.66% and F1-Measure rate of 87.75%. Similarly, NB classification system has the system Accuracy rate of 80%, system Error rate of 20%, Precision rate of 83.80%, Recall rate of 80% and F1-Measure rate of 81.56%.

5.6.2 Experiment 2

Table 5.10 Confusion Matrix for Apriori Algorithm (Experiment 2)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	10	0	0
Guns	0	10	0
Sports	0	4	4

Table 5.11 Confusion Matrix for Naïve Bayes (Experiment 2)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	10	0	0
Guns	1	8	1
Sports	0	3	5

In Experiment 2, the 83 % of the total documents were used for training and only 17 % of total documents were used in testing. Training dataset for Experiment 2 is shown in Table 5.2. Similarly, testing dataset for this experiment is show in Table 5.5.

Table 5.10 shows the CM for Apriori Algorithm and Table 5.11 shows the CM for NB Classifier.

In Experiment 2, the document categorization from Apriori Algorithm performed better than document categorization from NB classifier in the basis of Accuracy and F1-Measure. The performed output of this Experiment 2 is shown in Table 5.12.

Table 5.12 Experiment Result (Experiment 2)

Algorithm	Avg. Sys. Acc.(%)	Error(%)	Precision(%)	Recall(%)	F1-Measure(%)
Apriori Algorithm	85.75	14.24	90.40	83.33	86.72
Naïve Bayes	82.14	17.86	82.31	80.80	81.54

