International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

TEXT CLASSIFICATION USING A CONCEPT OF SUPERVISED LEARNING ALGORITHM

Keyur J Patel, Master of Technology, Department of Information Technology, Ganpat University, Kherva, Mehsana, Gujarat, India; Ketan Sarvakar, Assistant Professor, Department of Information Technology, Ganpat University, Kherva, Mehsana, Gujarat, India;
Tejas S Patel, Master of Technology, Department of Information Technology, Ganpat University, Kherva, Mehsana, Gujarat, India;

Abstract

The process of classifying documents into predefined categories based on their content it known as text classification. It is the natural language texts to predefined categories based on automated assignment. The primary requirement of text classification is text retrieval systems, which retrieve texts in response to a user query, and text understanding systems, which transform text in some way such as producing summaries, answering questions or extracting data. This existing supervised learning algorithm is to automatically classify text need sufficient documents to learn accurately. This paper presents a new hybrid algorithm for text classification using data mining. Instead of using words, association rules e.g. word relation from these words is used to derive feature set from pre-classified text documents. And the concept of Naive Bayes classifier is then used on derived features. A system based on the proposed algorithm has been implemented and tested. To show the proposed system works as a successful text classifier in the experimental results.

Keywords: Association rule, Apriori algorithm, Naive Bayes classifier, Text classification.

Introduction

There are several text documents existing in electronic form. To a greater extent are becoming available every day. Such documents represent a massive amount of information that is easily accessible. Looking for value in this huge collection requires organization; much of the work of organizing documents can be automated through data mining. The accuracy and our understanding of such systems greatly influence their usefulness. The task of data mining is to automatically classify documents into predefined classes based on their con tent. Many algorithms have been developed to deal with automatic text classification [11]. The most common techniques used for this purpose include Association Rule Mining [1][3], Implementation of Naïve Bayes Classifier[1][3].

Association rule mining [1][3][9] finds interesting association or correlation relationships among a large set of data items [11]. The discovery of these relationships among huge amounts of transaction records can help in many decision making process. On the other hand, the Naïve Bayes classifier uses the maximum a posteriori estimation for learning a classifier. It assumes that the occurrence of each word in a document is conditionally independent of all other words in that document given its class [10].

This paper presents a new algorithm for text classification. Instead of using words, word relation i.e. association rules is used to derive feature set from pre-classified text documents. The concept of Naive Bayes Classifier is then used on derived features for final classification. A system based on the proposed algorithm has been implemented and tested. The experimental results show that the proposed system works as a successful text classifier.

The supervised learning algorithms still used to automatically classify text need sufficient documents to learn accurately while this proposed technique requires fewer documents for training. Here association rules from the significant words are used to derive feature set from pre-classified text documents. Our observed experiment on this concept shows that the classifier build this way is more accurate than the existing text classification systems.

Background Study

Data Mining [1]

Data mining refers to extracting or mining knowledge from large amounts of data. It can also be named by "knowledge mining form data". Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material. There are many other terms carrying a similar or slightly different meaning to data mining, such as knowledge mining from databases, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.

The fast-growing, tremendous amount of data, collected and stored in large and numerous databases, has far exceeded our human ability for comprehension without powerful tools. In such situation we become data rich but information poor. In addition, current expert system technologies rely on users or domain experts to manually input knowledge into knowledge bases.

Association Rule [1][3][9]

Association rule mining finds interesting association or correlation relationships among a large set of data items. In short association rule is based on associated relationships. The discovery of interesting association relationships among huge amounts of transaction records can help in many decision-making processes. Association rules are generated on the basis of two important terms namely minimum support threshold and minimum confidence threshold.

