



**An Efficient Associative Classification Algorithm  
for Text Categorization**

**خوارزمية فعالة معتمدة على التصنيف الترابطي لتصنيف النصوص**

**By:**

**Bashar Suleiman Abdallah Aburumman**

**Supervisor:**

**Dr Fadi Fayez Abdel-jaber (Thabtah) .**

**A Thesis**

**Submitted in Partial Fulfilment of the Requirements for the  
Master Degree in Computer Information Systems**


**Faculty of Information Technology  
Middle East University**

**December , 2012**

## AUTHORIZATION FORM

اقرار تفويض

أنا الطالب بشار سليمان عبدالله أبورمان أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأفراد عند طلبها

التوقيع : 

التاريخ : ٢٠١٣-٤-3

## Authorization Statement

I (Bashar Suleiman Abdallah Aburumman) authorize the Middle East University to supply a copy of my thesis to libraries ,establishments or individuals upon their request .

Signature :



Date :3-4-2013

**Middle East University****Examination Committee Decision**

This is to certify the thesis entitled " **An Efficient Associative Classification Algorithm for Text Categorization** " was successfully defended on (22-1-2013) .

**Examination Committee Members**

Signature

Dr Fadi F Abdaljaber  
Associate Professor ,Department of Computer Science ,  
Faculty of Information Technology , Philadelphia University .

Prof Musbah M. Aqel  
Professor ,Department of Computer Science ,  
Faculty of Science and Information Technology , Al zarqa University .

Dr Mamoun K. Ahmad

Professor Assistant , Department of Computer Science ,  
Faculty of Information Technology, Middle east University .

## Declaration

I declare hereby that the present research work has been carried out by me under the supervision of Dr Fadi F Abdaljaber , and this work has not been submitted elsewhere for any other degree ,fellowship or any other similar title .

Signature :



Date :3-4-2013

Bashar Suleiman Abdallah Aburumman

Department of Computer Information System

Faculty of Information Technology

Amman ,Jordan

Email :bashar\_aburumman@yahoo.com

## **DEDICATION**

I dedicate this dissertation to my parents, who first planted the seeds of knowledge and wisdom in me. From my birth and throughout the development of my life, they have encouraged me with love and care to seek out knowledge and excellence. They motivated me to pursue my dreams, which led me to the completion of this endeavor.

**ACKNOWLEDGMENTS**

In the name of Allah the Most Gracious, the Most Merciful my guidance cannot come except from Allah, in Him I trust, to Him I repent, and Him I praise and thanks always go. I offer my sincerest gratitude to my advisor, Dr. Fadi Fayez Abdel-jaber (Thabtah) for his valuable contributions, knowledge, encouragement and helpful advices. As well as his vision that brought this work forward, for being there any time I knocked at his door. I wish him both more and more success and giving.

I would like to take a moment to thank my university "Middle East University", lecturers and employees, for the moral support and encouragement during my entire graduate studies.

**ABSTRACT****An Efficient Associative Classification Algorithm for Text Categorization****By Bashar Suleiman Abdallah Aburumman**

Text categorization (TC) is an interesting research area which attracted several researchers because of the large quantities of textual documents online and offline. TC concerns about the automatic classification of textual data to one or multiple classes based on their content keywords. Many different classification approaches were developed to categorize textual data, these approaches can be evaluated mainly by accuracy and the knowledge they produce. Moreover, these classification approaches range from high accuracy methods such as neural network to low ones such as Naïve bayes which means that some of them produce high accurate classifiers and others low accurate ones.

However, one fundamental measurement criteria is the understandability of the end-user of the resulting classifiers. Some classification approaches like neural network outputted accurate classifiers yet difficult to understand ones. A recently a new classification data mining technique called Associative Classification(AC) is developed which combines high accuracy and understandability output together based on association rule. AC is a high efficient method that builds more predictive and accurate classification systems than traditional classification methods such as probabilistic and The k-nearest neighbor algorithm according to many research experimental studies. AC produces rule's based classifiers on the form (if  $\rightarrow$  then ) that are easy to understand

and manipulate by end-user. This research is devoted to develop a new model based on AC for text categorization problem.

Mainly we focus on three main steps in the TC problem and these are: (1) Developing an efficient and fast intersection method based DiffSet of the Eclat method of association rule and adopting it to unstructured classification data. (2) Proposing a novel rule filtering procedure that reduces the number of rules in the outputted classifiers by considering partially matching during building the classifier. This novel method significantly minimized the number of rules described by the proposed model when compared with current AC models like MCAR. Lastly, (3) we improved the accuracy of the outputted classifiers by considering multiple rules in the classification step rather than most current AC algorithms that use only one rule to assign the test case a class value. Experimental results on real world textual data set called the Reuter indicated that the proposed model outperforms text categorization techniques either traditional techniques or (AC) techniques .



## الملخص

### خوارزمية فعالة معتمدة على التصنيف الترابطي لتصنيف النصوص

بشار سليمان عبدالله أبورمان

يعد تصنيف النص هو مجال بحث جذب اهتمام العديد من الباحثين نظرا لكميات كبيرة من الوثائق النصية على الانترنت. وهذا المجال يبحث في التصنيف الآلي للبيانات النصية إلى تصنيف واحد أو أكثر اعتمادا على الكلمات المفتاحية للمحتوى الخاص بهم. العديد من الطرق والتقنيات المختلفة طورت لتصنيف البيانات النصية، هذه الطرق غالبا تقيّم على اعتمادا على الدقة والمعرفة التي تنتجها. وعلاوة على ذلك، فإن طرق التصنيف هذه تتراوح في دقتها ما بين طرق عالية الدقة مثل (الشبكة العصبية) الى طرق منخفضة الدقة (بايز الغبي) وهذا يعني بعض هذه الطرق تنتج مصنفات عالية الدقة وأخرى تنتج مصنفات منخفضة الدقة.

ومع ذلك، احد معايير التقييم الأساسية هي القابلية للتفسير من قبل المستخدم للمصنفات الناتجة عن ذلك. فمثلا الشبكة العصبونية تعد احدى تقنيات التصنيف التي تنتج مصنفات عالية الدقة لكنها صعبة التفسير. مؤخرا ابتكرت تقنية جديدة للتقريب عن بيانات تسمى التصنيف المترابط والتي تجمع بين الدقة العالية وسهولة الفهم. هذه التقنية هي طريقة فعالة التي تبني نظما للتصنيف أكثر دقة وأفضل في التنبؤ من أساليب تصنيف التقليدية مثل (KNN) وفقا لكثير من الدراسات التجريبية البحثية. هذه الطريقة تنتج مصنفات تعتمد على قواعد من نوع (اذا-ثم) الى التي هي سهلة الفهم من قبل المستخدم. ويخصص هذا البحث لتطوير نموذج جديد لتصنيف النصوص يعتمد على التصنيف المترابط. ونحن نركز بشكل رئيسي على ثلاث خطوات رئيسية في تطوير هذا النموذج وهي: (1) تطوير طريقة التقاطع (DiffSet) القائمة على كفاءة وسرعة على طريقة (Eclat) واعتماده لتصنيف البيانات غير المنظمة. (2) اقتراح طريقة جديدة لتقليل عدد القواعد المستخدمة لبناء المصنفات من بناءا على المطابقة الجزئية خلال عملية بناء المصنف. هذا الأسلوب يعمل على تقليل بشكل كبير من عدد من القواعد التي وصفها النموذج المقترح إذا ما قورنت مع طرق المعتمدة على التصنيف المترابط الحالية. وأخيرا، (3) قمنا بتحسين دقة المصنفات الناتجة من خلال النظر في قواعد متعددة في خطوة تصنيف بدلا من خوارزميات التصنيف المترابط التي تستخدم قاعدة واحدة فقط لتعيين حالة اختبار قيمة. أشارت النتائج التجريبية الحقيقية على مجموعة البيانات النصية العالم تسمى رويتر أن النموذج المقترح يتفوق على تقنيات التصنيف الاخرى سواء التقليدية او المعتمدة على التصنيف المترابط.

## List of Tables

<b>Table</b>	<b>Page</b>
<b>Table 4.2</b> The contingency table for a set of binary decisions .....	40
<b>Table ( 3.1 )</b> vertical representation of data .....	46
<b>Table (3.3) :</b> Horizontal Representaion of data .....	47
Table (3.3) Candidate rules .....	54
Table (3.4) Part of a training data .....	54
<b>Table 4.1</b> Number of documents per category (Reuters-21578).....	58
<b>Table 4.2</b> The contingency table for a set of binary decisions .....	61
<b>Table 4.3</b> Precision results for each class in the Reuter and by each algorithm.....	65
<b>Table 4.4</b> Recall results for each class in the Reuter and by each algorithm.....	65
<b>Table 4.5</b> F1 results for each class in the Reuter and by each algorithm.....	67
<b>Table 4.6</b> Rules number for the proposed and MCAR algorithms.....	67

## List of Figures

Figure	Page
<b>Figure 1.1</b> : <i>The KDD Main phases</i> .....	2
<b>Figure 1.2</b> Text Mining Functional Architecture .....	3
<b>Figure 2.1</b> Text Mining Process .....	15
<b>Figure 2.2</b> k-Nearest Neighbor .....	19
<b>Figure 2.3</b> Example of Decision Tree.....	21
<b>Figure 2.4</b> Neural Network .....	25
<b>Figure 2.5</b> CBA algorithm.....	29
<b>Figure 2.6</b> CBA itemset discovery example .....	30
<b>Figure 2-7</b> Model Construction .....	35
<b>Figure 2.8</b> Using the Model in Prediction .....	35
<b>Figure 3.1</b> Main phases of the proposed model .....	41
<b>Figure 3.2</b> Pre-processing operation in text mining .....	42
<b>Figure 3.3</b> Porter stemmer operations .....	44
<b>Figure 3.4</b> Pseudo-code for rule discovery .....	48
<b>Figure (3-5)</b> dElcat algorithm illustration .....	49
<b>Figure 3.6</b> Rule ranking process .....	50
<b>Figure 3.7</b> The proposed rule pruning method .....	51
<b>Figure 3.8</b> The proposed prediction method .....	54
<b>Figure (4.1):</b> Weka GUI Main interface .....	59
<b>Figure (4.2):</b> Weka Explorer Interface .....	60
<b>Figure 4.3</b> Average precision of the contrasted algorithm .....	63
<b>Figure 4.2</b> Average recall of the contrasted algorithm .....	64
<b>Figure 4.3</b> F1 average for the contrasted algorithms .....	66

## Table of Contents

Acknowledgments.....	II
Dedication .....	III
Abstract .....	IV
الملخص .....	V
Table of Contents .....	VI
List of figures .....	VII
List of Tables: .....	VIII
List of Abbreviations .....	IX
<b>Chapter One Introduction .....</b>	<b>1</b>
1.1 Overview .....	1
1.2 Research Motivation .....	4
1.3 Problem statement .....	5
1.6 Thesis Contributions .....	6
1.5 Thesis outline .....	10
<b>Chapter Two Literature Review &amp; Related Work .....</b>	<b>13</b>
2.1 Introduction .....	11
2.2 Text Mining vs Data Mining .....	12
2.3 Text categorization Problems .....	13
2.4 Text Mining phases .....	15
2.5 Text preprocessing phase .....	17
2.6 Categorization techniques (Learning phase).....	19
2.7 Pruning phase .....	33
2.8 Prediction phase .....	35
2.9 Applications of document categorization.....	37
2.10 Chapter Summary .....	38
<b>Chapter Three Methodology .....</b>	<b>39</b>
3.1 Introduction .....	39
3.2 The Proposed Algorithm .....	40

3.3 Chapter Summary .....	56
<b>Chapter Four Result Analysis .....</b>	<b>56</b>
4.1 Introduction .....	56
4.2 Data Used .....	57
4.3 Environment .....	58
4.4 Performance Evaluation Measures .....	61
4.5 Results Analysis .....	62
<b>Chapter Five Conclusion and Future work .....</b>	<b>68</b>
5.1 Conclusion .....	68
5.2 Future Work .....	70
<b>References .....</b>	<b>71</b>
<b>Appendices .....</b>	<b>78.</b>
MCAR Algorithm results .....	78
Our Algorithm Results .....	79

# CHAPTER ONE

## INTRODUCTION

### 1.1 Overview

The existence of computerized data collection tools and large memory capacities simplifies the process of gathering and storing large quantities of information .For example, sales numbers of transactions within one month for a large grocery store is numerous ,also the amount of digital data on the internet is also large. This massive growth of stored databases give the opportunity for new intelligent data techniques, that can extract useful information from these databases. The process of producing this useful knowledge is achieved using data mining techniques (Fayyad, et al. 1996) .

Data mining (DM) is an interesting research field which is derived from many researches fields , primarily statistics, database, and Artificial Intelligence (Elmasri and Navathe, 1999). (Witten and Frank, 2000) defined DM as an automatic process of extracting unobserved patterns in data to help decision makers in decisions making processes. The input data in data mining system is called training data set, while the output is knowledge which is represented in different forms. There are different tasks in data mining like clustering, classification, association rule discovery (Witten and Frank, 2000). So the input and the aim define the appropriate task.

DM is one of main phases in the knowledge discovery in database (KDD) process as shown in Figure (1-1), the aim of KDD is discovering the unobserved useful information. The KDD is defined by (Fayyad, et a., 1996) as ‘overall process of discovering useful knowledge from data.’ In addition to data mining, KDD also involves the steps of data selection, data pre-processing, transformation, data mining, and evaluation (Figure 1.1).

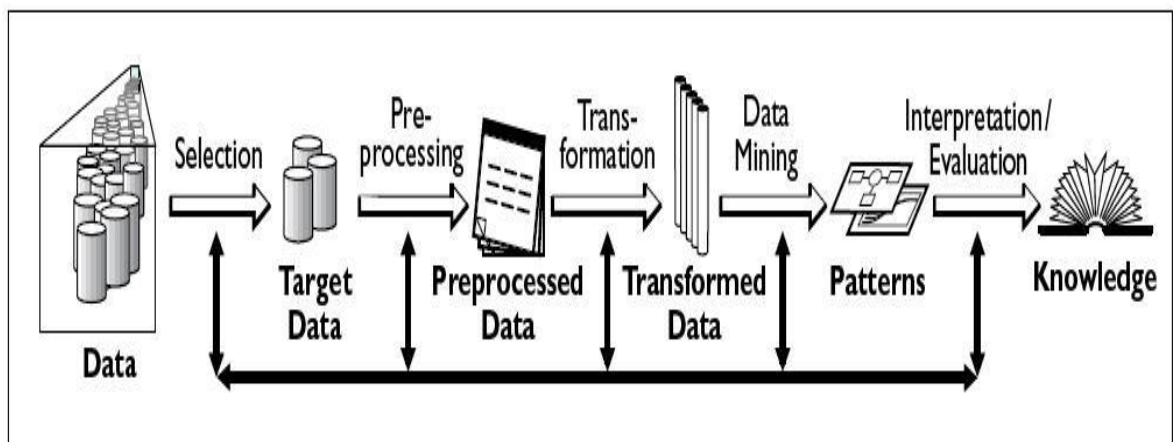
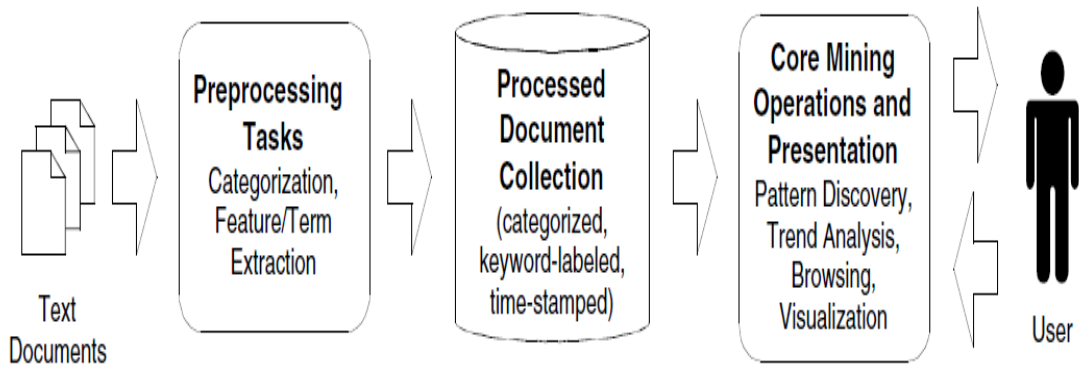


Figure (1.1): The KDD Main phases (Fayyad, et al. 1996)

Text mining (TM) is a process which domain experts can explore a number of documents in text collection utilizing automated data analysis techniques (Yoon and Lee, 2008). Unlike DM, TM looks for interested knowledge within unstructured textual data collection that is unobserved by users, and producing the knowledge through the recognizing and exploration of different interesting patterns (Yoon and Lee, 2008). Figure (1.2) the general phases of TM.



*Figure(1. 2): Text Mining Functional Architecture(Yoon and Lee, 2008)*

(Yoon and Lee, 2008) defined TM as “the process of extracting interesting patterns or knowledge from unstructured text documents”. It can be considered as a branch of DM. The main aim of TM is analyzing and classifying a number of unstructured textual data and to discover the knowledge .

Many TM methods from DM and machine learning exist like: decision trees (Quinlan, 1998), and Neural Network (Ruiz and Srinivasan, 2002). TM techniques concentrated on processing documents, learning a classifier (model), and then applying the classifier to classify un labeled text collection. However, few of these have focused on producing simple knowledge in the form of “If-Then” to end user that enable them to interpret and with maintain on classification accuracy. This can be achieved by applying the associative classification (AC) mining that was proposed by (Liu, et al., 1998 ) to construct accurate classifiers (sets of rules) by and later attracted many researchers, e.g. (Liu, et al., 1998; Thabtah, et al., 2006; Yoon and Lee, 2008; Niu, et al., 2009) .



Empirical studies (Kundu,et al., 2008;Yoon and Yoon, 2008; Niu et al, 2009) indicated that AC constructs classification systems which is more accurate than traditional classification techniques. In addition , AC produces rules that are understandable by end-user unlike traditional classification techniques like Naïve Bayes and Neural Network which construct classification models that are hard to interpret by end -user.

In this research, we develop an AC algorithm applicable to the classic English classification. In other words, the ultimate aim of this research is to evaluate the AC on large and unstructured text data collections.

## **1.2 Research Motivation**

English is an important and wide spread language in the world. Due to a vast amount of online English documents in most of the application domains such as finance, marketing, legal, etc, that include useful knowledge for decision makers. There has been great need for new research studies that can determine the appropriate techniques that are able to extract useful information from the English data collections. This study concentrates on learning knowledge important for planning and development from the field of documents classification.