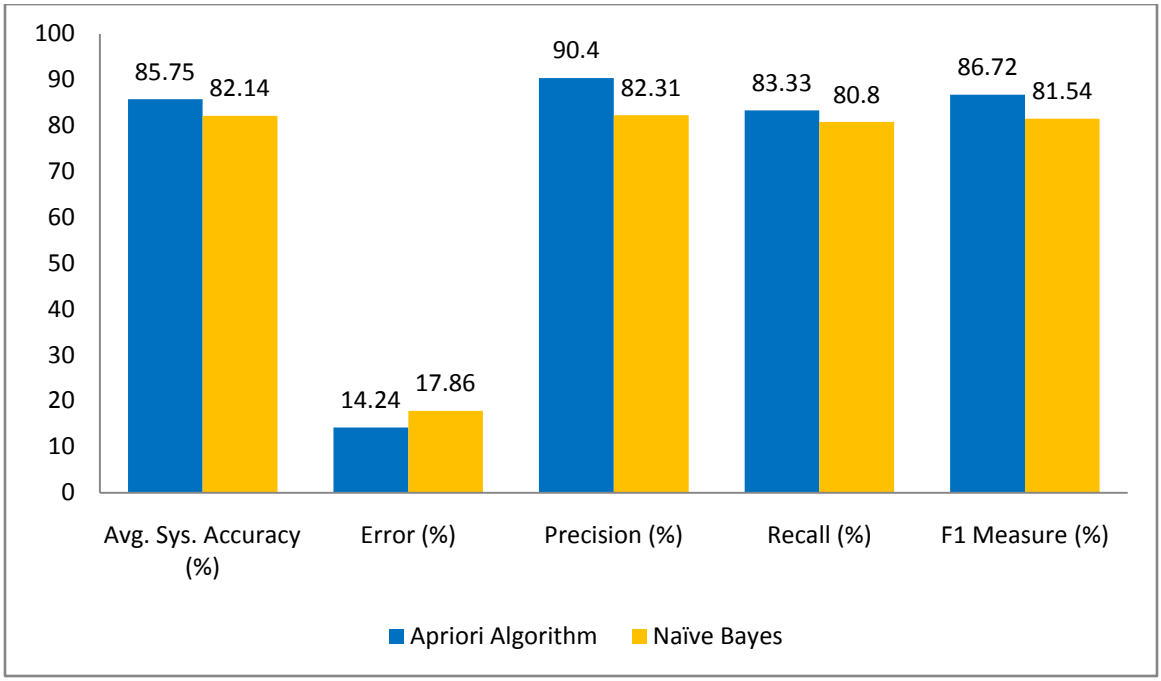


Fig. 5.7: Bar Chart of Experiment 2

Table 5.12 and the Fig. 5.7 show the result of Experiment 2 in tabular and chart form. From the result, it is clear that Apriori Algorithm has high system Accuracy and F1-Measure than that of NB classifier. Apriori Algorithm system has system Accuracy rate of 85.75%, system Error rate of 14.24%, Precision rate of 90.40%, Recall rate of 83.33% and F1-Measure rate of 86.72%. Similarly, NB classification system has the system Accuracy rate of 82.14%, system Error rate of 17.86%, Precision rate of 82.31%, Recall rate of 80.80% and F1-Measure rate of 81.54%.

5.6.3 Experiment 3

Table 5.13 Confusion Matrix for Apriori Algorithm (Experiment 3)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	7	4	0
Guns	0	12	0
Sports	4	0	6

Table 5.14 Confusion Matrix for Naïve Bayes (Experiment 3)

Actual \ Predicted	Graphics	Guns	Sports
Graphics	7	3	1
Guns	0	11	1
Sports	0	3	7

In Experiment 3, the 80 % of the total documents were used for training and only 20 % of total documents were used in testing. Training dataset for Experiment 3 is shown in Table 5.3. Similarly, testing dataset for Experiment 3 is show in Table 5.6.

Table 5.13 shows the CM for Apriori Algorithm and Table 5.14 shows the CM for NB Classifier for Experiment 3.

In Experiment 3, the document categorization from Apriori Algorithm performed better than document categorization from NB classifier in the basis of Accuracy and F1-Measure. The performed output of Experiment 3 is shown in Table 5.15.

Table 5.15 Experiment Result (Experiment 3)

Algorithm	Avg. Sys. Acc.(%)	Error(%)	Precision(%)	Recall(%)	F1-Measure(%)
Apriori Algorithm	75.75	24.25	79.33	74.33	76.76
Naïve Bayes	75.75	24.25	80.33	74.66	77.39