Let us consider the following assumptions to represent the association rule in terms of mathematical representation, K = {i1, i2, . . . , im} be a set of items. Let D the task relevant data, be a set of database transactions where each transaction T is a set of items such that T K. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if A T. An association rule is an implication of the form A B, where A K, B K, and $A \cap B = \Phi$. The rule A B holds in the transaction set D with support s, where s is the percentage of transactions in D that contain A B i.e. both A and B. This is taken to be the probability, P (A|B). The rule A B has confidence c in the transaction set d if c is the percentage of transaction in D containing A that also contain B. This is taken to be the conditional probability, P (B | A). That is,

$$\text{support } (A \rightarrow B) = P(AUB) \text{ and confidence } (A \rightarrow B) = P(B|A)$$

Association Rules that satisfy both a minimum support threshold and minimum confidence threshold are called strong association rules. A set of items is referred to as an itemset. In data mining research literature, "itemset" is more commonly used than "item set". An itemset that contains k items is a k-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply as the frequency, support count, or count of the itemset. An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of minimum support and the total number of transactions in D. The number of transactions required for the itemset to satisfy minimum support is therefore referred to as the minimum support count. If an itemset satisfies minimum support, then it is a frequent itemset. The set of frequent k-itemsets is commonly denoted by Lk.

The Apriori Algorithm[1][3][8][9]

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found. This set is denoted by L1. L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database.

To understand how Apriori property is used in the algorithm, let us look at how Lk-1 is used to find Lk. A two step process is followed, consisting of join and prune actions:

i) The Join Step:

To find Lk, a set of candidate k-itemsets is generated by joining Lk-1 with itself. This set of candidates is denoted by Ck. Let l1 and l2 be itemsets in Lk-1 then l1 and l2 are joinable if their first (k-2) items are in common, i.e., (l1[1] = l2[1]) . (l1[2] = l2 [2]) . . . . . . (l1[k-2]= l2[k-2]) . (l1[k-1]< l2 [k-1]).

ii) The Prune Step**:**

Ck is the superset of Lk. A scan of the database to determine the count if each candidate in Ck would result in determination of Lk (itemsets having a count no less than minimum support in Ck). But this scan and computation can be reduced by applying the Apriori property. Any (k-1)-itemsets that is not frequent cannot be a subset of a frequent k-itemset. Hence if any (k-1)-subset of a candidate k-itemset is not in Lk-1, then the candidate cannot be frequent either and so can be removed from Ck.

The algorithm is as follows:

Input: Database, D;
minimum support threshold, min_sup.

Output: L, frequent itemsets in D.

(1) L1 = find frequent 1-itemsets(D);
(2) for(k= 2; Lk-1 ≠ ø; k++)
(3) {
(4) Ck = apriori-gen(Lk-1, min_sup);
(5) for each transaction t Є D //scan D for counts
(6) {
(7) Ct = subset(Ck,t); //get the subsets of t that are candidates
(8) for each candidate c Є Ct
(9) c.count++;
(10) }
(11) Lk = {C Є Ck | c.count ≥ minimum_sup }
(12) }
(13) return L=Uk Lk;

The Apriori[1][3] achieves good performance by reducing the size of candidate sets. However, in situations with very many frequent itemsets, large itemsets, or very low minimum support, it still suffers from the cost of generating a huge number of candidate sets and scanning the database repeatedly to check a large set of candidate itemsets.

Illustration of Apriori Algorithm

Let us consider an example of Apriori, based on the following transaction database, D of figure 2.1, with 4 transactions, to illustrate Apriori algorithm.