The primary driving motivation of the research is to evaluate the effectiveness and applicability using AC mining on the complex problem of text categorization. Then, a comprehensive experimental study on the different learning algorithms for text classification is conducted with reference to predictive accuracy to identify the most appropriate approach.

### 1.3 Problem statement

In text classification problem, there is a collection of textual documents called  $D$  that has a predetermined number of categories (Yon and Lee, 2008). For Example articles published by a newspapers. So the input data collections are training data set and each document assigned to a definite category, thus a document in the training data set can be given as:

**(d, c) = (Czech Republic have defeated Netherlands in 2004 euro football champions, "Sports")**

where the class category is sport. Also there is a more sophisticated approach in TC called multi-labeled classification. This approach assumes that a document is categorized to more than one class. For example, a document in training data set as:

**(d, c) = (The US and China are the two largest consumers of energy in the world, "Mr. Obama said, "politics", "economy")**

So this document can be assigned for two classes "politics" and "economy" at the same time. However, in this project we will focus on single-label classification.

The main phase in TC is constructing a classification model (classifier) that assigns the text documents to their correct categories. The classification model is constructed by learning the content of the input training data set. Then the constructed model is utilized to predict the class for new documents. This is called supervised learning since the training data set involves labeled categories and the prediction is assigned only to these categories (Lan, et al., 2009). There are two main fundamental aspects during the learning process according to (Yon and Lee, 2008):

1- The classes are represented as symbolic labels. So there is no additional knowledge available to be utilized in the learning process to build the classification model.

2- The attributes of documents in learning depend on the content of that document, instead of contained metadata ,so the conception of relevance of each document to a specific category is inherently subjective.

## **1.4 Thesis Contributions**

### **Training Phase Improvement**

Textual data is often highly large in dimensionality and correlated, and thus the numbers of candidate keywords is relatively massive. So, mining high dimensional and correlated textual documents using AC is expected to be expensive. It is important to use an efficient learning algorithm for discovering of frequent keywords in the training phase. Currently utilized AC algorithms employs Apriori (Agrawal and Srikant, 1994) candidate generation function to derive frequent keywords. In Apriori, finding frequent keywords from textual data is handled in a level wise search, where keywords found to be frequent in iteration  $K$  are utilized to generate candidate keywords, at iteration  $K+1$ . In a specific iteration, an input data scan is vital to count the support for the candidate keywords. This requires high computation time and memory.

AC algorithms based CBA (Liu , et al., 1998) algorithm use the candidate generation function from Apriori to discover frequent keywords and due to the repetitive data scans, they suffer from high I/O costs. As a result, some AC techniques such as Negative-Rules (Antonie and Zaïane, 2004) employ a more efficient method than Apriori candidate generation step, called FP-growth (Han, et al., 2000) in order to

cut down the number of passes over the training data. In this thesis, we develop an efficient frequent keywords method, which decreases the number of database scans to one in the learning phase. For fast discovery of frequent keywords .

Unlike most AC techniques that use Apriori multi-scan approach to learn the rules, we use the vertical mining approach where each keywords has a list (transactions that contain that keywords in the textual data). We then use an efficient list intersection technique that requires only one data scan. We store frequent keywords and their locations (transaction Ids) during the scan. Then, by intersecting the list of the frequent keywords of size 1 we can easily obtain candidate *keywords* of size 2, and so forth. A detailed description of the learning procedure is demonstrated in Chapter 3. Experimental tests explained later on revealed that algorithms that utilize list intersection are more effective and better than CBA based ones.

### **Rule Filtering**

Since AC adopts association rule mining to discover frequent keywords and generate the rules. It considers all correlation between keywords in a textual data since the examples in textual data are often highly correlated, thus, the expected number of rules is enormous ranged into thousands, or even hundreds of thousands especially when support threshold is set to a very low value. Many of the produced rules are redundant, misleading or conflicting with other rules. Not all of the rules derived during the learning phase can be used in the prediction step. Hence, triggering pruning procedures including pre-pruning and post-pruning become essential to evaluate the generated rules and filter out such uninteresting rules. In this thesis, we introduce a new rule pruning in AC to do so. This resulted in moderate size classification models.

The proposed pruning methods are discussed in details in Chapter 3.

### **Prediction Phase**

The ultimate goal of any classification system is to build a model (Classifier) from labeled training data, in order to classify unlabeled data objects known as (testing data). Predicting the class labels of test cases in AC can be one of two types, either by predicting the class labels by the highest precedence single rule applicable to the test case (Single Accurate Rule Prediction) or prediction class labels by multiple rules (Group of Rules Prediction). The main problem associated with single rule prediction is favoring only one rule to perform the prediction even if there are many rules that are applicable to the test data to cover it. In this thesis, we introduce a new prediction method based on group of rules to overcome the above mentioned problem. The proposed class assignment method is discussed in details in chapter 3.

### **New AC algorithm for TC**

Recent experimental studies (liu et al, 1998)(Thabtah et al, 2004)(Yin & Han ,2003) in data mining revealed that AC builds more accurate classification models with reference to accuracy than traditional classification approaches. However, these algorithms suffer from few defects such as the generation of large number of rules which makes it hard for end-user to maintain and understand the classification models. In addition, most of them employ one rule strategy for test data forecasting. Lastly, these algorithms adopt the Apriori candidate generation approach from association rule to lean frequent keywords and generate rules. This normally necessitates high training time and memory use. To tear these problem in TC context, we present new AC that significantly reduces the number of generated rules by considering partly matching between the rule and the training example while building the classifier. This enables the

rule to have higher coverage and therefore ending up with moderate number of rules. Further, new prediction method that uses a group of rule in predicting a test case is employed. For fast learning of rules we use vertical mining based list intersection to reduce the data scans to one. Chapter 3 discusses the proposed, model in details and shows the experimental results against a number of datasets from the Reuter textual data set.

### **Adapting AC to TC**

In recent years, TC problem has attracted many researchers due to the availability of dense amount of documents online, digital libraries and digital journals. TC involves assigning text documents in a test data collection to one or more pre-defined categories based on their content (Sebastiani, 2002). Manual handling for TC is time and effort consuming whereas automated TC models indeed makes the classification process fast and more efficient. Despite the exponential growth of text documents, there are few AC research works on TC problem such as (Tang and Liao, 2007),( Yin & Han ,2003). Most of the current research works on TC problem are using traditional machine learning approaches including SVM (Vapnik ,1995), decision tree (Quinlan ,1999), and KNN (Tam et al ,2002) . To the best of our knowledge, there are few attempts to tackle the problem of TC using AC when contrasting to AC itself these are including (El-halees, 2006) (Qian et al., 2005) (Chen et al., 2005). In this thesis, the developed AC model has been adapted to TC problem by treating three main steps (Training, rule filtering and prediction).

## **1.5 Thesis Outline**

The remainder of this thesis is organized as follows .In chapter 2 ,we review the most important literature related to text categorization ,In chapter 3 we explain our proposed algorithm that depends on associative classification approach ,In chapter 4 we present the experiments with their used settings and results , Finally ,the conclusions and future work are presented in chapter 5 ,Our thesis includes also references and appendices .

## Chapter Two

### Literature Review & Related Work

#### 2.1 Introduction

Recently there is an urgent need to deal with a huge textual data that became easier to gain and obtain with existence of massive storage devices and high speed internet , Textual data can be more useful when extracting and employing them for important tasks such as classification and search engine that will enable users to make use of large text of data to get important information they need , Thus fields that deal with textual data attracted the researchers to study and develop methods to use in textual data like(Sebastiani,2002) and (Joachims, 1998).

Text mining (TM) is relatively a new research area that is utilized to extract useful information from large text of data ,which can be used in many tasks such as classification and visualization ,TM which is knowledge discovery from textual databases (Feldman & Dagan,1995), indicates to the operation of extracting knowledge from unstructured textual data . It is considered as an special field of data mining or knowledge discovery from stored data in structured databases (Fayyad, et a 1996).



Text mining is an interdisciplinary field which depends on

#### Information retrieval

"a field that combines information science and computer science, which is employed for indexing and retrieval of information from various textual information resources"( Hersh , 2003).

#### 1- Data mining

"process of extraction of implicit, unapparent and useful information from data in database".(Piatetsky-shapiro & Frawley ,1991)

#### 2- Machine learning

A scientific field that interests in design and development of algorithms that permits computers for improving behaviors utilizing empirical data.( Wernick et al, 2010) .

4- Statistics : mathematic science concerned with collection, analysis,interpretation and presentation of data .( Moses &Lincoln 1986)

5- Computational linguistics : is a field concerned with statistical or modeling of natural language from a computational perspective.

## 2.2 Text Mining vs Data Mining

TM is a process of analysing textual data to extract important information which utilised for many purposes. This definition infers that TM is similar to DM with few differences that are discussed by (Lan, et al, 2009) (Kroeze et. al, 2007):

1- TM concerns in extracting knowledge from only textual data while in DM we can extract from any data types such as numeric, images .

2- In TM ,the data are unstructured with large dimensionality while the data used in the DM are often processed ,and structured .

3- The knowledge in textual data is observed but the process of extracting consumes a lot of time. while knowledge in the data bases used in the data mining is usually non clear that cannot be extracted simply .

TC is the process of classification a set of documents into categories from a predefined set automatically ( Sebastiani,2002). For example if we have a set of documents for each category, then we need to extract knowledge from these documents to build a classification model that applied to determine the category for each new document.

### **2.3 Text categorization Problems**

As mentioned documents classification concerns about extracting main features document by mapping the document into a pre-determined group of categories. This process tries to find a model which describes and distinguishes class-labeled documents by analyzing a group of training textual data .so the model will be used to predict the category of the document with the right class category. In other words the ultimate goal TC is to develop a classifier for predicting class for a new data .This process is therefore predictive based on supervised learning(Abdel Hady et al., 2010) .On the other hand ,there are several problems which we may face when textual documents are used in the mining process like the unstructured form of the database. The main problems associated with TC are :

### 1-Large textual database

In the era of digital revolution ,textual data are simply stored in computers and electronic devices .

### 2- High dimensionality

In textual data each keyword has a separate dimension.( Ghahreman &Dastjerdi ,2011). Meaning the data has a complex interpretation property, This problem is often solved by data representation techniques such as dEclat algorithm( Zaki & Gouda,2003) and Aprioi (Agrawal &Srikant,1994) that will be illustrated later .

### 3- Unstructured nature

The textual data is actually in unstructured format that is not interpretable by computers meaning that there is no specific structure for textual data , So it is noticed that 80% of textual data is unstructured format (Khan et al ,2010), this problem will be solved by data representation techniques that will be illustrated later in the thesis .

### 4- Noisy Data

The textual databases are noisy so that they involves a lot of structural and spelling mistakes. This problem is solved by preprocessing techniques as seen in section (2.5) .

## **2.4 Text Mining phases**

Textual data are unstructured as mentioned earlier ,to prepare textual data and represent it in efficient manner ,we make use of it for the main purpose prediction later ,data goes through many phases, see figure(2.1). It is an reiterated process meaning that

the output of one phase is the input of next the phase. The phases that form a text mining process may contain according to (Shahbaz et al ,2011 ) :

#### 1-Text preprocessing

Text preprocessing is the first step of text mining process in which the document collection is prepared by analyzing it syntactically or semantically(Khan et al ,2010). In this phase the structure of the document is analyzed and organized .

#### 2- Text transformation

Text transformation focuses on the feature representation for the textual data analysis (Khan et al ,2010).The text document is treated as bag of words(BOW) since the words are employed represent to the textual data .The techniques applied in this phase are stop word removal and stemming .

#### 3- Feature selection

This phase employs techniques that select the most important words that represent the textual data .This will enable the specialists to reduce the large dimensions to smaller ones number. Examples of technique that are utilized feature selection process are information gain (Lee & Lee2006).Chi square(Yang & Pederson, 1997),in this process the selected words are used to represent the textual data collection, other unimportant words are eliminated .

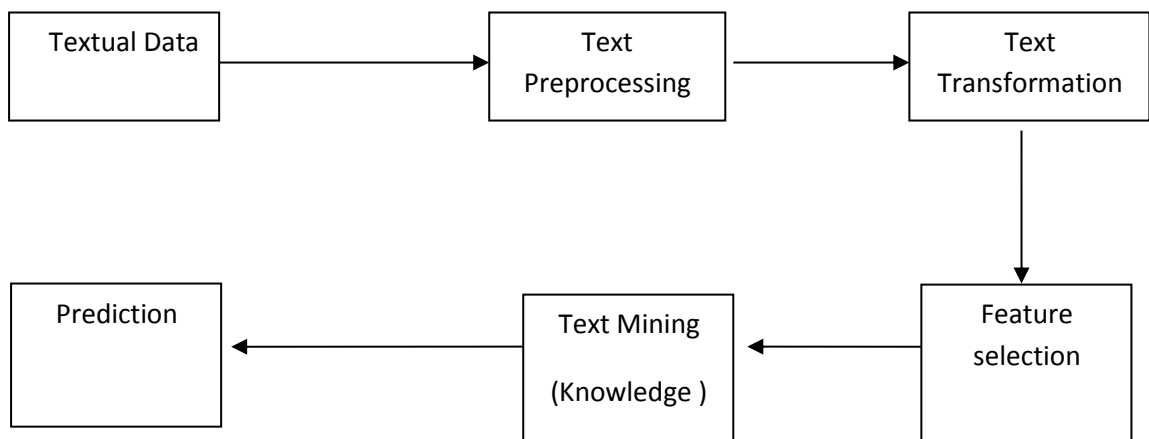
#### 4- Data mining

In this phase DM techniques are applied on textual data to discover knowledge patterns like(rules, tables).examples of these techniques are neural network support(Ruiz and Srinivasan, 2002). support vector machine(SVM) (Vapnik ,1995) and

associative classification(AC) (Thabtah, et al., 2010). that will be illustrated later , The interesting correlations are extracted from the document and represent them as valuable knowledge.

## 5- Result Evaluation

In the final stage the discovered knowledge is evaluated by testing the model accuracy against the test data . The knowledge is employed to predict new cases of textual data .



**Figure (2.1) : Text Mining Process**

## 2.5 Text preprocessing phase :

Textual data is often unstructured form which is considered as a problem when applying TM techniques due to existing of unimportant stopwords like( 'and' , 'or,' the') or being formatted in an inappropriate form. the advantage of preprocessing is applying techniques to clean and structuring the textual data for analysis in following stages, which is an important phase in practical TM studies, The aim of preprocessing is representing each document as a

feature vector, meaning that separating text into individual words. Here we explain different preprocessing text operations .

### 1- Stop Word Removal

Many of repeated words in English are useless in TM field . These words are 'Stop words' (Khan et al ,2010).Stop-words are frequent words that don't have important information( pronouns, prepositions, conjunctions). Examples of these words such as 'the', 'of', 'the ', 'if'. The aim of this step is removing these Stop words ,the most popular technique is SMART stop word list (Salton,1989).

### 2- Stemming

Stemming algorithms are utilized to discover the root of any word( Allan& Kumaran ,2003) .This step is associated with linguistic science .For instance, the words: responsible , responsibility can be stemmed to the word 'Response '.The most popular technique is the Porter Stemmer algorithm (Porter1980) .

### 3-Document Indexing

The aim of document indexing is improving the efficiency by discovering important terms from textual data that will be used for indexing the document(Khan et al ,2010).

Document indexing involves selecting the appropriate keywords from the all training textual data ,and specifying weights for these keywords to each specific document, then converting each document into a table of keyword weights.

#### 4-TermWeighting

Term weighting specifies if the classification system is successful or not .Because each terms has its importance in a document , the term weight is considered as an important indicator for each term(Wang,& Chiang ,2007).

The three main factors which impact on the importance of a term in a textual data are the Term Frequency (TF) factor, Inverse Document Frequency (IDF) factor and Document length normalization( Karbasi, & Boughanem,2006) ,TF of each word in a document is a weight that associates with the distribution of each word in documents, which indicate the weight of each word in a document. IDF of a word in the document is a weight that associate with the distribution of the word in that document .so it indicates the importance of a word in a document (Diao & Diao, 2000).

#### 3- Dimensionality Reduction

Document frequency (DF) is the number of documents involve a term(kwok,1998) . DF is the simplest process for dimensionality reduction. Stop word removing which is illustrated earlier, eliminates all words that are unimportant to the classification process , while DF remove unrepeated words. So all words that appear in less than "N" documents of all text documents are not treated as features, where "N" is a pre-determined .

#### 7- Tokenization

A document is considered as a string, and then each string is divided into a group of tokens(Khan et al,2010).Tokenization is a process which includes dividing the sequence characters into more meaningful Tokens. In text categorization context, the text document gets

broken into sentences, and words (barcala et al,2002) .We use WEKA (Weka, 2002) filtering in the Explorer GUI to tokenize the input database .

## **2.6 Categorization techniques (Learning phase)**

After preprocessing phase ,the important words are represented in tables ,These tables will be used in learning phase, from this phase we can construct a model by these tables that are used as training data sets ,after constructing the model by one of those techniques we illustrate below, this model will be used to predict values of new data in prediction phase, the most powerful techniques that used in text categorization are :

### 1- K-nearest neighbor (k-NN)

The k-nearest neighbor algorithm (k-NN) (Tam et al ,2002) is applied to evaluate the degree of resemblance between documents and (K) training data and storing particular amount of classification documents ,so that specifying the class of documents that will be tested .This technique is an instant-based learning algorithm that categorize document based on nearest feature area in the training textual data set (Eui-Hong et al,1999). The feature area is divided into areas based on the class of the training data set. A point in the feature area is allocated to a particular class when it holds the most repeated category in the k nearest training data. The important point of this technique is the ability to determine neighbors of a specified document by using similarity degree(Eui-Hfiong et al).The training phase involves storing the feature vectors with categories of the training textual data .While in the classification phase, spaces from the new vector, form the input document, so all stored vectors are calculated and k samples are chosen. The supposed category of a document is forecasted depending on the closet point that already allocated to a specified class.



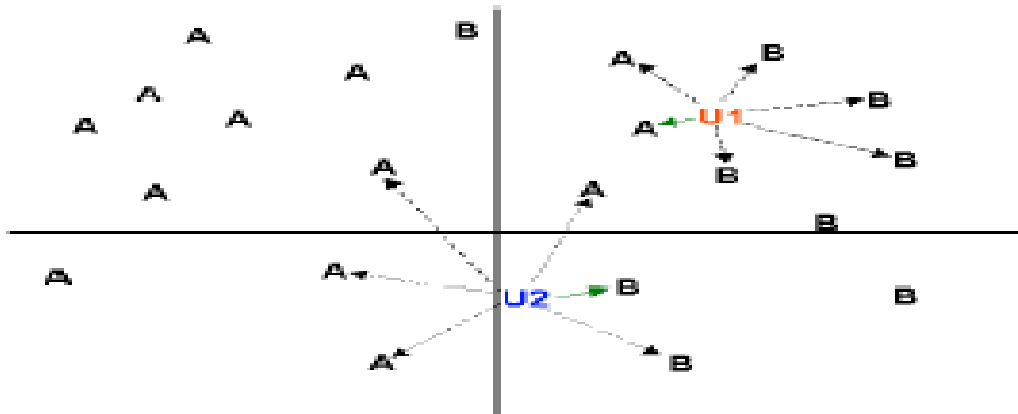


Figure (2.2) : k-Nearest Neighbor (Khan et al ,2010 )

Advantage(s) :

- It is simple and wide applied technique for text classification .