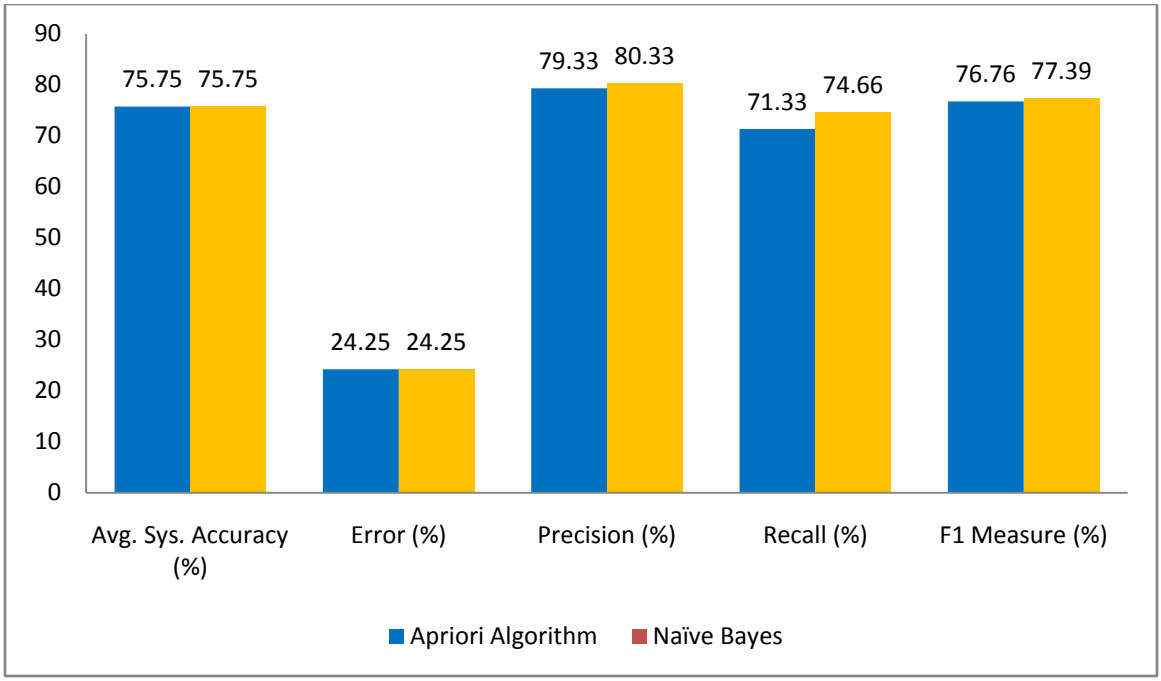


Fig. 5.8: Bar Chart of Experiment 3

Table 5.15 and the Fig. 5.8 show the result of Experiment 3 in tabular and chart form. In above analysis, Apriori Algorithm and NB classifier has same system Accuracy of 75.75%. Here, Apriori Algorithm system has system Error rate of 24.25%, Precision rate of 79.33%, Recall rate of 71.33% and F1-Measure rate of 76.76%. Similarly, NB classification system has the system Error rate of 24.25%, Precision rate of 80.83%, Recall rate of 74.66% and F1-Measure rate of 77.39%

5.7 Result Analysis

Aggregate results of all experiments are shown in Table 5.16.

Table 5.16 Aggregate System Result

Algorithm	Avg. Sys. Acc.(%)	Error(%)	Precision(%)	Recall(%)	F1-Measure(%)
Apriori Algorithm	82.63	17.27	86.20	81.44	83.74
Naïve Bayes	79.29	20.71	82.14	78.48	80.16