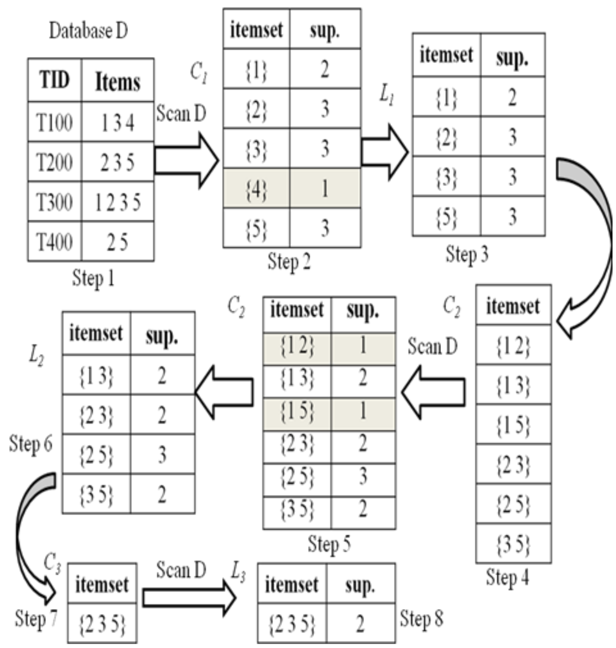

Figure 1 Apriori Algorithm

- In the first iteration of the algorithm, each item is a number of the set of candidate 1-itemsets, C1. The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.
- Suppose that the minimum transaction support count required is 2 (i.e.; min_sup = 2/5 = 40%). The set of frequent 1-itemsets, L1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support.
- To discover the set of frequent 2-itemsets, L2, the algorithm uses L1 | L2 to generate a candidate set of 2-itemsets, C2.
- The transactions in D are scanned and the support count of each candidate itemset in C2 is accumulated.
- The set of frequent 2-itemsets, L2, is then de-

ISSN:2319-7900

termined, consisting of those candidate-itemsets in C2 having minimum support.

- The generation of the set of candidate 3-itemsets, C3 is observed in step 7 to step 8. Here C3 = L1 | L2 = {{1, 2, 3}, {1, 2, 5}, {1, 3, 5}, {2, 3, 5}}. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent.
- The transactions in D are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support.
- The algorithm uses L3 | L3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{1, 2, 3, 5}}, this itemset is pruned since its subset {{2, 3, 5}} is not frequent. Thus, C4 = {}, and the algorithm terminates.

Naive Bayes Classifier [1][3][8][9]

Bayesian classification is based on Bayes theorem. A simple Bayesian classification namely the Naïve classifier is comparable in performance with decision tree and neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large database.

Naïve Bayes classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve" [8].

While applying Naïve Bayes classifier to classify text, each word position in a document is defined as an attribute and the value of that attribute to be the word found in that position. Here Naïve Bayes classification can be given by:

$$V_{NB} = \operatorname{argmax} P(V_j) \Pi P(a_j | V_j)$$

Here V_{NB} is the classification that maximizes the probability of observing the words that were actually found in the example documents, subject to the usual Naive Bayes independence assumption. The first term can be estimated based on the fraction of each class in the training data. The following equation is used for estimating the second term:

$$\frac{n_k + 1}{n + |vocubulary|} \qquad (1)$$

Where, n is the total number of word positions in all training examples whose target value is V_j, n_k is the number of items that word is found among these n word positions, and | vocubulary | is the total number of distinct words found within the training data.

The Proposed Method [1]

Our proposed method to classify text is an implementation of Association Rule with a combined use of Naive Bayes classifier. We have used the features of association rule to make association sets. On the other hand, to make a probability chart with prior probabilities we have used Naive Bayes classifier's probability measurements [1][3].

One thing that has to be noticed is that Genetic Algorithm's conventional phases like crossover, mutation is not included in this method. The only thing of genetic algorithm that we used is the matching to the desired class and mismatching to the other classes. Here he associated word sets, which do not mach our considered class is treated as negative sets and others are positive.

Experimental Evaluation

Preparing Text for Classification

Abstracts from different research papers have been used to analyze the experiment. Five classes of papers from Chemistry (CH), Computer Architecture (CA), Computer Graphics (CG), Data Structure (DS) and Web Technology (WT), were considered for our experiment. We used a total of 425 abstracts (110 from Physics, 80 from Chemistry, 65 from Algorithm, 100 from Educational Engineering and 70 from AI).