Drawback(s) :

- 1- The accuracy of this approach is decreased by the existence of noisy or unrelated features .
- 2-The main disadvantage of this approach is utilizing all features in distance calculation, so this technique requires high computational resources when the volume of training textual dataset increases particularly.

## 2-Decision Tree

The overall process of decision tree is reconstructing the tables of training textual data sets in the form of tree built by (true/false)queries(Quinlan,1999).In decision tree ,leaves denote the identical class of documents and branches denote correlations of features that infer to those categories. The well formed decision tree will simplify the

classification of any document by tracing this tree from the root node through the query structure to a specific leaf, which ensures the aim for the documents categorization .

Advantage(s) :

The main benefit of decision tree is the simplicity and understandability even for non-expert users. In addition ,the interpretation of a given outcome can be easily formulated by applying simple mathematics algorithms, which support the aim of the classification logic, that is considered as valuable information of classification .

Drawback(s) :

The major drawback of implementing decision tree is over fitting the training textual data with the existence of another tree that does not categorize the training data accurately, so the classified documents may appear to be classified better(Quinlan,1999). Since the decision tree depends on classification algorithm which is focusing on classifying training data efficiently, with ignoring the efficiency of classifying testing documents .

Dependent Variable: study

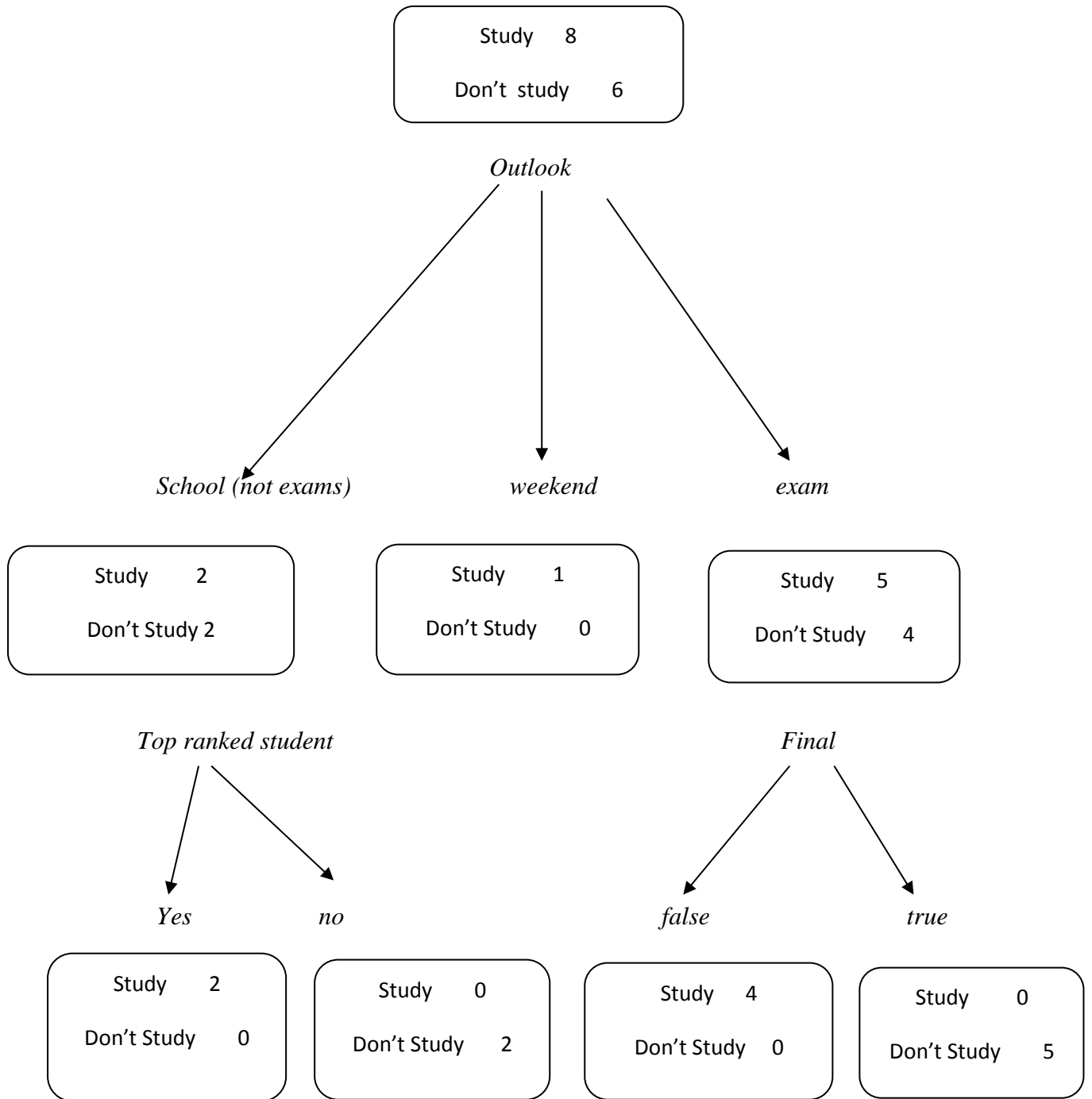


Figure (2.3) : Example of Decision Tree

## 2- Naïve Bayes Algorithm

Classification based on Naïve Bayes(NB) (Thabtah et al,2009) is a simple statistical classification system that based on employing Bayes' Theorem with strong independence hypothesis. it is considered as probability model that based on independent feature model. These independence hypotheses of features make the features order is unrelated ,so the existence of one feature does not impact on other features in classification processes(Heide et al ,2002). This make the calculation of Bayesian classification method more accurate ,but this hypothesis restricts its applicability.

The NB classifiers can be utilized effectively by using a relatively small size of training textual data to speculate the important variables for classification. Since independent variables are supposed ,so only the variances of each class variables is required and not the overall covariance matrix. Although it seems that this approach is simple especially in using hypothesis, the NB classifiers actually is more efficient in complex real-world situations than person may think .

Advantage (s) :

The main feature of the NB classifier is that users need small amounts of training data to speculate the variables required for classification.

-Drawback(s) :

The main drawback of NB classification approach is its relatively low categorization performance comparing with other discriminative algorithms like SVM, NB is efficient when applying it on numeric and textual data, it can be implemented and computed easily comparing with other approaches, but conditional independence

hypothesis is destructed by real-world data and the performance is bad when features are highly related and neglecting frequency of word occurrences.

When the NB classifier is applied on the TC problem :

$$P(\text{Class/ Document}) = P(\text{Class}).P(\text{Document/Class})/P(\text{Document}) \dots\dots\dots(2.1)$$

Where :

$P(\text{class}|\text{document})$ : The possibility that a document D relates to a class C.

$P(\text{document})$ : The possibility of a document, noticing that  $p(\text{document})$  is a constant divider to every computation .

$P(\text{class})$ : The possibility of a class, which is computed from the number of documents that relate to a specific class divided by number of documents in all class .

$P(\text{document}|\text{class})$  denotes the possibility of document given class, and documents can be represented as group of words, so the  $p(\text{document}|\text{class})$  can be written like:

$$p(\text{document}|\text{class}) = \prod_i p(\text{word}_i|\text{class}) \dots\dots\dots(2.2)$$

So :

$$p(\text{class}|\text{document}) = p(\text{class}) \prod_i p(\text{word}_i|\text{class}) \dots\dots\dots(2.3)$$

Where :

$P(\text{word}_i|\text{class})$  : The possibility that the i-th word of a given document occurs in the document from class C, and this can be calculated :

$$P(\text{word}_i|\text{class}) = (T_{ct} + \lambda) / (N_c + \lambda V) \dots\dots\dots(2.4)$$

Where

Tct: The number of word occurrences in that category .

V: The size of the words table .

$\lambda$ : A positive constant .

NB was successfully applied to TC problem and showed high accurate values comparing with other TC techniques (Thabtah et al., 2009).

### 3- Artificial Neural Network

Artificial neural networks (ANN) are built from a large number of nodes with an ordered entries volumes greater than computational entries of traditional architectures (Ruiz and Srinivasan, 2002). These nodes are artificial neurons which connected into group by applying a mathematical model for information processing depends on a technique used for calculation. The neural networks have their neuron that used to store data. Different kinds of ANN techniques were implemented to document categorization tasks. Some of the researches use the single-layer network, which consists of two layers only : an input layer and an output layer which is considered simple for implementation (Ruiz and Srinivasan, 2002). Inputs are connected directly to the outputs through a series of weights, also the multi-layer network, that contains an input layer, one or more hidden layers, and an output layer, is widely applied for classification tasks (Ruiz and Srinivasan, 2002)

- Advantage (s) :

The main advantage of applying (ANN) in categorization tasks is the ability to process documents with high-dimensional features, also documents with noisy data (Ruiz and Srinivasan, 2002).

- Drawback(s) :

1- The main disadvantage of the ANN is that they require high computing resources and high CPU and memory .

2- ANN are so complicated to interpret for non-expert users, so applying ANN for document classification produce high accurate results ,but its more suitable for complex fields such as numerical discrete and continuous data .

Input

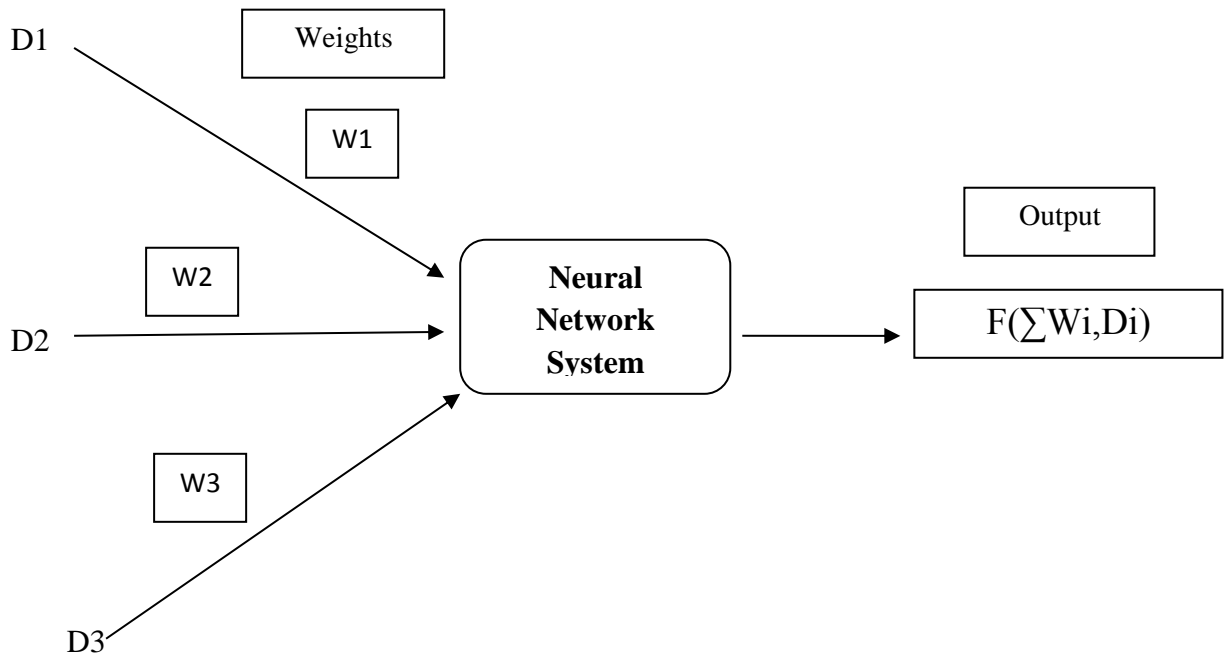


Figure (2-4) : Neural Network

#### 4- Support Vector Machine (SVM)

Support vector machines (SVM) are one of the popular categorization techniques that are more accurate than others. The SVM classification approach depends on the structural risk minimization principle from computational learning theory (Vapnik ,1995). The main idea for this principle is finding assumptions that ensures the lowest error rate. In addition , The SVM requires positive and negative values for data training set rarely used in other categorization methods. These positive and negative values for training data set are used in SVM to help specialists in separating the positive from the

negative data in the (N) dimensional space, The document features that are closest to the decision surface are the support vectors. The performance of the SVM categorization will not be affected if documents that do not relate to the removed support vectors from the set of training textual data (Heide et al ,2002 ) .

- Advantage(s) :

- 1- Efficient statistical model for cases of large feature sets.
- 2- Applied for cases in which the discriminator surface of two classes is not linear( Ghahreman &Dastjerdi 2011)

- Drawback(s)

- 1- The main disadvantage of SVMs is their relatively complex training and classifying algorithms that consumes time and memory .
- 2- Errors emerged within the categorization tasks since that the documents may be assigned to several classes due to the similarity which is computed independently for each class (Heide et al ,2002 ) .

6- Associative Classification :

Classification using Association is called associative classification(AC) is a research field in DM that combines association rule discovery and classification. Actually , AC applied association rule discovery algorithms to discover the knowledge ,after that choosing a group of rules to construct a classifier (Thabtah et al,2010). The aim of AC is to build a model (classifier) that contains a number of rules(extracted knowledge)from labeled input which is training data set that used for classification prediction for a unclassified test data as accurate as possible(Baralis et al,2008).



Many research experimental studies (liu et al, 1998;Thabtah et al, 2004;Yin & Han ,2003) indicated that AC is a high efficient method that builds accurate and predictive classification systems better than traditional classification approaches such as decision tree (Quinlan, 1998) and The k-nearest neighbor algorithm(k-NN) (Tam et al ,2002) ,as well as the output of (AC) techniques are formed as simple if–then rules, that enables the user to understand and interpret the output easily(Thabtah,2007) .

Recently , several AC approaches have been developed such as CMAR(Li et al ,2001), MCAR (Thabtah, et al, 2005), CACA (Tang and Liao, 2007), SARC (Christopher,2011) and others, also These studies improve and explain the benefits of AC algorithms over traditional techniques with respect to accuracy and understandability on particular .

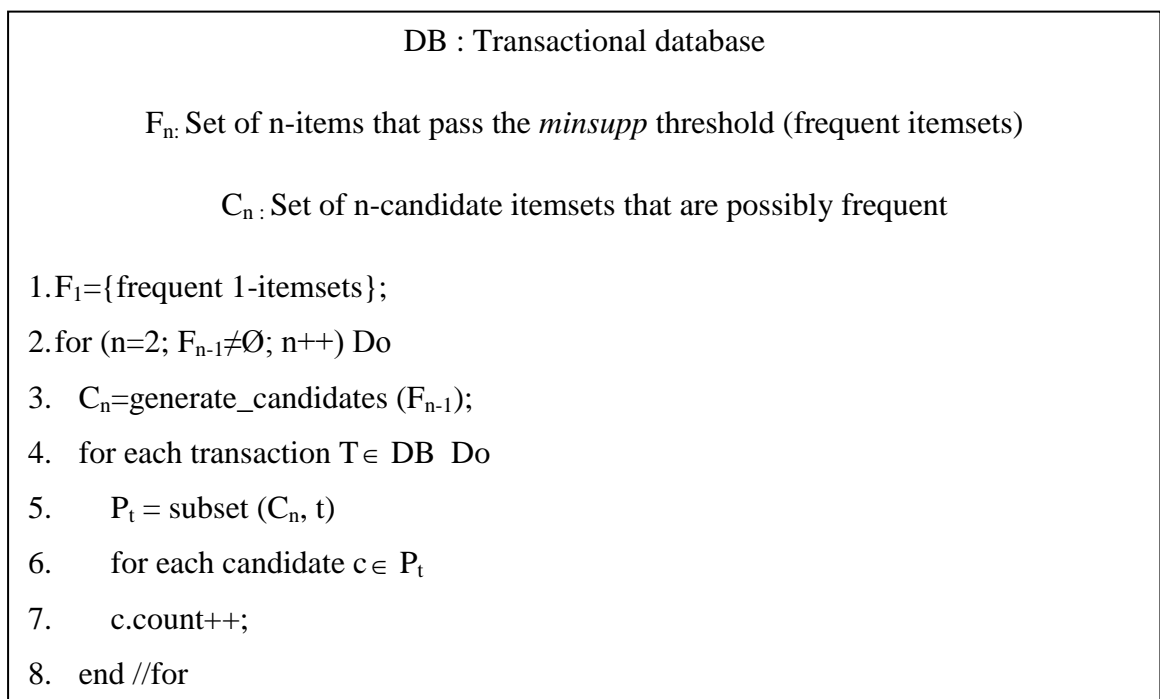
On the other hand the main challenges of AC are the exponential growth of rules, which means that they produce huge number of rules knowledge which make the constructed outsized classifiers that will hinder the usability because specialists will face difficulty in interpreting them, also the majority of AC algorithms exploit the search based CBA technique (Liu, et al., 2001) in producing the rules that normally needs high capacity of memory and processor resources, which will limit the usability of these techniques in practical applications (Li, et al., 2001) (Thabtah, et al., 2010),AC algorithms processes depend on two important definitions :Support and Confidence , where the support of the itemset is number of transactions in  $D$  that contain the itemset. An association rule is an expression  $X \Rightarrow Y$  , where  $X, Y \subseteq I$  are two sets of items and  $X \cap Y = \emptyset$ . While confidence is the probability a transaction contains  $Y$  given that it contains  $X$ , and is given as  $\text{support}(XY)/\text{support}(X)$

The most popular AC algorithms are :

### 1- Classification using Association (CBA) :

Classification using association (CBA) is proposed by (Liu, et al., 1998) ,this algorithm is developed based on the association rule mining algorithm Apriori (Agrawal and Srikant,1994) that used in order to discover rules (knowledge) . Frequent rule items are discovered in an efficient manner ,so that new candidate item sets are generated by scanning all the training data in each iteration from frequent item sets already discovered.

Figures (2.5) illustrates the rule discovery process of CBA ,candidate item sets ( $C_n$ ) are generated by merging frequent items ( $F_n$ ) with previous ones , item sets in  $C_n$  that are less than support threshold value are eliminated .the subset of candidate item sets involved in the actual database transactional ( $t$ ) are discovered by the subset function (line 5). Candidate item sets supports values are increased when these candidate itemsets are specified from  $C_n$ , (line 6-7). If there are no frequent itemsets  $F_n$  in the  $n^{th}$  iteration is discovered , the algorithm process ends



∈

**Figure (2.5) : CBA algorithm**

Figure (2-6) explains the frequent rule items discovery process in CBA, supposing that  $minsupp$  is 50% (count =2). CBA passes over the database to discover candidate item sets of length one, that exceeds the support threshold ( $F_1$ ). In the second iteration ,the algorithm produces candidate itemset of size two ( $C_2$ ) and passes over the database to specify frequent subset( $F_2$ ). When frequent item sets of length three ( $F_3$ ) are discovered The algorithm ends.