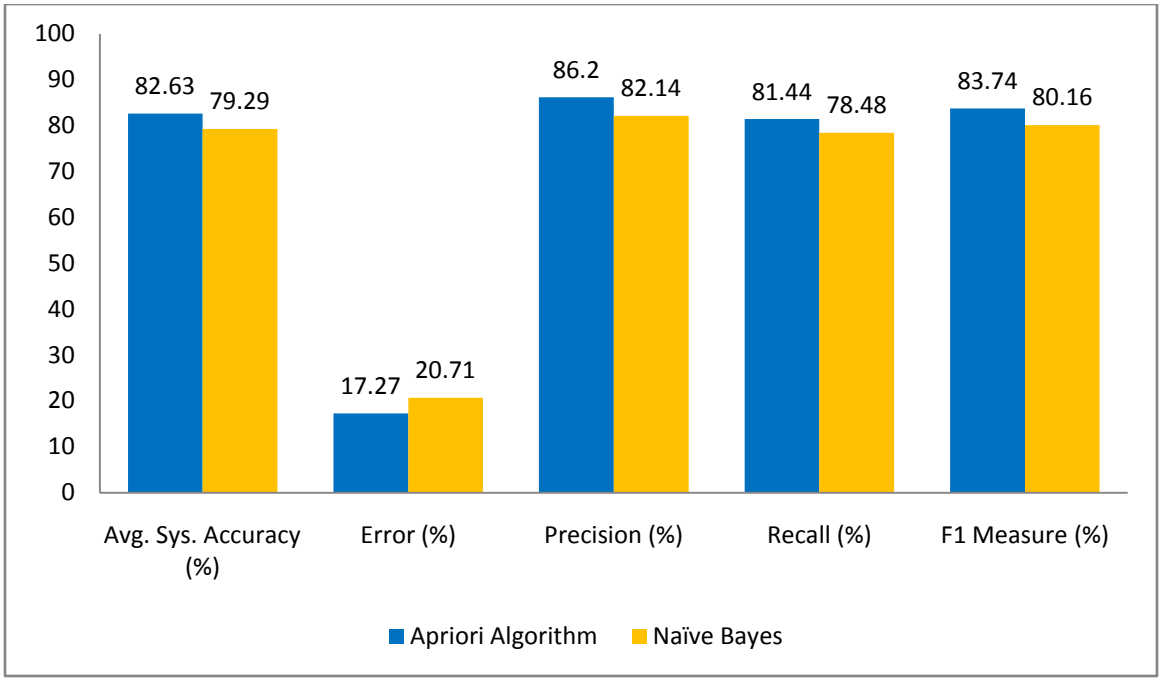


Fig. 5.9: Bar Chart of Result Analysis

From both tabular and chart analysis shown in Table 5.16 and Fig. 5.9, Apriori Algorithm has 3.44% less error rate than NB based classifier. Similarly, from F1-Measure Apriori Algorithm has 3.58% more performance than NB Classifier.

Likewise, Apriori Algorithm system has system accuracy rate of 82.63%, system error rate of 17.24%, precision rate of 86.24%, recall rate of 81.44% and F1-Measure rate of 83.74%. Similarly, NB classification system has the system accuracy rate of 79.29%, system error rate of 20.71%, precision rate of 82.14%, recall rate of 78.48% and F1-Measure rate of 80.16%.

Chapter 6

CONCLUSION

6.1 Conclusion

The premise of this dissertation was to classify electronic documents into its appropriate categories using NB and Apriori Algorithm. Here, the performance is measured for both techniques. Multiple unknown documents were categorized to its appropriate classes. Input text document was passed through various pre-processing steps like stop-word removal and stemming. Then, fine grained document was passed into classification systems-which are previously trained with given datasets and given classes in supervised manner.

On the basis of three cross fold validation, the system performance was measured. Over all experimentation results shows, Apriori Algorithm has higher classification accuracy and F1-Measure rate than NB based classifier. Apriori Algorithm system has the average system accuracy rate of 82.63%, system error rate of 17.27%, precision rate of 86.20% recall rate of 81.44% and F1-Measure rate of 83.74%. Similarly, NB classification system has the average system accuracy rate of 79.29%, system error rate of 20.71%, precision rate of 82.14% recall rate of 78.48% and F1-Measure rate of 80.16%.

6.2 Limitations and Future Scope

The performance of the system is limited due to minimum number of categories and less documents within each category respectively. Therefore, the performance of the system can be enhanced by increasing the number of documents and categories along. Similarly, enhancing data dictionaries can improve classification accuracy.

REFERENCES

[1] Prathima Madadi "Text Categorization Based on Apriori Algorithm's Frequent Itemsets", University of Nevada, Las Vegas - 5-1-2009

[2] S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan "Text Classification using Data Mining", ICTM 2005

[3] Kim S., Han K., Rim H., and Myaeng S. H. Some effective techniques for naïve bayes text classification. *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, no. 11, 2006.

[4] Meena M. J., and Chandran K. R. Naïve bayes text classification with positive features selected by statistical method. *In proceedings of the IEEE International conference on Advanced Computing,* 2009.

[5] Kataria Aman , M.D. Singh, A Review of Data Classification Using K-Nearest Neighbour *Algorithm, International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013

[6] Margaret H. Dunham, 'Data Mining Introductory and Advanced Topics', Chapter 1, 2 and 4, Southern Methodist University, Pearson Education Inc, 2003.

[7] Frans Coenen, 'Data Mining: Past, Present and Future', Knowledge Engineering Review, Vol. 00:0, 1–24.c©2004, Cambridge University Press DOI: 10.1017/S000000000000000 Printed in the United Kingdom

[8] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining concepts and Techniques" Chapter 8, Third Edition, Reprinted 2013

[9] Isa D., Lee L. H., Kallimani V. P., and RajKumar R.. Text document pre-processing with the Bayes formula for classification using the support vector machine, 2008

[10] Wang Z., He Y., and Jiang M. A comparison among three neural networks for text classification. *In proceedings of the IEEE 8th international conference on Signal Processing, 2006*

[11] E. Alpaydin, Introduction to machine learning, 2nd ed. Pg 4, The MIT Press, 2010.