Table 1 Data sets

Data to Train						
To-tal%	**Total Amount of Data**	**CH**	**CA**	**CG**	**DS**	**WT**

15	44	16	13	4	2	9
25	72	26	21	7	4	14
35	104	36	31	10	5	22
45	135	47	40	12	8	28
55	166	57	49	17	9	34
65	199	66	57	24	13	39

To make the raw text valuable, that is to prepare the text, we have considered only the keywords. That is unnecessary words and symbols are removed. For this keyword extraction process we dropped the common unnecessary words like am, is, are, to, from...etc. and also dropped all kinds of punctuations and stop words. Singular and plural form of a word is considered same. Finally, the remaining frequent words are considered as keywords.

Let an abstract:

Satisfactory durability and stiction results have been obtained with hydrogenated carbon, zirconia, and silica overcoated disks lubricated with a perfluoropolyether with functional end groups. Higher levels of lubricant bonding at room temperature are observed on silica and zirconia surface. It is believed that this is related to the high levels of surface OH for these overcoat. The lubricant affinity for the overcoat is believed to be a primary factor in the durability and stiction performance of the disk. Thus, oxide overcoats should have an inherent advantage over diamond - like carbon because of surface chemistry; however, processing restrictions such as the need for RF sputtering and sensitivity to sputter defects make their use less attractive.

Keywords extracted from this abstract are:
hydrogenated, carbon, zirconia, silica, lubricant, temperature, disk, oxide, chemistry

Originate Associated Word Sets

Each abstract is considered as a transaction in the text data. After pre-processing the text data association rule mining [1][3][9] is applied to the set of transaction data where each frequent word set from each abstract is considered as a single transaction. Using these transactions, we generated a list of maximum length sets applying the Apriori algorithm [1][3][9]. The support and confidence is set to 0.1 and 0.78 respectively. A list of the generated large word set for 65% of training data with their occurrence frequency is illustrated in Table 1.

Table 2 Word sets with Occurrence Frequency

Large Word Set Found	Number of Occurrence in Documents				
	CH	CA	CG	DS	WT
chemistry, area, technology	2				
chemical, chemistry	2				
chemical, chemistry, bond	2				
chemistry, atom, molecule	2				
paper, quantum, chemistry	2				
plasma, nitride, chemistry	2				
chemistry, pigment	2				
quantum, chemistry	2				
environment, chemistry, chemical	2				
environment, chemistry	2				
chemistry, chemical, molecule	2				
chemistry, paper	28				
chemistry, oxide	9				
chemistry, environment, paper	7				
area, technique	3				
quantum, paper	4				
environment, paper, science, laboratory	2				
technology, chemical, chemistry	5				
area, environment, chemistry	3				
organic, chemical, chemistry	4				
gas, plasma, chemistry	5				
computer, architecture, hardware		24			
computer, archi-		2			

tecture, organization, logic, simulator					
computer, architecture, time, processor		6			
computer, architecture, hardware, processor, complex		2			
computer, architecture, hardware, processor, software		2			
computer, architecture, simulator, hardware, core		2			
computer, architecture, system		10			
computer, architecture, cache, memory		3			
computer, architecture, system, hardware		10			
computer, architecture, application, system		9			
computer, architecture, processor, system		16			
computer, architecture, structure, processor		4			
computer, architecture, key, multiprocessor, system		2			
computer, architecture, key, simulation, simulator		2			
computer, architecture, memory, processor		8			
computer, architecture, hardware, system, software		2			
computer, archi-		2			
tecture, system, digital, complex					
computer, architecture, system, digital		3			
vision, computer, image, graphics			4		
computer, graphics, image, color			4		
computer, graphics, design, display, art			2		
computer, graphics, image, art			3		
computer, graphics, image, design			4		
computer, graphics, design			12		
computer, graphics, image			29		
graphics, interactive			12		
computer, graphics, display			14		
computer, graphics, light			4		
computer, graphics, picture			7		
computer, graphics, line			5		
graphics, point, vector			2		
computer, graphics, video, line			2		
data, dynamic				16	
data, database				3	
data, bit, array				2	
Data, algorithm				27	
data, algorithm, query				4	
data, error, algorithm				4	
data, knowledge				7	
data, queue,				2	