After discovering all frequent item sets ,the CBA algorithm assigns class labels and generates all class association rules(CARs) after computing the confidence value for each CAR. The rule item that passes the minconf value becomes a rule. Then rules are ranked according to confidence and support values and pruning that is used to delete misleading and (repeated ) rules.

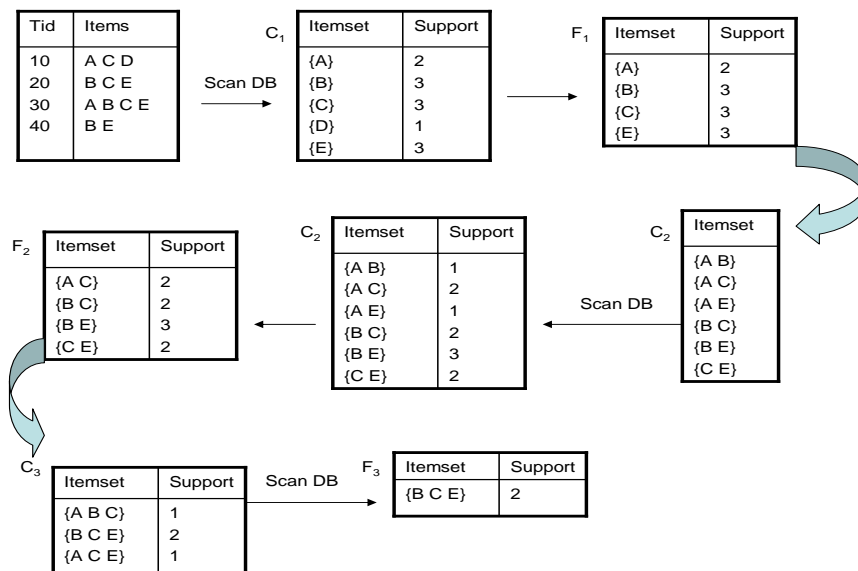


Figure (2-6) CBA itemset discovery example

## 2- Class Based Associative Classification Algorithm (CACA)

(Tang and Liao, 2007) developed a new algorithm based on AC approach that produces rules set and constructs the classifier in one phase. CACA justify its principle that AC algorithms involves many phases that produces repeated rules which consume time and I/O resources ,which is considered a challenge for AC algorithms ,so CACA constructs the classifier that reduces the searching space of the frequent items by producing concurrent rules that are stored in OR-tree.

## 3- Vertical Mining

This algorithm is proposed by (Zaki et al,1997), the idea is to minimize the number of input database scans, so that only one database scan is required , this algorithm which is called (Eclat) employs a vertical database transaction layout, where frequent item sets are produced by utilizing simple tid-lists intersections, complex data structures in this process is not required .

dEclat is a developed variation of Eclat which is proposed in (Zaki and Gouda, 2003). The dEclat algorithm employs a new vertical layout representation technique called a diffset, that stores the differences in the transactions identifiers (tids) of a candidate itemset from its produced frequent itemsets .Thus the stored tids requires less size of the memory by applying this algorithm. So the difference between the class and its member itemsets are stored , this process avoids storing the complete tids of each item set.

## 4- Multi Class Associative Classification (MCAR) Algorithm

MCAR algorithm(Thabtah, et al., 2005) includes of two main phases: rules generation and a classifier builder. In the first phase, the training dataset is scanned only

one time to find the rules of size one, and then MCAR intersects the rules tid-lists of size one to find potential rules of size two and so on . This rules discovery phase is not required for passing over the training data many times. In the second phase, produced rules are utilized to construct a classifier by evaluating the effectiveness of these rules on the training dataset. Rules that holds particular number of training objects will be stored in the final classifier.

#### 5- Co-training based Approach

This algorithm is proposed by (Wang, et al., 2011) ,it is an AC algorithm also called ADA(Approach for adapting associative classification)that produces rules from the input training data set and the classified resources like the training data set, actual classification rules, and test data .This means that the classifier is modified frequently until the classified resources matches a particular value . The researchers applied a co-training method (Mei, et al., 2006) to update the classifier by filtering the new discovered rules by the existing classification rules. ADA is a semi-incremental AC algorithm because it constructs AC classifier using few training cases or frequent patterns (keywords) rather than complete training cases. Then the classified cases and the classification rules are employed to modify the classifier by adding or removing rules.

### **2.7 Pruning phase :**

Decision trees produced by algorithms like ID3 and C4.5 are accurate and efficient, but actually they suffer from the drawback of large trees existence that make them ambiguous to specialists(Quinlan, 1999). To handle this problem, researchers have great interest in tree pruning. So tree pruning techniques are used to reduce the size of a large tree, so it becomes easier to interpret. Such methods typically applied statistical

criteria to eliminate the least important branches, the benefit of these technique is to accelerate classification process and produce more accurate model(classifier)the most popular pruning methods are:( Patil et al ,2010)

#### 1- Reduced Error Pruning

This technique was introduced by Quinlan(Quinlan,1999).It is a simple and interpretable pruning technique for decision tree ,so the technique tests the misclassification problems that occur on the test data set for every non-leaf sub tree of the original decision tree, The misclassification occurs if this sub tree is exchanged by the best possible leaf .If the error rate for the produced tree is equal to or smaller than original tree and this sub tree have sub tree with the same property, is by the leaf. Otherwise, kill the process. This operation will reduce error pruning in bottom-up induction( Esposito et al ,1997). Because each node is scanned one time to estimate the pruning probability for the tree ,the importance of this technique is its linear computational complexity( Esposito et al ,1997), so this method needs a test data set different from the training data that built the tree (Quinlan,1999) , pruning occurs if the training set is greater than the test set .

#### 2- Minimum Error Pruning .

The technique was proposed by( Niblett and Brotko,1991). It is a bottom-up technique that searches for single tree that reduces the estimated error rate on an independent textual data set. (Mingers .1989) indicates to the drawbacks of this technique: it is rarely practical for the similar classes, and producing only a single tree,

which is not practical for expert systems, where several trees are more efficient ,also the number of classes impacts the degree of pruning, that may lead to violated results .

### 3- Cost-Complexity Pruning .

The technique was introduced by( Breiman et al,1984), this technique applied the errors number and tree complexity for its calculations. The complexity of the tree is represented by the size of the tree. This technique needs a pruning set differs from the original training set. The main drawback of this technique is choosing only one tree in the set, rather than selecting set of all possible sub trees( Esposito et al ,1997).

## **2.8 Prediction phase :**

The ultimate goal of TM is prediction ,and predictive TM is the most common type of DM which is used in business applications ,the prediction phase is used after constructing a model, in prediction phase we use the constructed model to predict unknown or missing values ,within this phase we speculate the accuracy of the classifier by testing the known label of test data with the classified result from the classifier, so accuracy average is the percentage of test set that are correctly classified by the classifier, this kind of constructing is called supervised learning ,so the training data are assigned by labels referring to the class of the new data that is classified based on the training dataset ,while in unsupervised learning the class labels of training data is not defined , the figures (2.7) and (2.8)illustrate the whole process of constructing model and using it in prediction .

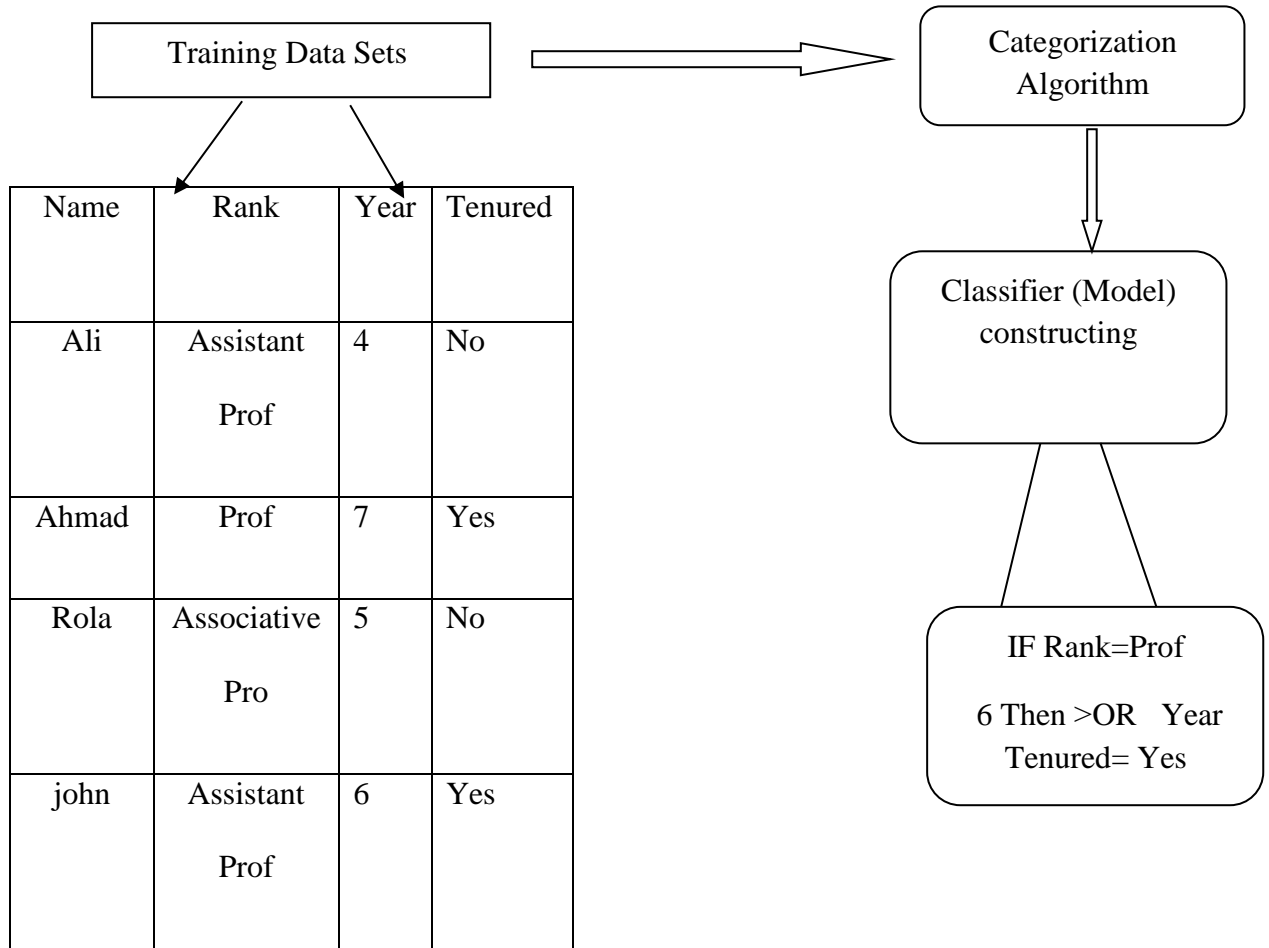


Figure (2.7) : Model Construction

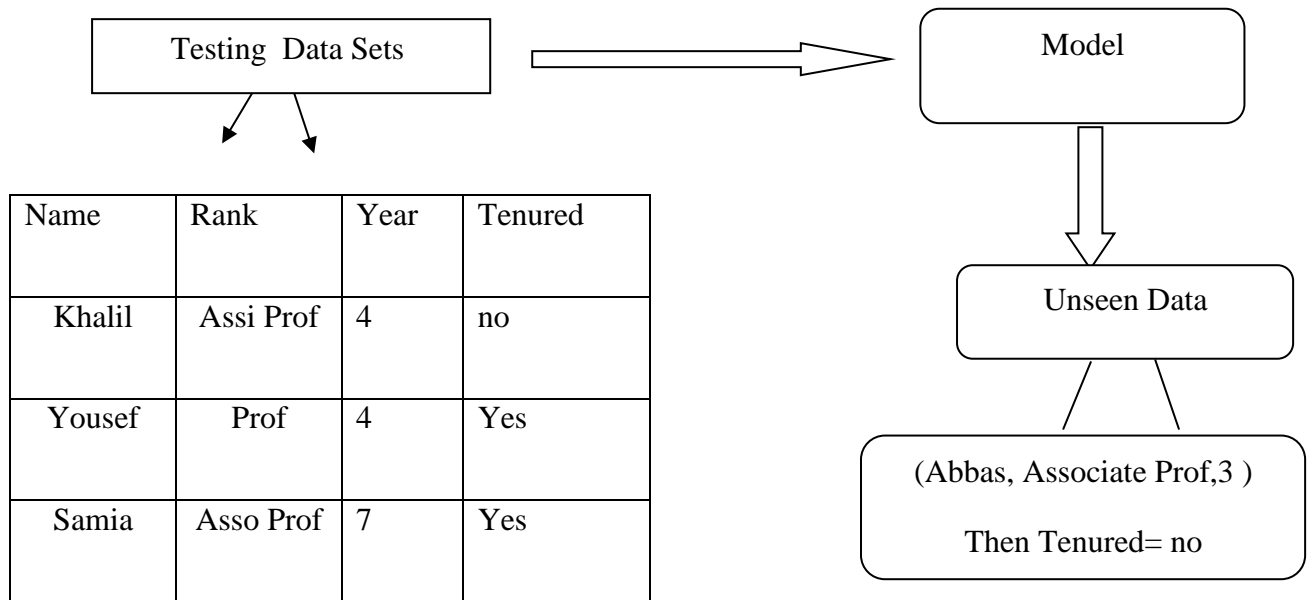


Figure (2.8) : Using the Model in Prediction



## **2.9 Applications of document categorization :**

### 1- Automatic indexing for Boolean information retrieval systems :

In these systems, every document is allocated to keywords that indicates its content, where these keywords refers to a finite group of words, called (controlled dictionary)and this operation is done by human indexers, which is an too cost operation .

### 2- Document organization :

Generally ,all issues related to document organization ,may be applied by automatic classification system. For example, incoming news should be classified based on categories used in the categorization system before publication, such as sport, politic, economic, this system is widely used in newspapers that contains high daily number of subjects, such system is used to select automatically the best category for a given subject.

### 3- Document filtering :

Document filtering indicates to the process of dynamic categorizing documents collection for incoming documents sent by an information producer to a consumer(Belkin & Croft ,1992). Such as news feed, the filtering system removes the documents which are not interested by customer. Filtering is viewed as a special case of classification with (non- interfered)categories, so incoming documents are classified in two categories, the related and unrelated .

## 2.10 Chapter Summary

In this chapter we introduced the TM field and the scientific fields that have been derived from it ,we differentiate between TM and DM fields ,and introduced TC .After that we summarized problems of TC with its solutions, particularly we demonstrated briefly phases of TC supported with graphic figures, the following step we illustrated preprocessing phase that is used to represent useful data effectively, and presented and explained deeply the most effective algorithms that used in learning phase to build classifier .In addition we explained pruning techniques that used to reduce the number of rules from the learning phase, we explained the prediction phase ,and illustrated applications used for TC . In the next chapter ,we will explain the proposed model as a powerful tool for building learning classifier for TC, we then explain main steps in building an our AC classifier .

## **Chapter Three**

### **METHODOLOGY**

#### **3.1 Introduction**

As stated earlier (chapter 2), AC is a promising classification approach which is rarely used in text categorization, also there is a lot of researches interested in this approach in an unstructured textual data. AC constructs more predictive and accurate classification system that traditional approaches such as (KNN) and decision tree.

However, this approach faces an important problem which is the production of huge number of rules which restrict the ability to interpret and control them for specialists and limits its use certain domain applications.

In this chapter, a new AC model is proposed that consternates on developing new pruning technique that reduces the number of rules produced during constructing the classifier phase by removing redundant and useless rules to produce appropriate size classifiers. Therefore, the end-user can deal with controllable numbers of rules which enables him in prediction. Moreover, the proposed model employs a prediction procedure that makes use of more than one rule in forecasting test cases limiting the use of a single rule as in other AC algorithms such as CBA (Liu, et al,1998) and MCAR (Thabtah et al, 2005). This model will be utilized against TC unstructured data collection in this chapter to achieve the ultimate goal of the thesis which is the proposing of TC model based on AC data mining. This intelligent model will

automatically predict the type of any text document based on their content without using any predefined knowledge except the training textual data set and (MinSupp and MinConf thresholds).

TC is considered as challenging problem in TM. This research area is complex due to the size of data used and its high dimensionality and the format of such data is often unstructured. This chapter seeks to answer to the following questions:

- 1- Will a new AC model reduces the exponential growth of rules when applied to TC problem?
- 2- Is AC approach improves accuracy of the problem classifying textual data into their appropriate categories?

The proposed algorithm is illustrated in Section (3.2) that consists of details about data preparation, rule discovery, rule pruning (classifier builder), rule sorting and class assignment of test cases. We also explain the differences between the proposed algorithm and other AC techniques and finally the chapter summary is given in section (3-3)

## **3.2 The Proposed Algorithm**

The new model consists of four main phases: the pre-processing phase, learning the rules, construction of the classifier, and prediction of new cases as shown in Figure (3-1). During the pre-processing phase – that is used in cases of unstructured text collection that requires removing unnecessary stopwords such as (is, was) - is

performed. This phase converts the unstructured text into structured one in order to enable us to represent text (document) as a feature vector. In other words, separating the text into individual words. During the rules learning phase, the input data get scanned to discover frequent items, in case that any keyword that appears in the input data set with a frequency less than the MinSupp threshold it will be neglected .

Once all frequent keywords are found, our algorithm compares their confidence values with the MinConf threshold. The algorithm chooses only the frequent keywords which have confidence values larger than the MinConf and converts them into rules. Any frequent keywords that does not satisfy MinConf is eliminated. Therefore the complete set of rules represented in the training data set and hold high confidence values are generated .The next step is to rank the rules according to certain measures and then choose a subset of the complete set of rules to build the classifier.

After producing the rules by our algorithm, they must be sorted according to certain criteria ,e.g support ,confident , etc, in descending manner (from highest to the lowest one). In case two or more rules have the same confident, they will be ranked according to support value. After ranking we filter the rules in order to choose the most accurate rules which have training data coverage. This has been achieved during constructing the classifier phase, and finally we test the classifier on test data to evaluate its effectiveness. We will illustrate these phases in the subsequent sections .