[12] R. Sathya, Annamma Abraham, Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification, *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*, Vol. 2, No. 2, 2013

[13] V. Jothi Prakash, Dr. L.M. Nithya, A Survey On Semi-Supervised Learning Techniques, *International Journal of Computer Trends and Technology (IJCTT)* – volume 8 number 1–Feb 2014

[14] Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2):67–88, 1999.

[15] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining concepts and Techniques" *Chapter 8, Third Edition, Reprinted 2013*

[16] S. M. Khalessizadeh, R. Zaefarian, S.H. Nasseri, and E. Ardil "Genetic Mining: Using Genetic Algorithm for Topic based on Concept Distribution", *International Journal of Computer, Information Science and Engineering* Vol:2 No:1, 2008

[17] K. F. Man, *Member, IEEE,* K. S. Tang, and S. Kwong, *Member, IEEE* "Genetic Algorithms: Concepts and Applications", *IEEE Transactions on Industrial Electronics,* Vol. 43, No. 5, October 1996

[18] Miao, Yingbo B00181251, Wei, Gang B00344693, Yu, Zheyuan B00182683, Sheng, Xin B00140586 "The Implementation of Text Categorization with Term Association", Instructor: Dr. Vlado Keselj, Winter – 2003

[19] Bangaru Veera Balaji, Vedula Venkateswara Rao "Improved Classification Based Association Rule Mining", *International Journal of Advanced Research in Computer and Communication Engineering,* Vol. 2, Issue 5, May 2013

[20] Huan Wu, Zhigang Lu, Lin Pan, Rongsheng Xu, Wenbao Jiang "An Improved Apriori-based Algorithm for Association Rules Mining", *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*

[21] Goswami D.N., Chaturvedi Anshu., Raghuvanshi C.S. "An Algorithm for Frequent Pattern Mining Based On Apriori", *Goswami D.N. et. al. / (IJCSE) International Journal on Computer Science and Engineering* Vol. 02, No. 04, 2010, 942-947

[22] Chowdhury Mofizur Rahman, Ferdous Ahmed Sohel, Parvez Naushad, Kamruzzaman S M "Text Classification using the Concept of Association Rule of

Data Mining", *In Proceedings of International Conference on Information Technology*, Kathmandu, Nepal, pp 234-241, May 23-26, 2003.

[23] G.SenthilKumar, S.Baskar, M. Rajendran *"*Online Message Categorization Using Apriori Algorithm", *International Journal of Computer Trends and Technology*- May to June Issue 2011

[24] Kim S., Han K., Rim H., and Myaeng S. H. Some effective techniques for naïve bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, 2006.

[25] Zhang W., Yoshida T., and Tang X. Text classification using multi-word features. *In proceedings of the IEEE international conference on Systems, Man and Cybernetics, 2007.*

[26] Hao Lili., and Hao Lizhu. Automatic identification of stopwords in Chinese text classification. *In proceedings of the IEEE international conference on Computer Science and Software Engineering, 2008.*

[27] Porter M. F. An algorithm for suffix stripping. Program, 14 (3), 1980

[28] Aws Saad Shawkat & H K Sawant "Effective Content Based Data Retrieval Algorithm for Data Mining", *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, Volume 2, Issue 1 March 2012

[29] Yiming Yang, Jan O. Pederson, 'A comparative study on feature selection in text categorization', Proceedings of the fourteenth international conference on machine learning, pages: 412-420, 1997.

[30] Ishtiaq Ahmed, Donghai Guan, and Tae Choong Chung "SMS Classification Based on Naïve Bayes Classifier and Apriori Algorithm Frequent Itemset", *International Journal of Machine Learning and Computing*, Vol. 4, No. 2, April 2014

[31] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manage., vol. 45, pp. 427–437, jul 2009.