ISSN:2319-7900

dynamic					
object, data, program				3	
data, method, tree				2	
data, tree, binary				7	
data, program, effective				2	
data, analysis, complexity, algorithm				3	
algorithm, data, efficient				4	
data, tree, algorithm				6	
data, object, information				2	
server, internet					5
internet, service, user					4
service, semantic					14
semantic, domain					6
semantic, language					7
user, software					7
XML, service					3
server, remote					4
internet, domain					3
survey, software					2
framework, semantic					8
service, semantic, framework					4
service, prototype					3
software, service					6
user, interne, software					6
service, method					4
semantic, internet					4
software, server					7
software, server, Apache					2
source, internet					2

Associated Word Set with Probability Value Using Naive Bayes

To use the Naive Bayes classifier for probability calculation the generated associated sets are required. The calculation of first term of this classifier is based on the fraction of each target class in the training data. From the generated word set after applying association mining on training data we have found the following information:

total number of word set = 89

total number of word set from Chemistry (CH) = 21

total number of word set from Computer Architecture(CA) = 18

total number of word set from Computer Graphics (CG) = 14

total number of word set from Data Structure (DS) = 16

total number of word set from Web Technology (WT) = 20

Prior probability we had for CH, CA, CG, DS and WT are 0.23, 0.20, 0.15, 0.17 and 0.22 respectively. Then the second term is calculated according to the equation (1). The probability values of word set are listed in Table 2.

Table 3 Word set with Probability Value

Large Word Set Found	**Probability**				
	CH	**CA**	**CG**	**DS**	**WT**
chemistry, area, technology	0.027778	0.009259	0.009259	0.009259	0.009259
chemical, chemistry	0.027778	0.009259	0.009259	0.009259	0.009259
chemical, chemistry, bond	0.027778	0.009259	0.009259	0.009259	0.009259
chemistry, atom, molecule	0.027778	0.009259	0.009259	0.009259	0.009259
paper, quantum, chemistry	0.027778	0.009259	0.009259	0.009259	0.009259
plasma, nitride, chemistry	0.027778	0.009259	0.009259	0.009259	0.009259
chemistry, pigment	0.027778	0.009259	0.009259	0.009259	0.009259
quantum,	0.027	0.00	0.00	0.00	0.00

chemistry	778	9259	9259	9259	9259
environment, chemistry, chemical	0.027778	0.009259	0.009259	0.009259	0.009259
environment, chemistry	0.027778	0.009259	0.009259	0.009259	0.009259
chemistry, chemical, molecule	0.027778	0.009259	0.009259	0.009259	0.009259
chemistry, paper	0.268519	0.009259	0.009259	0.009259	0.009259
chemistry, oxide	0.092593	0.009259	0.009259	0.009259	0.009259
chemistry, environment, paper	0.074074	0.009259	0.009259	0.009259	0.009259
area, technique	0.037037	0.009259	0.009259	0.009259	0.009259
quantum, paper	0.046296	0.009259	0.009259	0.009259	0.009259
environment, paper, science, laboratory	0.027778	0.009259	0.009259	0.009259	0.009259
technology, chemical, chemistry	0.055556	0.009259	0.009259	0.009259	0.009259
area, environment, chemistry	0.037037	0.009259	0.009259	0.009259	0.009259
organic, chemical, chemistry	0.046296	0.009259	0.009259	0.009259	0.009259
gas, plasma, chemistry	0.009434	0.235849	0.009434	0.009434	0.009434
computer, architecture, hardware	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, organization, logic, simulator	0.009434	0.066038	0.009434	0.009434	0.009434
computer, architecture, time,	0.009434	0.028302	0.009434	0.009434	0.009434