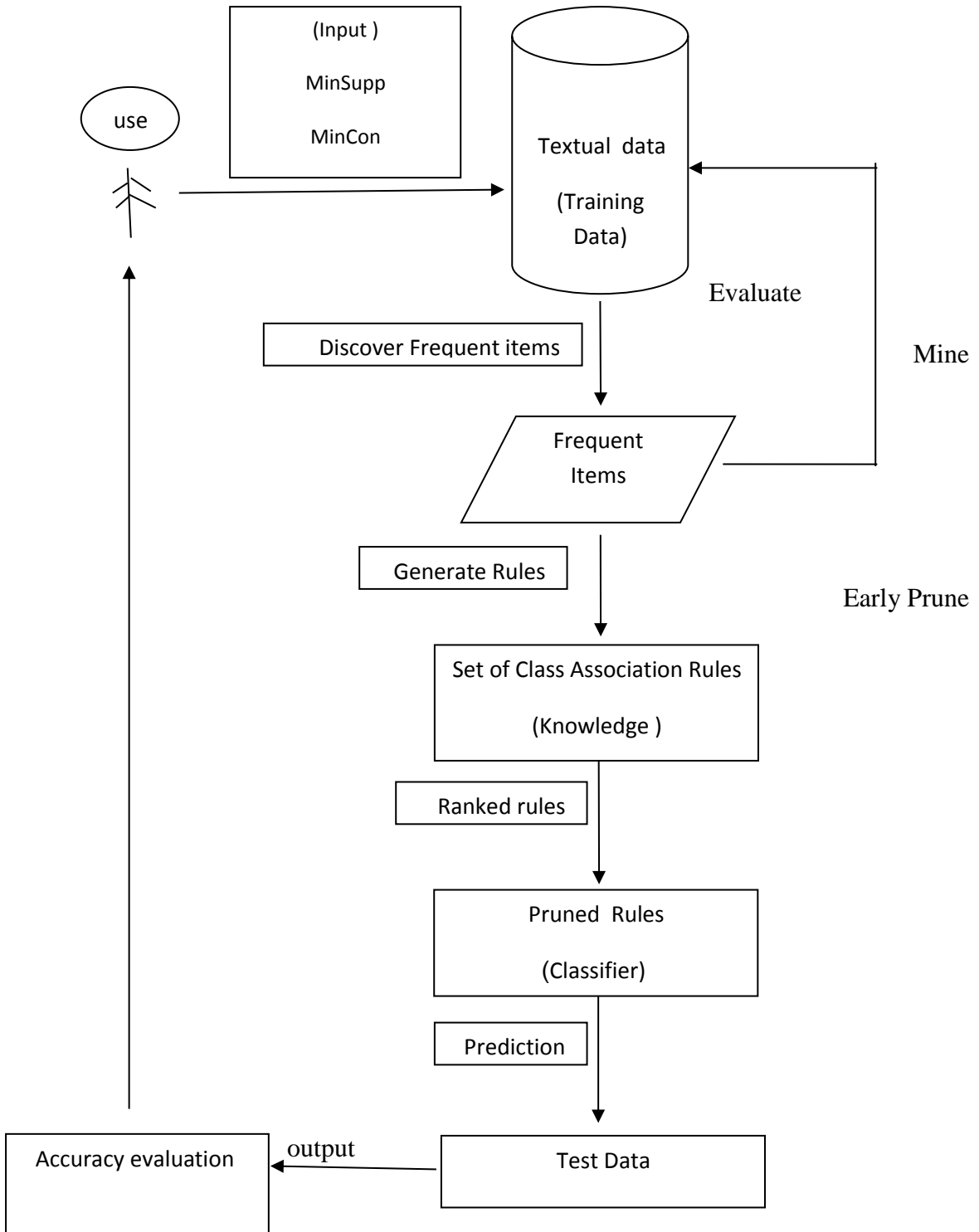
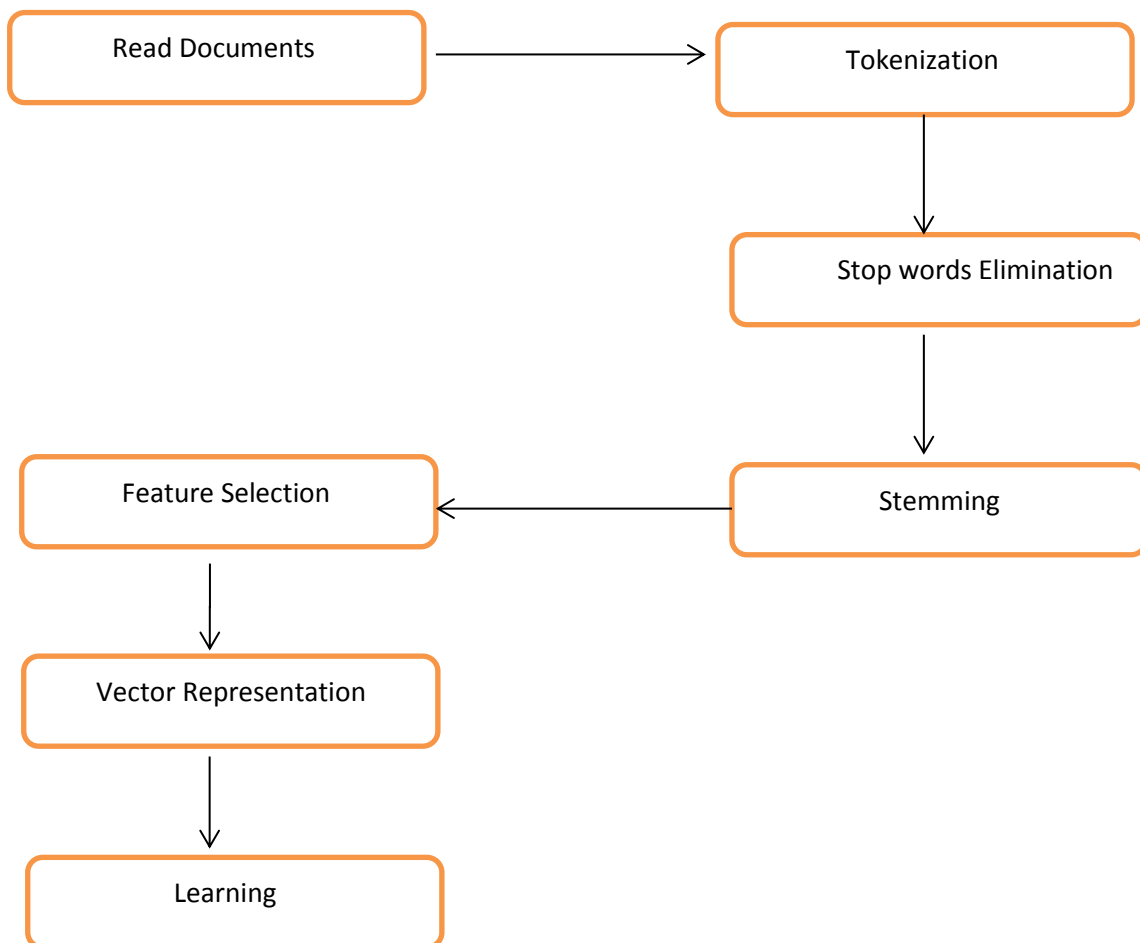


Figure (3.1) main phases of the proposed model

### 3.2.1 Preprocessing Phase

Preparing the input data for mining is considered as an important phases in TC. Since the input textual data sets are often sparse, unstructured, and it may contain noise like records redundancy, incomplete transactions, missing values, and so on .Therefore, the quality of the constructed models is significantly affected by the quality of the input data set. Figure(3-2 )shows the preprocessing phase of our model which consists of : feature selection, vector representation ,tokenization, stop word elimination, and stemming, that are usually applied in order to reduce the possibility of increasing error rate during evaluation step .



Figure(3.2) : Pre-processing operation in TM

### 1- Stopwords Filtering

Text documents usually contain a lot of words that are not useful for the learning algorithms such as ('is', 'that', 'the'). Such words should be removed within preprocessing phase, because these words negatively affect classifier construction (Khan et. al., 2010). In the proposed model, we adopted the most popular technique which is SMART stop word list (Salton,1989) since it is effective and has been utilized in many previous research on TC.

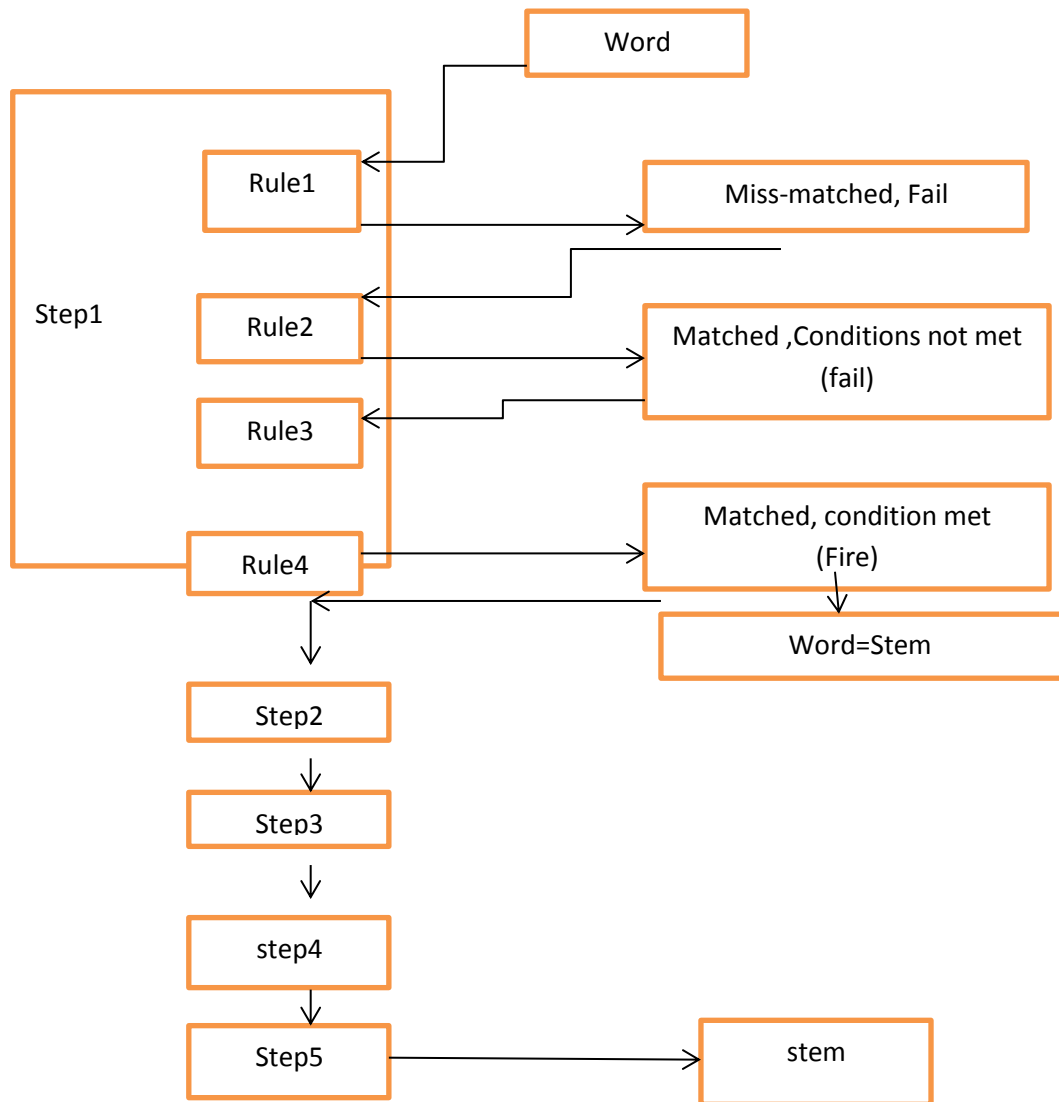
### 2- Tokenization

Tokenization is a process that includes dividing the sequence characters into more meaningful Tokens. In TC, the text document is divided into sentences, and words .We use WEKA (Weka, 2002) filtering in the Explorer GUI to tokenize the input database.

### 3- Stemming

As mentioned earlier, stemming is the process of reducing derived words to their root, for example, 'playing' to 'play' , 'construction' to 'construct', 'diver' to 'dive'. We used in the proposed model a popular technique which is porter stemmer (porter,1980). Porter algorithm proposed five applied steps of word transformation. Each step consists of group of rules in the form (condition)(suffix)→ (new suffix). For instance, a rule  $(m > 0) \text{ EED} \rightarrow \text{EE}$  , means “if the word has at least one vowel and consonant plus EED to the end of the word ,convert the ending to "EE". So "agreed" becomes "agree" while "feed" is unchanged ,Figure (3.3 ) illustrates Porter stemmer operations .





**Figure (3.3) : porter stemmer operations(if any)**

#### 4- Data Representation

We adopted the vertical representation to represent the input data. In the vertical representation, the data is arranged as a group of columns, every column has a key identifier, it is called item identifier (IID) and a group of transaction identifiers (TIDs) (Zaki & Gouda, 2003). While in the horizontal representation, the data is arranged as a group of rows. Where each row has a key identifier that is the transaction identifier (TID) and a group of IIDs (Item Identifiers). A major disadvantage of using horizontal data format is the multiple data scans while searching for frequent items that require high computational resources. Vertical representation have shown to be effective and outperformed the horizontal approaches (Song, and Rajasekaran,2006).We can use this advantage by representing frequent keywords using TID set intersections, rather than repetitive scans of the horizontal algorithms(Zaki & Gouda,2003). Also in the vertical mining, the candidate generation process is done in a single step since vertical mining provides natural pruning of unrelated transactions as a result of an intersection. Figures (3-4) and (3-5) show the difference between vertical and horizontal data representation.

A	C	D	T	W
1	1	2	1	1
3	2	4	3	2
4	3	5	5	3
5	4	6	6	4
	5			5
	6			

**Table ( 3.1 ) vertical representation of data**

1	A	C	T	W	
2	C	D	W		
3	A	C	T	W	
4	A	C	D	W	
5	A	C	D	W	T
6	C	D	T		

**Table (3.2) : Horizontal Representaion of data**

#### 5- Features Selection

Most known TC algorithms represent a document as bag of words (BOW) ,the result of using the BOW is an explosion in the number of features .The major disadvantage of this representation is the high dimensions of feature space and information loss of the original texts. Feature selection (FS) (Yang & Pederson ,1997) is one technique to solve such problems. The main idea of feature selection is to select a group of frequent terms that appears in the training set and employing this group as features in TC. In other words to treat the high dimensionality problem, most specialists choose certain keywords to represent the complete text collection. Then, the selected keywords are converted into a simpler form, e.g. matrix, that represents the whole documents collection in order to minimize data complexity and processing costs especially during learning the rules.

In our model we have used a Chi-square(Yang & Pederson, 1997) to transform the high dimensionality of the Reuter text collection into a numeric matrix and then we use simple TID list intersections to compute the frequent items. The Chi-square is generally employed to evaluate the independence of two random variables, where two variables A and B are considered to be independent if  $P(AB)=P(A)P(B)$  or  $P(A/B)=P(A)$  and  $P(B/A)=P(B)$ . In TC, the two random variables represent the frequency of the term (t) and frequency of the class (c). Chi-square technique evaluate the independence between t and c as following :

$$X^2(t,c) = N \times (AD-CB)^2 / (A+C) \times (B+D) \times (A+B) \times (C+D) \dots\dots\dots(3-1)$$

where A denotes the number of documents term (t) frequency related with class(c), B denotes the number of documents term (t) frequency but it is not related with class(c), C denotes the number of documents term t does not exist with class(c), D denotes the number of documents term t does not exist while class is not (c )and N denotes the total number of documents.

### 3.2.2 Rule Learning

After data pre-processing and extracting the frequent words, we represented these keywords in an efficient manner to ease the data processing using vertical mining. As stated earlier, vertical mining have shown to be effective and outperform horizontal mining due to the fact that frequent items can be computed using TID set intersections in one data scan instead of the repetitive scans of horizontal algorithms. In our model we used dEclat algorithm( Zaki & Gouda,2003) that utilized the vertical database layout

in which we stored the difference of TIDs(Transaction ID's) called (diffset) between a candidate k item set and its prefix k-1 frequent item sets, rather than the tids intersection set .We calculated the support by subtracting the cardinality of diffset from the support of its prefix k-1 frequent item set. It is worth mentioning that we are first researchers who adopt dEclat vertical mining in constructing classification models in TC .

```

DiffEclat([P]):
for all  $X_i \in [P]$  do
for all  $X_j \in [P]$ , with  $j > i$  do
 $R = X_i \setminus X_j$ ;
 $d(R) = d(X_j) - d(X_i)$ ;
if  $d(R) \geq \text{min sup}$  then
 $T_i = T_i \cup R$ ; //  $T_i$  initially empty
for all  $T_i \neq \emptyset$ ; do DiffEclat( $T_i$ );

```

Figure (3.4) Pseudo-code for rule discovery (Zaki and Gouda, 2003)

dEclat is an enhancement of an earlier algorithm called Eclat (Zaki ,et al, 1997) that uses a vertical database transaction layout, where frequent items are obtained by applying simple tid-lists intersections, without the need for complex data structures.

This considerably reduces the size of the memory required to store the tids. The diffset approach avoids storing the complete tids of each item, rather the difference between the class and its member itemsets are stored. Two items share the same class if they share a common prefix. A class represents items that the prefix can be extended with to obtain new class. For instance, for a class of items with prefix  $x$ ,  $[x] = \{a_1, a_2, a_3, a_4\}$ , one can perform the intersection of  $x a_i$  with all  $x a_j$  with  $j > i$  to get the new classes. From  $[x]$ , we can obtain classes  $[x a_1] = \{a_2, a_3, a_4\}$ ,  $[x a_2] = \{a_3, a_4\}$ ,  $[x a_3] = \{a_4\}$ .

Experimental results on real world data and synthetic data (Zaki and Gouda, 2003) revealed that dEclat and other vertical techniques like Eclat usually outperform horizontal algorithms like FP-growth with regards to processing time and memory usage. Furthermore, dEclat outperforms Eclat on dense data, whereas the size of the data stored by dEclat for sparse databases grows faster than that of Eclat. Consequently, the authors concluded that for dense databases, it is better to start with a diffset representation; however, for sparse databases, it is better to start with a tid list representation then switch to a diffset at later iterations. Figure (3-7) illustrates rules generating using dEclat.

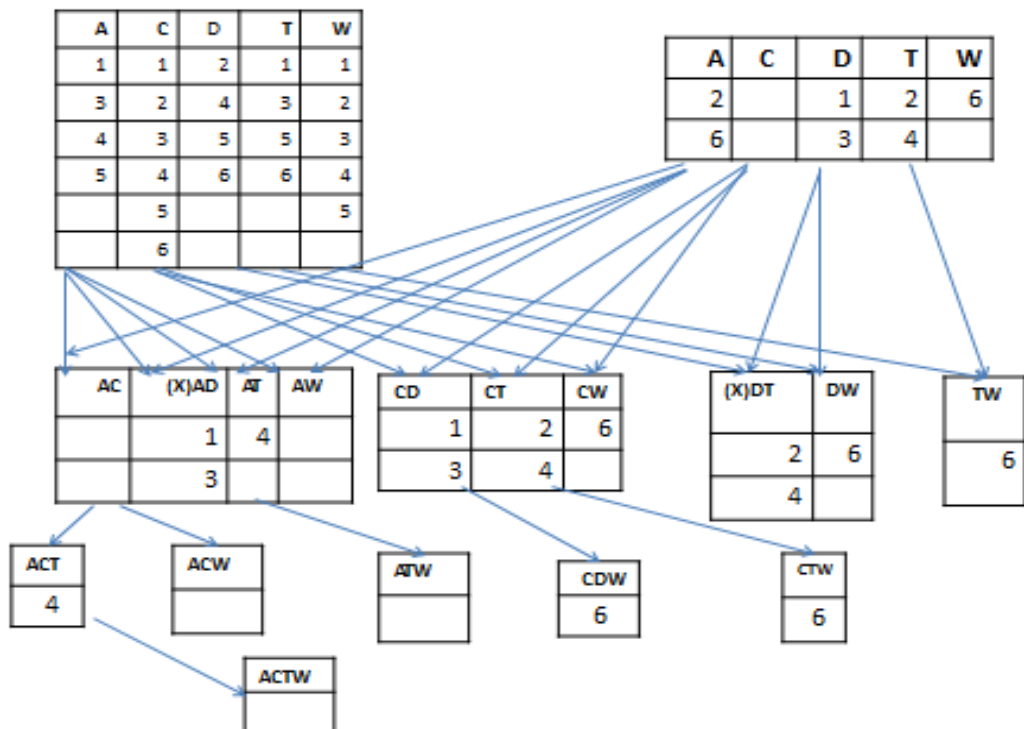


Figure (3-5) dEclat algorithm illustration(Zaki and Gouda, 2003)

### 3.2.3 Rule Ranking

After we generate the rules, an important step here is to rank these rules because the top sorted ranking rules play an important role in the classifying test data. The majority of AC algorithms such as those in (Wang et al. 2011), (Liu et al. 1998) employ rule ranking process for selecting the classifier within pruning phase. The precedence of the rules is determined according to several measures involving confidence, support and rule antecedent length. For our model we used rule ranking process according to Figure (3-7) since it simple and effective approach (Liu et al., 1998) .The ranking technique considers confidence and support in order to rank rules, and when two rules have the same support and confidence, the ranking here is based on which rule is generated earlier.

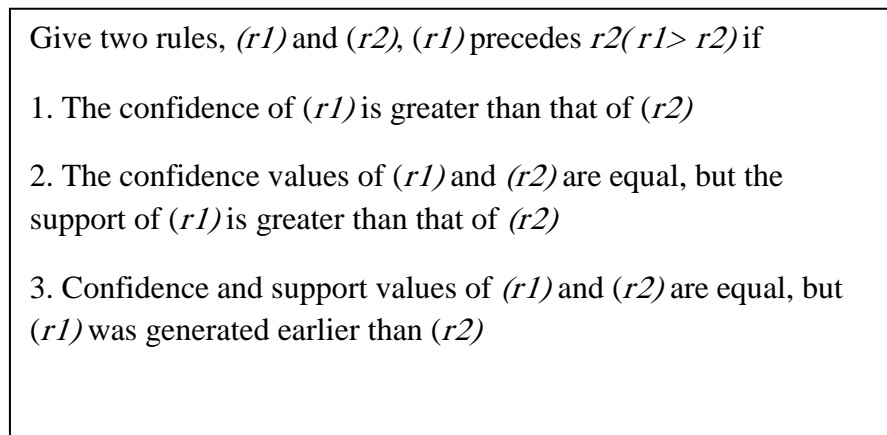


Figure (3-6) Rule ranking process(Thabtah et al,2007)

### 3.2.4 Classifier Construction

The new rule evaluation method used is illustrated in Figure (3-8). This method is used after rules have been generated and ranked .We check the produced rules by comparing them with the training data set cases, and only rules that cover at least one

training case which not considered by any other higher ranked rule are kept for later classification. So each ordered rule starting from the top ranked one ( $r_1$ ), we find all cases that match  $r_1$ 's body . Once these training cases are located , then they will be eliminated and  $r_1$  will be appended into the classifier. If the rule does not cover any training case it will be removed. The process is repeated for the remaining ordered rules until all training cases are covered or all ordered rules have been considered. This method is different than CBA one in that we consider partly coverage between the rule and the training cases .Further we do not require that the rule class must be similar to the training case class so the rule can be considered and thus reduced overfitting. This will result in less number of rules because a rule now has larger training coverage.

<p>Input: candidate rules <math>R</math> generated ,and the training data set <math>T</math> ,  Output: Classifier (<math>C</math>)  Rank <math>R</math> in descending order based on ranking procedure of confidence, support, rule length, class distribution</p> <ol style="list-style-type: none"> <li>1. For each rule <math>r_i</math> in <math>R</math>,do</li> <li>2. begin</li> <li>3. Find all applicable data in <math>T</math> that matches only <math>r_i</math>'s body</li> <li>4. If <math>r_i</math> covers a training data in <math>T</math></li> <li>5. Insert <math>r_i</math> into the <math>C</math></li> <li>6. Delete all data covered by <math>r_i</math></li> <li>7. End if</li> <li>8. end</li> <li>9. If there are still data in <math>T</math> not covered</li> <li>10. Begin</li> <li>11. For each remaining rule <math>r_j</math> in <math>(R-C)</math> do</li> <li>12. If <math>r_j</math> partly covers any data in <math>T</math></li> <li>13. begin</li> <li>14. Insert <math>r_j</math> into the <math>C</math></li> <li>15. Remove all data in <math>T</math> covered by <math>r_j</math></li> <li>16. End</li> <li>17. End if</li> <li>18. Any remaining rules in <math>R</math> gets removed</li> </ol>
---

**Fig. 3-7**The proposed rule pruning method



The classifier builder procedure which we propose in this chapter aims to (1) Reduce the number of rules extracted, (2) Evaluate the influence of rule filtering on predictive accuracy. We have considered two conditions when testing a rule in the classifier building phase. The first condition considers the selected rule's body as a whole piece with that of the training example, and when this fails the second condition considers part of the rule's body rather than leaving the training example to be considered for the default class label. We have considered both conditions in the rule filtering method in the proposed model and during building the classifier rules.

To elaborate on partly matching, it means at least one of the items of a rule (left hand side), e.g.  $R$ , matches an item of a training example  $t$ . This condition is usually tested when the training example cannot be covered by any discovered rule. When this happens  $R$  is marked as a classifier rule and inputted into the classifier. The reason for the partial matching when building the classifier is to allow more rules to participate in the prediction step of test data, which may improve the accuracy.

In the remainder of this section we distinguish between our classifier builder method and those of MCAR and CBA using example. Consider Table 1 that shows candidate rules and Table 2 that lists 6 training data. Please note that the last two columns of Table ? denotes the rule that have been used by our method and those of (MCAR, CBA) respectively. For training data number (1), the proposed classifier builder and those of MCAR and CBA use rule number (2). This is because there was a fully matching between (rule (# 2)) and training data (#1). The same thing happens for training data (#2) in which rule (#1) has been used to cover it by all methods. Though, for the third training data CBA and MCAR classifier builder has to use the default class because there is no candidate rule to cover it. This is since all candidate rule has no

match with training data (#3) and therefore both full match methods of CBA and MCAR are unable to treat this case. On the other hand, our classifier builder takes on the first partly match rule (#2) to cover training data (#3). This surely minimizes the use of the default class and may improve the classification performance of the algorithm in general. The same scenario happens again for training data (#4) in which our classifier builder method uses the partly matching rule (#6) whereas other AC algorithms utilizes the default class which in turn may increase the chance of error. This is since the default class has been formed from the unclassified examples in the training data set.

The above example shows the demonstration of the proposed classifier builder method that indeed reduces error by allowing partly matching rule to be part of the classifier instead on taking the default class label .

Table (3.3) Candidate rules

RuleID	Rule Detail	Rule Support	Rule Confidence	Rule Rank
1	Overcast $\square$ yes	0.285715	1	1
2	high $\wedge$ sunny $\square$ no	0.214286	1	2
3	normal $\wedge$ sunny $\square$ yes	0.142858	1	3
4	hot $\wedge$ sunny $\square$ no	0.142858	1	4
5	false $\wedge$ hot $\square$ yes	0.142858	0.6667	5
6	Cool $\wedge$ normal $\square$ yes	0.214286	0.75	8

Table (3.4) Part of a training data

	Outlook	Temperature	Humidity	Windy	Actual Class	Rule Applied using our method	Rule applied using MCAR and CBA methods
1	sunny	hot	high	true	no	2	2
2	overcast	hot	high	false	yes	1	1
3	rainy	mild	high	false	yes	2	Default class
4	rainy	cool	high	true	no	6	Default class
5	sunny	mild	normal	true	yes	3	3
6	sunny	mild	high	false	no	2	

### 3.2.6 Prediction phase

The prediction procedure of our model is concerned about the accuracy and according to the classifier, which is predicting new class as accurately as possible, so we ensure if the rule actually matches test data in our model when a test data to be predicted. The idea of the proposed procedure as shown in Fig (3-9) is selecting the best rules among a set of high confidence rules in order to cover the test data. In classifying a test data (line1), the proposed procedure shows that the rules in the group of ranked rules that matches the test data is obtained(line 5). If no rules matching the test data condition, our prediction procedure uses the default class rule (line 8), then all rules matches (ts) are grouped with respect to class labels and the group average confidence is calculated ,then ts is given the class belonging to the highest average confidence ,this procedure ensures that all rules applicable to the test case participate in the prediction. unlike AC algorithms that use only one rule

Input: model  $M$ , test data set  $T$ ,  $R$  : Ranked Rule ,  $max$  : maximum ranked rule ,  $d$ :Default class  
 ,Output: Prediction error rate  $P$

Given a test data ( $Ts$ ), the classification process works as follow:

- 1 For each test data  $ts$  Do
- 2 For each rule  $r$  in the set of ranked rules  $R$  Do
- 3 Find all rules that match  $ts$  and store them in  $T$
- 4 if nor rules match  $ts$
- 5 assign the default class to  $ts$
- 6 else
- 7 begin
- 8 Group  $Tr$  according to class  $c$
- 9 compute average confidence per group
- 10 assign the group class of the largest confidence to  $ts$

**Fig. 3-8 The proposed prediction method**

### 3.3 Chapter Summary

In this chapter we proposed a new model based on AC mining for the hard problem of TC. Our model proposed a novel rule pruning method that does not consider the similarity between the candidate rule class and the training data cases during building the classifier and therefore reducing overfitting on the training data set and generating moderate size classifiers. It reduces the overfitting since in our method there is no cheating when a candidate rule is about to be evaluated against the training data by looking at the training data class. Furthermore, the pruning method proposed allows the classifier rule to have larger training coverage and this has indeed overcome one major problems associated with AC algorithms which is the very large size classifiers. Another main contribution for our AC algorithm is the utilization of dEclat vertical mining during the rule discovery step. In fact we believe that we are the first researchers to use dEclat learning for classification problems. Lastly, we have improved upon current AC algorithms that utilize single rule for predicting test cases where we have developed a new procedure that takes into account multiple rules applicable to the test case in making the prediction of the class for test cased. This surely limits the use of one rule and allows more than one rule to contribute into the class assignment task. In the next chapter, we test the proposed AC model on real world textually data set (Reuter) and contrast it with other known classification algorithms in TC research field. The main evaluation measures that we will use is the accuracy, precision, recall and harmonic mean.

## Chapter Four

### EXPERIMENTAL EVALUAION

#### 4.1 Introduction

In this section, different classification algorithms are compared with the proposed model according to a number of TC evaluation measures including accuracy, number of rules produced, precision, recall and F1. The data set used in the experiments is the Reuter TC corpus and four different classification techniques: Decision trees (C4.5) (Quinlan, 1993), Naïve Bayes algorithm (Langley and Thompson, 1992), and the k-nearest neighbour algorithm (k-NN) (Tam et al, 2002) have been contrasted with our model. The reason behind selecting these algorithms is to study the performance of our algorithm comparing to classic algorithms as well as AC ones. Another reason for selecting these algorithms for comparison is the different training strategies they use in discovering the rules. For example, Naïve Bayes is a probabilistic classification algorithm. C4.5 employs divide and conquer to construct decision trees as a classifier. The decision of which attribute goes into the root node is based on mathematical formulas such as Entropy or Gini Index (Mehta et al,1999). Further, Knn depends on the similarity between documents and number of training data (K) , then determining the class of document. Lastly, MCAR is an AC algorithm that discovers hidden correlations among attribute values and the class attribute in historical classification data.

## 4.2 Data Set

A popular TC collection called Reuter has been investigated in this section (Lewis, 1998). The Reuters-21578 is the most widely used test collection for TC research. We used the ModApte version of Reuters-21578. This split leads to a corpus of 9,174 documents consisting of 6,603 training and 2,571 testing documents, respectively. We tested all contrasted algorithms on the seven most populated categories with the largest number of documents assigned to them in the training data set. The experiments are conducted on 2.5 GHz Pentium IV machine with 2GB RAM, and the proposed method is implemented using VB.Net programming language with a MinSupp and MinConf of 2%, and 40%, respectively.

The *minsupp* has been set to 2% since more extensive experiments reported in (Liu, et al., 1998; Li, et al., 2001, Thabtah, et al., 2005) suggested that it is one of the rates that achieve a good balance between accuracy and the size of the classifiers. The confidence threshold, on the other hand, has a smaller impact on the behaviour of any AC method and it has been set to 40%. Table 4.1 represents the number of documents for each category in the Reuters-21578 data set. On these documents we selected the top 1000 features using Chi Square.

In the following sections, we show the results for our proposed prediction methods and other well-known algorithms on the Reuter data collection.

**Table(4.1) Number of documents per category (Reuters-21578)**

Category Name	Training set	Testing Set
Acq	1650	719
Crude	389	189
Earn	2877	1087
Grain	433	149
Interest	347	131
Money-FX	538	179
Trade	369	117
<b>Total</b>	<b>6603</b>	<b>2571</b>

### 4.3 Environment

WEKA is an open source business intelligence tool that implements different machine learning and data mining methods. This business intelligent tool was developed within the computer science department at the University of Waikato in New Zealand (Witten and Frank, 2000). WEKA was implemented using a JAVA Object Oriented programming language, which is practical for researchers, post graduate students, lecturers, and scientists interested in business intelligence all over the world to discover hidden information from textual and numeric data collections (Hall, et al, 2009).

WEKA contains a collection of packages where each holds a group of classes for each data mining and machine learning tasks. For instance, there is a package for the

classification task, which contains a set of various classification algorithms such as decision trees, rule induction, statistical, etc. Another package is for clustering, which contains most of the common clustering methods, such as K-means, EM, etc. In general, the most of common supervised learning, unsupervised learning, pre-processing methods of data mining and machine learning are included within these packages of WEKA.

There are two general ways to access WEKA applications: one through a Graphical User Interface (GUI) (Explorer, Knowledge Flow, Experimenter), and the second is through a simple Command Line editor (SimpleCLI). In most cases experienced users prefer the SimpleCLI since it allows some sophisticated features, such as outputting the produced models to external files (Witten and Frank, 2000). However, the majority of WEKA users prefer to use the GUI applications, especially, the Explorer owing to its simplicity (drag and drop) and flexibility. Figure (4.1) illustrates the main environment of WEKA and its associated applications.

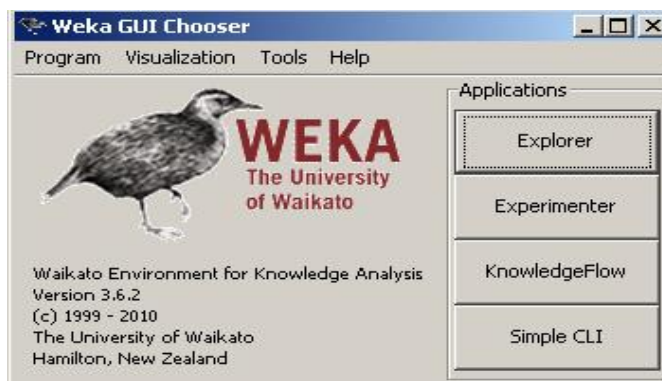


Figure (4.1): Weka GUI Main interface

As stated above, the Explorer is the main application used by most of the WEKA users. In this application a user may upload input data collection through three different ways: 1) Simple text files, 2) Uniform Resource Location (URL) and 3) External relational



database. The easiest method of uploading the source data file is the text file format (TXT) or in WEKA so called “Attribute-Relation File Format” (ARFF). In addition, recent versions of WEKA accept source files which are type of spreadsheet (CSV) (Hall, et al, 2009).

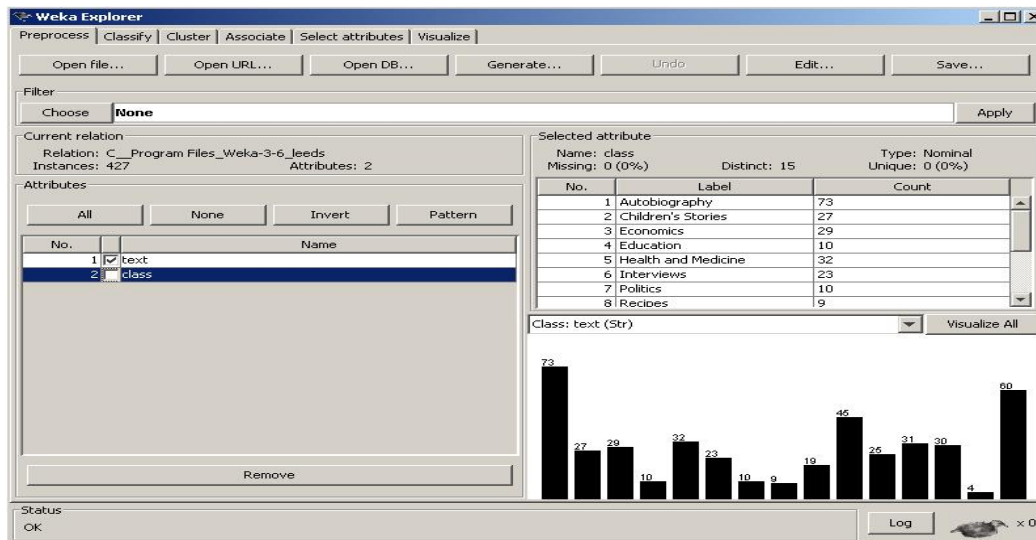


Figure (4.2): Weka Explorer Interface

The Explorer Figure(4.2) has 6 tab pages which are activated once the user has uploaded the data set. “Preprocess” tab page is responsible for data pre-processing such as terms removal, discretisation, etc. “Classify” tab page contains the classification algorithms for the supervised learning such as decision tree, statistical , rules, etc. “Cluster” tab page is for the unsupervised learning clustering algorithms like EM, K-means, CobWeb, etc. “Associate” tab page holds the unsupervised association rule mining algorithms such as Apriori, predictive Apriori, etc. “Select Attributes” tab page contains feature selection methods for data representation and dimensionality reduction. Lastly, “Visualize” is a graphical representation for the data set which is used for mining. In general, WEKA is a very useful intelligence tool as it consists of variant packages that can be utilised in different fields according to user’s preferences.