processor					
computer, architecture, hardware, processor, complex	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, hardware, processor, software	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, simulator, hardware, core	0.009434	0.103774	0.009434	0.009434	0.009434
computer, architecture, system	0.009434	0.037736	0.009434	0.009434	0.009434
computer, architecture, cache, memory	0.009434	0.103774	0.009434	0.009434	0.009434
computer, architecture, system, hardware	0.009434	0.09434	0.009434	0.009434	0.009434
computer, architecture, application, system	0.009434	0.160377	0.009434	0.009434	0.009434
computer, architecture, processor, system	0.009434	0.04717	0.009434	0.009434	0.009434
computer, architecture, structure, processor	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, key, multiprocessor, system	0.009434	0.028302	0.009434	0.009434	0.009434
computer,	0.009	0.08	0.00	0.00	0.00

ISSN:2319-7900

architecture, key, simulation, simulator	434	4906	9434	9434	9434
computer, architecture, memory, processor	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, hardware, system, software	0.009434	0.028302	0.009434	0.009434	0.009434
computer, architecture, system, digital, complex	0.009434	0.037736	0.009434	0.009434	0.009434
computer, architecture, system, digital	0.009804	0.009804	0.04902	0.009804	0.009804
vision, computer, image, graphics	0.009804	0.009804	0.04902	0.009804	0.009804
computer, graphics, image, color	0.009804	0.009804	0.029412	0.009804	0.009804
computer, graphics, design, display, art	0.009804	0.009804	0.039216	0.009804	0.009804
computer, graphics, image, art	0.009804	0.009804	0.04902	0.009804	0.009804
computer, graphics, image, design	0.009804	0.009804	0.127451	0.009804	0.009804
computer, graphics, design	0.009804	0.009804	0.294118	0.009804	0.009804
computer, graphics, image	0.009804	0.009804	0.127451	0.009804	0.009804
graphics, interactive	0.009804	0.009804	0.147059	0.009804	0.009804
computer,	0.009	0.00	0.04	0.00	0.00

graphics, display	804	9804	902	9804	9804
computer, graphics, light	0.009804	0.009804	0.078431	0.009804	0.009804
computer, graphics, picture	0.009804	0.009804	0.058824	0.009804	0.009804
computer, graphics, line	0.009804	0.009804	0.029412	0.009804	0.009804
graphics, point, vector	0.009804	0.009804	0.029412	0.009804	0.009804
computer, graphics, video, line	0.009615	0.009615	0.009615	0.163462	0.009615
data, dynamic	0.009615	0.009615	0.009615	0.038462	0.009615
data, database	0.009615	0.009615	0.009615	0.028846	0.009615
data, bit, array	0.009615	0.009615	0.009615	0.269231	0.009615
Data, algorithm	0.009615	0.009615	0.009615	0.048077	0.009615
data, algorithm, query	0.009615	0.009615	0.009615	0.048077	0.009615
data, error, algorithm	0.009615	0.009615	0.009615	0.076923	0.009615
data, knowledge	0.009615	0.009615	0.009615	0.028846	0.009615
data, queue, dynamic	0.009615	0.009615	0.009615	0.038462	0.009615
object, data, program	0.009615	0.009615	0.009615	0.028846	0.009615
data, method, tree	0.009615	0.009615	0.009615	0.076923	0.009615
data, tree, binary	0.009615	0.009615	0.009615	0.028846	0.009615
data, program, effective	0.009615	0.009615	0.009615	0.038462	0.009615
data, analysis, complexity, algorithm	0.009615	0.009615	0.009615	0.048077	0.009615
algorithm, data, efficient	0.009615	0.009615	0.009615	0.067308	0.009615
data, tree, algorithm	0.009615	0.009615	0.009615	0.028846	0.009615