#### 4.4 Performance Evaluation Measures

In this section, we describe effectiveness measures to evaluate the text classification methods. Effectiveness is a measure of ability of the system to satisfy user in term of the relevance of documents retrieved. For text classification, we define it as a measure of ability of the system to classify documents into their appropriate categories.

Consider a simple contingency table from a binary decision system as shown by Table 2.2. The system makes  $n$  binary decisions, each of which has exactly one correct answer, either Yes or No (Lewis, 1991). Each entry specifies the number of decisions of the specified type. For example, (a) is the number of times the system decided Yes, where Yes was the correct answer. And (b) is the number of times the system decided Yes (incorrect decision), where No was the correct answer.

**Table 4.2: The contingency table for a set of binary decisions**

Category		Expert Judgements		
		Yes	No	
Classifier judgments	Yes	$a$	$b$	$a + b$
	No	$c$	$d$	$c + d$
Total		$a + c$	$b + d$	$a + b + c + d = n$

The following are two important measures of effectiveness that are derived from the contingency table:

Recall (R) is the number of categories correctly assigned divided by the total number of correct categories that should be assigned:

$$R = a / (a + c) \dots\dots\dots(2-1)$$

Precision (P) is the number of categories correctly assigned divided by total number of categories assigned:

$$P = a / (a + b) \dots\dots\dots(2-2)$$

The effectiveness measures could be misleading if we examined the recall and precision alone. We may sacrifice precision to obtain a high recall, and vice versa. To summarize and make composite measures, we use the F1-measure as an evaluation criterion (Van Rijsbergen, 1979):

$$F1 = 2PR / P + R \dots\dots\dots(2-3)$$

In single-label TC, effectiveness is usually measured by accuracy (A) and error rate (E) (Yin and Han, 2003; Thabtah et al., 2005; Li et al., 2008). The accuracy is percentage of correct classification decisions but error is the converse of accuracy, mathematically can we say  $E = 1 - A$ .

Given the contingency table as shown in Table 2.2, accuracy and error are defined as:

$$A = (a+d) / (a+b+c+d) \dots\dots\dots(2-4)$$

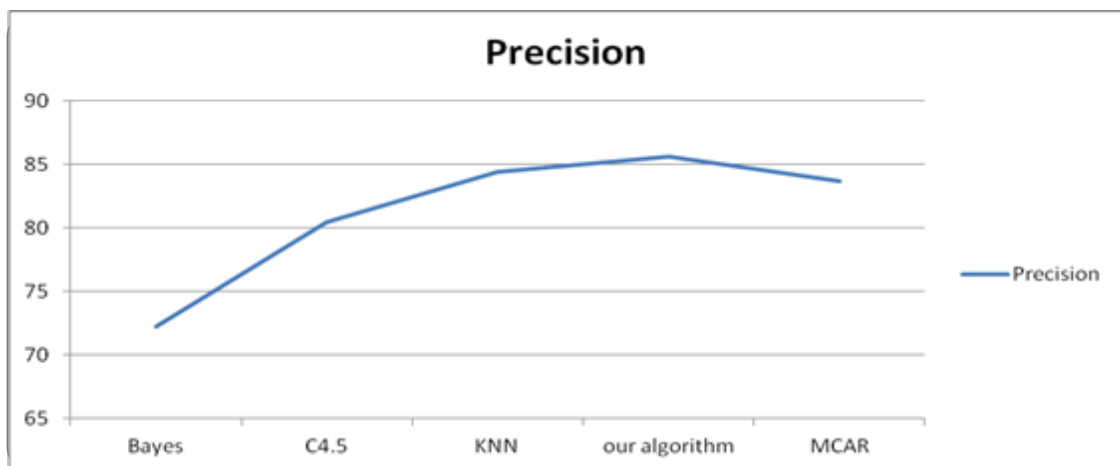
$$E = (b+c) / (a+b+c+d) \dots\dots\dots(2-5)$$

## 4.5 Results Analysis

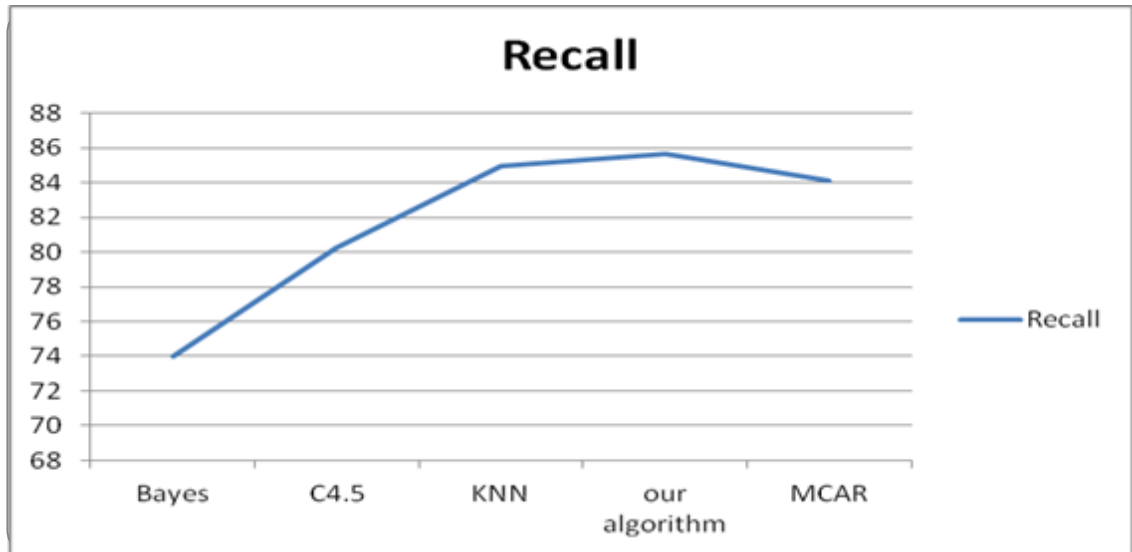
Figure (4.1) represents the average precision results for the contrasted classification algorithms. The numbers in that figure give an indication that the proposed algorithm outperformed the remaining learning algorithm on the Reuter data set in regards to precision evaluation measure. In particular, it achieved higher precision on average 13.36%, 5.19%, 1.27% and 1.89% respectively than Naïve bayes, C4.5, Knn, and MCAR algorithms, respectively. The second best performed algorithm with reference to precision was KNN but in general AC algorithms including our model and MCAR performed very competitive on precision figures if contrasted to traditional

classification algorithms. The results on precision showed that the statistical method of Naïve bayes was the least scored precision algorithm on the Reuter TC data set.

Figure (4.2) shows again that the Naïve bayes algorithm is the least applicable classification approach towards the Reuter English data set due to the low results of recall. This is since Naïve bayes algorithm derives often high class likelihood for majority class labels since they are often more representative in the input textual collections like “Earn”. In the Reuter data collection, there are some class labels that are connected with low numbers of documents. Further, Naïve bayes algorithm is based on independence assumption which is violates real-world data and perform very poorly when features are highly correlated and does not consider frequency of word occurrences. On the other hand, the proposed algorithm and MCAR for instance produced classifiers that represent most of the classes in the input textual data set. Meaning they are not impacted with the unbalanced class labels for the Reuter’s documents. In general, most of the considered classification algorithms except Naïve Bayes showed competitive performance with regards to precision and recall on the data set, as their derived results are close to each other. In particular, MCAR, KNN and the proposed algorithm.



**Figure (4.3) Average precision of the contrasted algorithm**



**Figure (4.4) Average recall of the contrasted algorithm**

A deeper investigation on the performance of the documents category has been conducted to show the behaviour of the contrasted algorithms. Table (4.3) and Table (4.4) illustrate the precision, and recall results for each document category and for each algorithm we consider. The recall, and precision measures results for proposed algorithm for the document's categories are consistent except the "Trade" and "Interest" class labels which have achieved the least results. After investigating the Reuter data collection for such class labels it seems that during the prediction step, the proposed algorithm has misclassified many of the "Trade" and "Interest" to the rest of the document's class labels. This is since the "Trade" and "Interest" document categories contain general terms and the classifier has a hard time to predict its right class label. The same principle applies to the most of the classification algorithms used in the experiments in the context of "Trade" and "Interest" class labels.

**Table (4.3) Precision results for each class in the Reuter and by each algorithm**

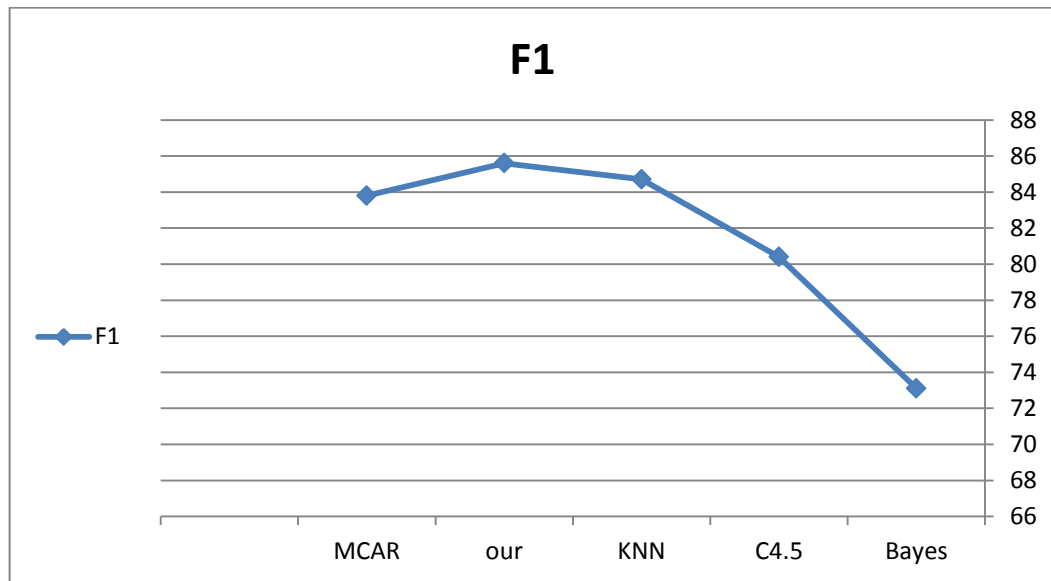
<b>Class</b>	<b>Bayes</b>	<b>C4.5</b>	<b>KNN</b>	<b>our algorithm</b>	<b>MCAR</b>
Acq	90.8	85.7	91.5	94.78	90.3
Crude	79.9	74.7	85.1	81.9	88.5
Earn	95	96.3	97.4	99.9	100
Grain	71.7	97.6	88.5	95.44	94.5
Interest	56.5	56.11	73.6	54.9	42.7
Money-FX	64	76	77.3	76.18	73.4
Trade	47.8	77.12	77.6	96.12	96.6
Average	72.2	80.5	84.4	85.6	83.6

**Table (4.4) Recall results for each class in the Reuter and by each algorithm**

<b>Class</b>	<b>Bayes</b>	<b>C4.5</b>	<b>KNN</b>	<b>our algorithm</b>	<b>MCAR</b>
Acq	92.2	84.9	92.5	94.44	90.1
Crude	82.1	76.3	86.3	83.7	87.7
Earn	96.8	95.9	97.2	99.6	99.6
Grain	73.3	98.6	87.9	96.76	96.1
Interest	59.5	48.7	74.4	53.88	44.12
Money-FX	61.8	77	79.1	76.88	75.2
Trade	52.2	80.5	77.2	94.6	95.8
Average	73.9	80.2	84.9	85.6	84

Figure( 4.3 ) shows the harmonic mean (F1) results for the contrasted classification algorithms, we notice that the F1 results of the compared algorithms are consistent with previous recall and precision results. This is because the harmonic mean measure depends on recall and precision values, as illustrated in Section (2.9) .This figure

insures that the proposed algorithm is the most suitable classification algorithms among the contrasted ones when it comes to the TC problem. Further, our classification algorithm outperformed MCAR which is a known AC technique with reference to different evaluation measures. Also the F1 figures indicate that KNN outperformed other traditional TC algorithms due to its high accurate measures results. Lastly, it seems that Naïve bayes is the least applicable algorithm to the TC problem.



**Figure (4.5) : F1 average for the contrasted algorithms**

An analysis about the proposed pruning method has been investigated by generating the proposed algorithm classifier size. We would like to answer whether the partly matching pruning method presented reduces the number of rules if compared with the known CBA or MCAR pruning methods (database coverage). Table (4.5) displays rules number for each class in our proposed algorithm and MCAR. The table indicates that our model generated less number of rules per class than MCAR, which helps in reducing execution time for the implemented model with more accurate results than MCAR. This is due to the proposed partly pruning technique which depends removing more training cases per rule ending up with reasonable classifiers.

**Table (4.5) F1 results for each class in the Reuter and by each algorithm**

<b>Class</b>	Bayes	C4.5	KNN	our algorithm	MCAR
Acq	92.2	85.2	91.9	94.7	90.2
Crude	80.9	75.1	85.6	82.7	88.1
Earn	95.8	96.1	97.3	99.7	99.7
Grain	72.4	98.1	88.1	96	95.3
Interest	58	52.4	74	54.4	43.4
Money- FX	62.9	76.5	78.2	76.5	74.3
Trade	50	78.8	77.4	95.3	96.2
Average	73.1	80.4	84.7	85.6	83.8

**Table(4.6) : Rules number for the proposed and MCAR algorithms**

Class	Our Algorithm	MCAR
Acq	26	27
Crude	4	4
Earn	16	17
Grain	5	5
Interest	1	2
Money-FX	11	12
Trade	6	6
<b>Total</b>	<b>69</b>	<b>73</b>



## Chapter 5

### CONCUSION AND FUTURE WORK

#### 5.1 Conclusion

The proposed model shows high accuracy values comparing with traditional and (AC) approaches which indicates that the proposed model is efficient and applicable for text categorization problem. We can summarize the advantages gained by our model as following :

##### 1-Training Phase Improvement

In this thesis, we developed an efficient frequent keywords method, which decreases the number of database scans to one in the learning phase. For fast discovery of frequent keywords. So, unlike most AC techniques that use Apriori multi-scan such as such as CBA and Negative-Rules to learn the rules, we use the vertical mining based on DiffSet intersection where each keywords has a list (transactions that contain that keywords in the textual data). We then use an efficient list intersection technique that requires only one data scan. We store frequent keywords and their locations (transaction Ids) during the scan. Then, by intersecting the list of the frequent keywords of size 1 we can easily obtain candidate *keywords* of size 2. Experimental tests explained later on revealed that algorithms that utilize list intersection are more effective and better than CBA based ones.

## 2- Rule Filtering

We introduced a novel rule pruning method which depends on evaluating the generated rules during building the classifier and filtering out redundant and useless rules from taking any part in the final classifier. The proposed rule filtering method checks the coverage of each rule after ranking against the training data set and keeps the rule that covers at least one training data. The test of rule coverage does not necessitate that all the rule body must be contained in the training case to cover it. Instead, partly matching is allowed in which at least one attribute value of the rule body is contained in the training data it will cover it. This ensures that a rule has large number of coverage and therefore many redundant rules will be discarded. The proposed pruning methods are discussed in details in Chapter 3. Experimental results showed that the proposed rule pruning method decreased the classifier size of the Reuter textual data if contrasted with other AC algorithms like MCAR.

## 3- Prediction of Test Data

In this thesis, we introduced a new prediction method based on group of rules to overcome the single rule prediction problems, which depends on preferring one rule only perform the prediction even if there are many rules that are applicable to the test data to cover it. The proposed class assignment method is discussed in details in chapter 3. Results against the textual data collection showed that the proposed prediction method increased the prediction rate slightly if compared with other classification algorithms. Meaning when multiple rules are utilized for predicting a test case class surely this positively effects classification accuracy of the salting classifiers.

#### 4- New AC algorithm for TC

In this thesis, we presented new AC model that combines new rule discovery method, rule pruning procedure and a prediction method into a novel algorithm applicable to mine structure and unstructured data sets. The developed AC model has been applied successfully to the hard problem of TC by treating the abovementioned three main steps (Training, rule filtering and prediction). This AC based model achieved results better than traditional TC techniques such as decision tree (Quinlan ,1999), and KNN (Tam et al ,2002) and other AC techniques like MCAR with respect to different measurement criteria including recall ,precision and F1 .

### **5.2 Future work**

- 1- The proposed model focuses on single label class , so the next work is to develop a new model for multi labels class , such as politic-economic or sport- politic .
- 2- Reducing more rules without effecting classification process and accuracy values .

## References

- Abdel Hady M, Schwenker F.(2010)" Combining committee-based semi-supervised and active learning". *journal of computer science and technology* 25(4): 681–698 DOI 10.1007/s11390-010-1053-z
- Abdel, M., Hady, F ,and Schwenker F and Palm G. (2009). "Semi-supervised learning for tree-structured ensembles of RBF networks with Co-Training", *Neural Networks*, Volume 23 Issue 4, P 497-509 ,UK .
- Abd-Elmegid L , El-Sharkawi M , El-Fangary L , and Helmy Y ,(2010)" Vertical Mining of Frequent Patterns from Uncertain Data" , *Computer and Information Science* , Vol. 3, No. 2.
- Agrawal, R., Imielinski, T. and Swami, A. (1993). "Mining Association Rules between Sets of Items in Large Databases". *Proc. Of the ACM SIGMOD Conference on Management of Data.*, p. 207-216, Washington, D.C
- Agrawal, R. and Srikant, R(1994) "Fast algorithms for mining association rules". In *Proc. of the Int. Conf. on Very Large Databases*, pages 487.499, SanDiago, Chile,
- Almuallim H(1996). "An efficient algorithm for optimal pruning of decision trees", *Artif. Intell.*, vol. 83, no. 2, pp. 347-362 .
- Antonie M and. Zaiane O(2002), "Text document categorization by term association". In *Proc. of ICDM*, pages 19.26.
- Antonie M, Zaiane O, Holte R(2006) ," Learning to Use a Learned Model: A Two-Stage Approach to Classification, Data Mining". *ICDM '06. Sixth International Conference* ,p 33-42 .
- Apte C , Damerau F ,and Weiss. S (1994.) "Towards Language Independent Automated Learning of Text Categorization Models", In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, p. 23-30.
- Baralis E., Chiusano S. and Garza P. (2008)." A Lazy Approach to Associative Classification". *IEEE Trans. Knowledge Data Engineering*. 20(2): 156-171.
- Baralis, E, Chiusano, S. and Graza, P.( 2004) ." On support thresholds in associative classification". In *Proceedings of the ACM Symposium on Applied Computing*. ACM Press, pp. 553–558, Nicosia, Cyprus.