data, object, information	0.009259	0.009259	0.009259	0.009259	0.055556
server, internet	0.009259	0.009259	0.009259	0.009259	0.046296
internet, service, user	0.009259	0.009259	0.009259	0.009259	0.138889
service, semantic	0.009259	0.009259	0.009259	0.009259	0.064815
semantic, domain	0.009259	0.009259	0.009259	0.009259	0.074074
semantic, language	0.009259	0.009259	0.009259	0.009259	0.074074
user, software	0.009259	0.009259	0.009259	0.009259	0.037037
XML, service	0.009259	0.009259	0.009259	0.009259	0.046296
server, remote	0.009259	0.009259	0.009259	0.009259	0.037037
internet, domain	0.009259	0.009259	0.009259	0.009259	0.027778
survey, software	0.009259	0.009259	0.009259	0.009259	0.083333
framework, semantic	0.009259	0.009259	0.009259	0.009259	0.046296
service, semantic, framework	0.009259	0.009259	0.009259	0.009259	0.037037
service, prototype	0.009259	0.009259	0.009259	0.009259	0.064815
software, service	0.009259	0.009259	0.009259	0.009259	0.064815
user, interne, software	0.009259	0.009259	0.009259	0.009259	0.046296
service, method	0.009259	0.009259	0.009259	0.009259	0.046296
semantic, internet	0.009259	0.009259	0.009259	0.009259	0.074074
software, server	0.009259	0.009259	0.009259	0.009259	0.027778
software, server, Apache	0.009259	0.009259	0.009259	0.009259	0.027778
source, internet	0.009259	0.009259	0.009259	0.009259	0.027778

Here we derive the table about accuracy regarding different test data sets and to see the Table 4.4 the last page of this paper.

Comparative Study

In this section we have tried to represent comparative presentations in different point of views. We studied three thesis papers for the comparison purpose.

Association Rule and Naïve Bayes Classifier

The following results are found using the same data sets for both Association Rule with Naive Bayes Classifier and proposed method. The result shows that proposed approach work well using only 50% Training data.

Table 5 Comparison of Association Rule with Naïve Bayes Classifier and Hybrid Method

% of Training Data	% of Accuracy	
	Association Rule with Naive Bayes Classifier	Hybrid Method
10	40	31
20	17	36
30	42	59
40	60	67
50	32	81

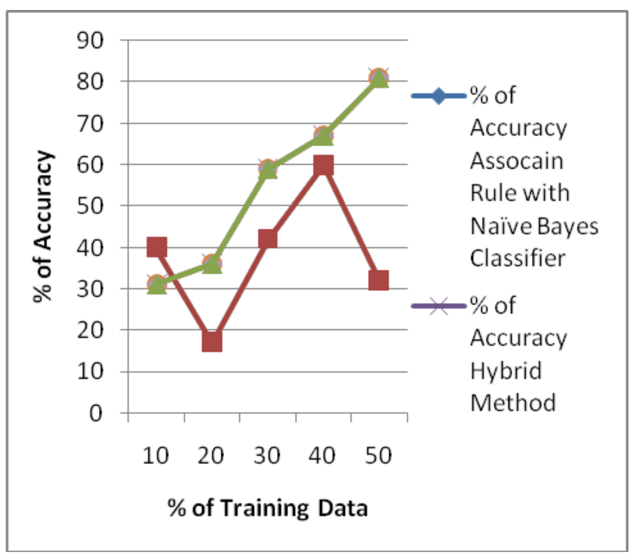

Figure 2 % of Training Data vs % of Accuracy

Limitations and Future Work

ISSN:2319-7900

As we have observed in this method better accuracy is found with increasing confidence up to 0.78. The algorithm will be more effective if the training set is set in such a way that it generates more sets. That is training set with all the different sections of total data can give more dependable result. In this the computational time is too much using Apriori algorithm. Moreover, if Frequent Pattern (FP) Growth tree could be formed time would be shorter enough [7]. Though the experimental results are quite encouraging, it would be better if we work with larger data sets with more classes.