Baralis, E and Torino, P.( 2002) ." A lazy approach to pruning classification rules". Proceedings of the IEEE International Conference on Data Mining (ICDM'02), p. 35, Maebashi City, Japan .

Belkin N and Croft W (1992)." Information filtering and information retrieval: two sides of the same coin?" Communications of the ACM, 35(12):p29–38.

Briemann L , Friedman J , Olshen R and Stone C.(1984)." CART:Classification and Regression Trees", EBelmont, s CA:Wadsworth Statistical Press, 1984.

Breslow L and Aha W . (1996) "Simplifying Decision Trees: A Survey", Technical Report No. AIC-96-014, Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory Washington, DC.

Brücher H , Knolmayer G, and Mittermayer M ,(2002)“Document Classification Methods for Organizing Explicit Knowledge”, Research Group Information Engineering, Institute of Information Systems, University of Bern, Engehaldenstrasse 8, CH - 3012 Bern, Switzerland.

Byardo, R. J. (1997) " Brute Force Mining of High Dimensional Classification Rules". In Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD-97), AAAI Press,P 123-126.

Chisholm, E. and Kolda, T.F. (1998) "New term weighting formulas for the vector space method in information retrieval", Technical Report, Oak Ridge National Laboratory.

Creswell, J., (2005). "Research design qualitative, quantitative, and mixed methods approaches"(2nd ed)., Thousand Oaks [u.a.]: Sage.

Debole, F., Sebastiani, F .(2003 ) " Supervised Term Weighting for Automated Text Categorization". In Proceedings of SAC-03, 18th ACM Symposium on Applied Computing. ACM Press 784–788,Pisa ,Italy

Dhanabal S and Chandramathi.S (2011)" Article: A Review of various k-Nearest Neighbor Query Processing Techniques". International Journal of Computer Applications 31(7):14-22., New York, USA

Diao, Q. and Diao, H. (2000) "Three Term Weighting and Classification Algorithms in Text Automatic Classification", The Fourth International Conference on High-Performance Computing in theAsia-Pacific Region, Vol. 2, P.629.

Elmasri, R. and Navathe, S.(2003) "Fundamentals of Database Systems" (4th Edition), Addison Wesley.

Esposito F, Malerba D , and Semeraro G(1997). "A comparative Analysis of Methods for Pruning Decision Trees", *IEEE transactions on pattern analysis and machine intelligence*,19(5): pp. 476-491

Eui-Hong (Sam) H, Karypis G, and Kumar V;(1999)“ Text Categorization Using Weighted Adjusted k-Nearest Neighbor Classification”, Department of Computer Science and Engineering. Army HPC Research Centre, University of Minnesota, Minneapolis, USA.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. and Uthursamy R(1996b),"Knowledge Discovery in Databases", AAAI/MIT Press.

Feldman R (1998). "Knowledge Management: A Text Mining Approach". In *Proceedings of the 2nd Conference on Practical Aspects of Knowledge Management*. Basel, Switzerland .

Frans, C., Paul, L. and Lu, Z.(2005) "Threshold Tuning for improved Classification Association Rule Mining". In *Proc. Of PAKDD 2005, LNAI 3518*, pages 216-225.

Frietas, A(2000). " Understanding the Crucial Differences between Classification and Discovery of Association Rules" . A Position Paper, *SIGKDD, ACM*, pages 1-5.

Greiner R ,and Schaffer J (2001)." AI Exploratorium – Decision Trees, Department of Computing Science", University of Alberta,Edmonton,ABT6G2H1, Canada.

Ghahreman N, Dastjerdi A (2011) " Semi-Automatic Labeling of Training Data Sets in Text Classification " *Canadian Center of Science and Education Vol. 4, No. 6*.

Han W, and Pei J , (2001):"CMAR: Accurate and efficient classification based on multiple class-association rules". In *Proc. of ICDM*, pages 369.376 .

Han J and Kamber M. (2006)"Data Mining: Concepts and Techniques".

Hersh W (2004):" A Health & Biomedical Perspective" (Second Edition). , M.D. *Information Retrieval, Springer-Verlag* .

Hong H., and Li J.(2005)" Using Association Rules to Make Rulebased Classifiers Robust", In *Proceedings of Australian Database Conference, Vol. 39, Newcastle, Australia*.

Joachims, T. (1998). " Text categorisation with support vector machines: Learning with many relevant features". *Proceedings of Tenth European Conference on Machine Learning*, (pp. 137-142). Chemnitz, Germany.

Karbasi, S. and Boughanem, M. (2006)" Document length normalization using effective level of term frequency in large collections", *Advances in Information Retrieval, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 3936/2006,Pp.72-83*.

Khan A , Baharudin B , Hong Lee L, and khan K(2010) ," A Review of Machine Learning Algorithms for Text-Documents Classification", journal of advances in information technology , vol.1 , no.1

Kim, J., Lee, B., Shaw, M., Chang, H., and Nelson W(2001)“Application of Decision-Tree Induction Techniques to Personalized Advertisements on Internet Storefronts”, International Journal of Electronic Commerce 5(3) pp.45-6.

Kohavi R and Provost F . (1998). "Glossary of Terms." In: Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, Machine Learning, 30(2-3). DOI:10.1023/A:1017181826899

Kroeze, J. Matthee, M. and Bothma, T, (2003) "Differentiating between data-mining and text mining terminology", ACM: Proceeding of the 2003 annual research conference of the south African institute, Vol. 47, PP.93-101.

Kundu G, Islam M., Munir S. and Bari M. (2008)." ACN: An Associative Classifier with Negative Rules", Computational Science and Engineering, vol. 0, no. 0, (pp. 369-375), 11th IEEE International Conference on Computational Science and Engineering.

Kui Yu, Xindong Wu<sup>1</sup>, , Wei Ding, Hao Wang, and Hongliang Yao(2008)," Causal Associative Classification", National Natural Science Foundation of China (60975034, 61070131, 61175051 and 61005007), the US National Science Foundation (CCF-0905337)

Lan, M., Tan, C., Su,J. and Lu, Y. (2009) "Supervised and Traditional Term Weighting Methods for Automatic Text Categorization", Pattern Analysis and Machine Intelligence, Vol. 31, no. 4, pp. 721-735 .

Lee, C.,and Lee, G. (2006). "Information gain and divergence-based feature selection for machine learning-based text categorization". Information Processing and Management, 42, 155–165.

Liu B., Hsu W, and. Ma Y(1998)," Integrating classification and association rule mining". In Proc. of SIGKDD, pages 80.86.

Li, W., Han, J. and Pei, J. (2001). " CMAR: Accurate and efficient classification based on multiple-class association rule". In Proceedings of the International Conference on Data Mining (ICDM'01), San Jose,CA, pp. 369–376.

Lim, T., Loh, W. and Shih, Y.(2000) ." A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms". Machine Learning 40, 203–228.

Li W ,Qin D , and Yu C (2008) "ACCF: Associative Classification Based on Closed Frequent Itemsets",Fifth International Conference on Fuzzy Systems and Knowledge Discovery .

Li, W. (2001) . "Classification based on multiple association rules". MSc thesis, Simon Fraser University, BC, Canada .

Manning C, Raghavan P, and Schutze H , " Introduction to Information Retrieval", Cambridge", Cambridge University Press, 2008.

Mehta M, Agrwal R ,Ressanen J,(1999) "SLIQ: fast scalable classifier for data minig ",Proceeding Intel conference ,Extending Database technology (EDBT 96) ,France .

Morishita, S., and Sese, J. "Traversing Item set Lattices with Statistical Metric Pruning", In PODS, Dallas, Texas, USA, ACM, pages 226-236, 2000.

Myllymaki P , and Tirri H , "Bayesian Case-Based Reasoning with Neural Network", In Proceeding of the IEEE International Conference on Neural Network'93, Vol. 1, pp. 422-427. 1993.

Ng H , Goh W , and Low K , (1997) "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization", In Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 67-73.

Niu Q., Xia S. and Zhang L. (2009). "Association Classification Based on Compactness of Rules", wkdd, (pp.245-247), Second International Workshop on Knowledge Discovery and Data Mining.

Patil D, Wadhai V, and Gokhale J ,(2010)," Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy", International Journal of Computer Applications (0975 – 8887) Volume 11– No.2

Porter, M. (1980) "An algorithm for suffix stripping ", Program, Vol. 14, No. 3, Pp. 130–137.

Quinlan, J. (1998)" C4.5: Programs for machine learning". San Mateo, CA: Morgan Kaufmann. the KDD, (pp. 80-86). New York, NY.

Quinlan, J. R., and Cameron-Jones R.(1993)" FOIL : A mid term report", In Proc. European Conference on Machine Learning, Vienna, Austria.

Quinlan. J , (1999) "Simplifying decision trees", Int. J. Human- Computer Studies, 51, pp. 497-491.

Ruiz M. and Srinivasan P. (2002)," Hierarchical Text Categorization Using Neural Networks", The University of Iowa, Iowa City, .Swinburne, Richard, Bayes's Theorem, Oxford University Press .



Sebastiani, F. (2002). "Machine learning in automated text categorization". *ACM Computing Surveys*, 34(1), 1-47.

Song M ,and Rajasekaran S. (2006). "A Transaction Mapping Algorithm for Frequent Itemsets Mining" , *IEEE Transactions on Knowledge and Data Engineering* , Vol.18, No.4, pp. 472-481.

Swami D , Jain R ,(2005)" a survey of associative classification algorithm" , *ADIT journal of engineering* , vol 2 , no 1.

Tam, V, Santoso, A, and Setiono, R.(2002) , "A comparative study of centroid-based, neighborhood-based and statistical approaches for effective document categorization", *Proceedings of the 16th International Conference on Pattern Recognition*, pp.235–238.

Tang Z. and Liao Q. (2007). "A New Class Based Associative Classification Algorithm". *IMECS 2007*: 685-689.

Thabtah, F., Cowling, P., and Peng, Y(2004)."MMAC: A New Multiclass, Multi-label Associative Classification Approach". In *Proceeding of fourth IEEE International Conference on Data Mining (ICDM .04)*, pages 217-224. Brighton, UK .

Thabtah F , Cowling P ,and Peng Y (2005)," A Study of Predictive Accuracy for Four Associative Classifiers", *Journal of Digital Information Management*:205-209.

Thabtah, F., Cowling, P., and Peng, Y(2005)." MCAR: Multi-classClassification based on Association Rule Approach". *ADIT journal of engineering* , vol 2,no 1.In *Proceeding of third IEEE International Conference on Computer Systems and Applications* . pages 1-7. Cairo,Egypt. 2005.

Thabtah, F., Cowling, P., and Peng, Y. (2006):" Multiple label classification rules approach". *Journal of Knowledge and Information System*, 2005:1-21.

Thabtah F (2007) . "A review of associative classification mining" .*Knowledge Engineering Review*, 22 (1). pp. 37-65

Thabtah F., Mahmood Q., McCluskey L., and Abdel-jaber H (2010a). "A new Classification based on Association Algorithm". *Journal of Information and Knowledge Management*, Vol 9, No. 1, pp. 55-64. World Scientific .

Tong S, Koller D (2001) , "Support Vector Machine Active Learning with Applications to Text Classification" , *Journal of Machine Learning Research* .

Vapnik V. (1995)." *The Nature of Statistical Learning Theory*", chapter 5. Springer-Verlag, New York.

Wang, T. Y., and Chiang, H. M. (2007). Fuzzy support vector machine for multi-class text categorization. *Information Processing and Management*, 43, 914–929.

Wang X , Yue K , and Shi Z (2011)," An approach for adaptive associative classification" *Expert Systems with Applications* Volume 38, Issue 9, Pages 11873–11883.

Wang, K., Zhou, S. and He, Y. (2000 ) "Growing decision tree on support-less association rules". In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, MA: ACM Press, pp. 265–269.

WEKA: Data Mining Software in Java: <http://www.cs.waikato.ac.nz/ml/weka>

Witten, I., H., and Frank, E. (2000) "Data Mining: Practical Machine Learning Tools and Techniques", San Francisco: Elsevier Inc.

Yang, Y. and Pederson, J.O. (1997). "A comparative Study on Feature Selection in Text Categorization", In *Proceedings of the 14th International Conference on Machine Learning*, pp. 412-420.

Yang Y(2001), "A Study on Thresholding Strategies for Text Categorization", *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, USA* .

Yin, X. and Han, J. (2003):" CPAR Classification based on Predictive Association Rules". In *Proc. of the Int. Conf. on Data Mining, SDM*. SIAM.

Yoon Y. and Lee G. (2008). "Text Categorization Based on Boosting Association Rules", *IEEE International Conference on Semantic Computing*,(pp.136-143).

Zaiiane and Antonie (2005). "Lecture Notes in Computer Science", Volume 3683/2005, 176, DOI: 10.1007/11553939\_136 Zaki M and K. Gouda. (2003). "Fast Vertical Mining Using Diffsets", In *Knowledge Discovery and Data Mining (KDD)*, pp. 326-335.

Zaki. M (2000). "Scalable Algorithms for Association Mining", *IEEE Transactions on Knowledge and Data Engineering* , Vol.12, No.3, pp. 372-390

Zaki M.J., Parthasarathy S., Ogihara M., and Li W.(1997)." New algorithms for fast discovery of association rules". In *Proceeding of third KDD Conference*, pages 283-286.

Zaki M and K. Gouda. (2003). "Fast Vertical Mining Using Diffsets", pp. 326-335.

Zimmermann, A. and Raedt, L. CorClass D(2004) : " Correlated Association Rule Mining for Classification", In *Proc .Discovery Science* . 04, ,pages , 60-72. Springer Verlag, Padova, Italy

## Appendices

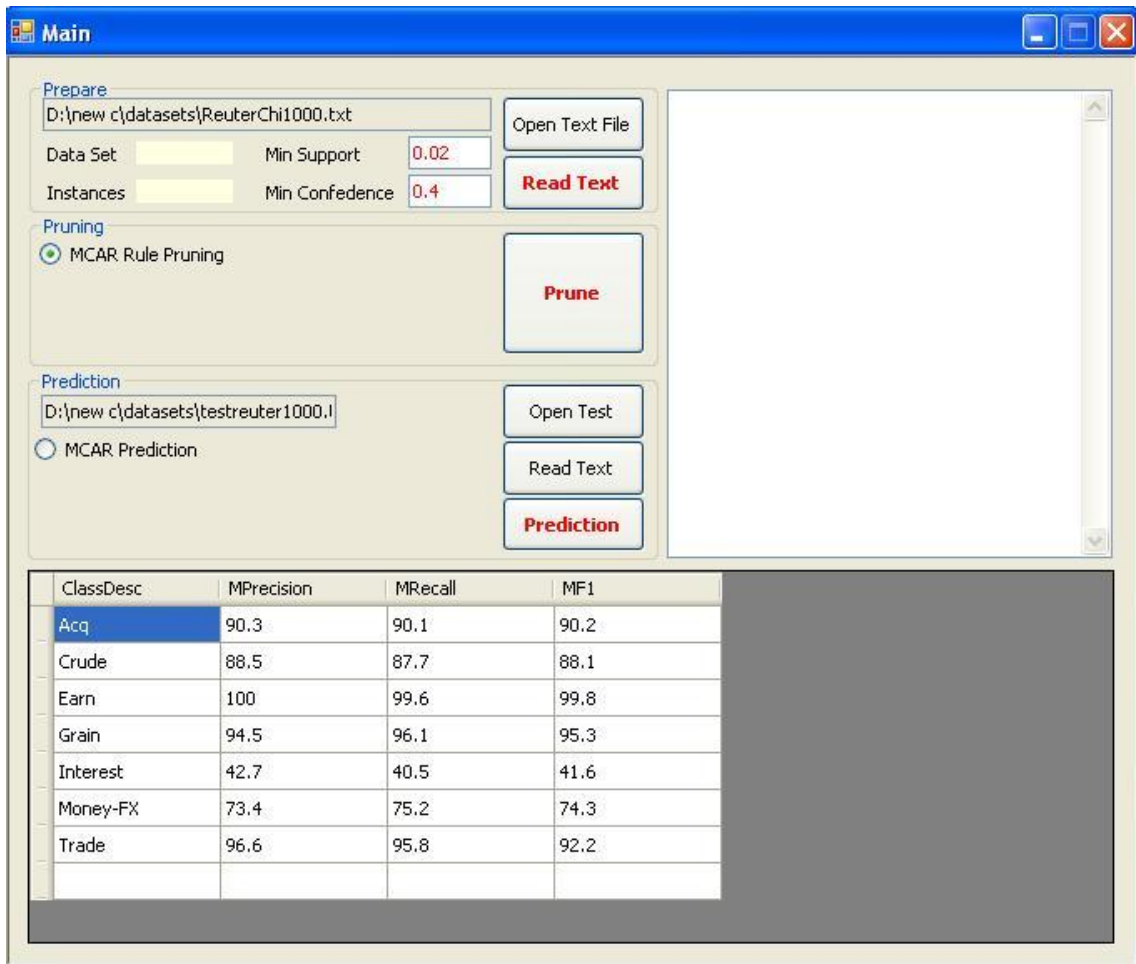


Figure (A1.1): MCAR Algorithm results

The screenshot shows a software window titled "Main" with a blue title bar. The interface is divided into three main sections: "Prepare", "Pruning", and "Prediction".

**Prepare Section:**

- File path: D:\new c\databases\ReuterChi1000.txt
- Buttons: "Open Text File" and "Read Text"
- Min Support: 0.02
- Min Confidence: 0.4

**Pruning Section:**

- Radio button:  Full Match
- Button: "Prune"

**Prediction Section:**

- File path: D:\new c\databases\testreuter1000.l
- Radio button:  Highest Rule
- Buttons: "Open Test", "Read Text", and "Prediction"

**Results Table:**

ClassDesc	MPrecision	MRecall	MF1
Acq	99.4	99.6	99.5
Crude	81.9	83.7	82.8
Earn	99.9	99.5	99.7
Grain	98.6	97.4	98
Interest	64.4	66.2	65.3
Money-FX	92.5	91.7	92.1
Trade	97	94.6	95.8

Figure (A.1.2) Our Algorithm Results