Conclusion

Here, this paper presented an efficient and simple technique for text classification. The existing techniques require more training data sets as well as the computational time of these techniques is also large. In contrast to the existing algorithms, the proposed hybrid algorithm requires less training data and less computational time. Despite the randomly chosen training set we achieved 92% accuracy for 50% training data. Still the experimental results are quite encouraging, it would be better if we work with larger data sets with more classes.

Acknowledgments

The authors are thankful to IJACT Journal for the support to develop this document.

References

[1] S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan “Text Classification Using Data Minig”, *ICTM 2005*.

[2] Qasem A. Al-Radaideh, Eman Al Nagi “Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance”, *International Journal of Advanced Computer Science and Applications(IJACSA)*, Vol. 3, No. 2, 2012

[3] XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg “Top 10 algorithms in Data Mining”, *Knowl Inf Syst (2008)* 14:1–37 DOI 10.1007/s10115-007-0114-2

[4] Thair Nu Phyu “Survey of Classification Techniques in Data Mining”, Proceedings of the *International MultiConference of Engineers and Computer Scientists 2009* Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

[5] Erhard Rahm, Hong Hai Do, “Data Cleaning: Problems and Current Approaches”, *IEEE Data Eng. Bull. 23(4):3--13* (*2000*), *University of Leipzig, Germany* http://dbs.uni-leipzig.de

[6] Chowdhury Mofizur Rahman and Ferdous Ahmed Sohel and Parvez Naushad and S M Kamruzzaman, “ Text Classification using the Concept of Association Rule of Data Mining”, *CoRR* (*2010*) http://arxiv.org/ftp/arxiv/papers/1009/1009.4582.pdf

[7] http://www.iasri.res.in/ebook/win_school_aa/notes/Data_Preprocessing.pdf (Last visited 24 Jan 2013)

[8] Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, Second Edition *Elsevier Inc*.

[9] Agarwal R., Mannila H., Srikant R., Toivonan H., Verkamo, "A Fast Discovery of Association Rules," *Advances in Knowledge Discovery and Data Mining*, 1996.

[10] Bing Liu, Wynne Hsu, Yiming Ma, "Integrating Classification and Association Rule Mining," *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98, Plenary Presentation)*, New York, USA, 1998.

[11] Canasai Kruengkrai , Chuleerat Jaruskulchai, "A Parallel Learning Algorithm for Text Classification," *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, Canada, July 2002.

[12] Kamruzzaman S. M, Farhana Haider, "A Hybrid Learning Algoritm for Text Classification", *Accepted for the Publication of the Proceedings of the Third International Conference on Electrical and Computer Engineering*, Going to be held at Dhaka, on December 28-30, 2004.

[13] Lewis, D., and Ringuette, M., "A Comparison of Two Learning Algorithms for Text Categorization," *In Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 81-93, 1994.

[14] McCallum, A., and Nigam, K., "A Comparison of

ISSN:2319-7900

Events Models for Naïve Bayes Text Classification," *Papers from the AAAI Workshop*, pp. 41-48, 1998.

Biographies

KEYUR J PATEL received the B.E. degree in Information Technology from the University of Hemchndracharya North Gujarat, Patan, Gujarat, in 2011, the M.Tech. persuing in Information Technology from the University of Ganpat, Kherva, Mehsana, Gujarat, His research areas include data mining and its complexity issues.

KETAN J SARVAKAR received B.E. degree and M.Tech degree. He currently works in Ganpat University,Kherva, Mehsana, Gujarat as Assistant Professor. His research areas include data mining and wireless sensor.

TEJAS J PATEL received the B.E. degree in Information Technology from the University of Hemchndracharya North Gujarat, Patan, Gujarat, in 2011, the M.Tech. persuing in Information Technology from the University of Ganpat, Kherva, Mehsana, Gujarat, His research areas include data mining, image processing and its complexity issues.