Rule Generation from Textual Data by using Graph based Approach

D.S Rajput
Department of Computer Applications
M.A.N.I.T., Bhopal-462051

R.S. Thakur
Department of Computer Applications
M.A.N.I.T., Bhopal-462051

G.S. Thakur
Department of Computer Applications
M.A.N.I.T., Bhopal-462051

ABSTRACT

In this study we investigate the significance of textual document which is now commonly recognized by researchers for better management, smart navigation, well-organized filtering, and finding the results. The challenging part is to extract the meaningfulness and to manage the purpose of the “best” Mining Rule .This research study is proposed to refine the Mining Rule from textual data set by performing Graph based approach.

Keywords

Association Rule, pre-processing Technique, Adjacency Matrix, Textual Data.

1. INTRODUCTION

The access to a large amount of textual documents becomes more and more effective due to the growth of the Web, digital libraries, technical documentation, medical data; these textual data represent a resource that has significance use. Text mining is a major research field due to the need of acquiring knowledge from the large number of available text documents, particularly on the Web. In this way knowledge discovery from textual databases, or for short, text mining (TM), is an important and difficult challenge, because of the richness and ambiguity of natural language (used in most of the available documents). Therefore, the problem is the existing huge amount of textual information available in textual form in databases and other online sources. So the question is, who is able to read and analyze it? Nowadays, a lot of database systems are built for storing documents and textual data. Thus, it is necessary to provide automatic tools for analyzing large textual collections. Accordingly, in analogy to data mining to structured data, text mining is defined for textual data [1] In fact; we define text mining to be the science of extracting additional information from hidden patterns in unstructured large textual collection [2]. It is all about extraction of Associations that were previously unknown from large text databases. There have been many algorithms developed for fast mining of frequent patterns in the last decades. [4, 9, 10, 11, 12].

In this research study we used graph based approach [4] which reduces the database scans and avoid candidate generation .In this approach dataset compressed into a directed graph which is stored in the form of lower triangular adjacency matrix.

2. LITERATURE REVIEW

With the explosive growth of the textual data, we face an increasing amount of information resources, of which most are represented in free text. As text data are inherently unstructured and difficult to directly process by computer programs, there has been great interest in text mining techniques for helping users to quickly gain knowledge. All the researchers worked in the area of finding association rules from textual data by using apriori, FP growth, pincer and many other algorithms [12].

R. Feldman at el. in 1996 proposed mining the Associations in Text in the Presence of Background Knowledge [21].This paper has described the FACT system for knowledge discovery in collections of textual documents, which finds associations amongst the keywords labeling the documents given background knowledge about the keywords and relationships between them.

K. Wang at el. in 1999 proposed a new category of text clustering algorithms. They address the special characteristics of text documents and use the concept of frequent word sets for the text clustering. In [26], they proposed a new criterion for clustering transactions using frequent itemsets, instead of using a distance function.

Ch. Cherif Latiri at el. in 2001 Generated Implicit Association Rules from Textual Data [5]. The objective of this paper is twofold. First, to propose a conceptual approach, based on the formal concept analysis and a semantic pruning, in order to discover explicit association rules, from large textual corpus.

B.C.M. Fung at el. in 2003 proposed the Frequent Itemset-based Hierarchical Clustering (FIHC) [27], algorithm in this direction. It measures the cohesiveness of a cluster directly by using frequent word sets, such that the documents in the same cluster are expected to share more frequent word sets than those in different clusters.

Huan at el. in 2004 developed a new algorithm which mines only maximal frequent subgraphs[15] that is subgraph, that are not a part of any other frequent subgraphs. This algorithm can achieve a five-fold speed up over the current state-of-the-art subgraph mining algorithms. This mining method is based on a novel graph mining framework in which they first mine all frequent tree patterns from a graph database and then construct maximal frequent sub graphs from trees.

W.L. Liu at el. in 2005 proposed the documents clustering algorithm on the basis of frequent term sets [29]. Initially, documents were denoted as per the Vector Space Model (VSM) and every term is sorted in accordance with their relative frequency. Then frequent term sets can be mined using frequent-pattern growth (FP growth). Lastly, documents were clustered on the basis of these frequent term sets. This approach was efficient for very large databases, and gave a clear explanation of the determined clusters by their frequent term sets. The efficiency and suitability of the proposed algorithm has been demonstrated with the aid of experimental results. It is an Efficient Approach for Text Clustering Based on Frequent Itemsets.

S. Ghanshyam Thakur at el. in 2007 generated Association Rule from Textual Documents [6]. The objective of this paper is to describe the concept of Binary Matrix Model (BMM) in this research study. The apriori methods were used on this matrix model for Association rule generation.

Hany Mahgoub at el. in 2008 proposed a Text Mining Technique Using Association Rules Extraction [8]. The objective of this paper was to extract more interesting rules. Extracting Association Rules from Text (EART) automatically discovers association rules from textual documents.

Yang at el. in 2010 presented a novel approach to data representation for computing this kernel, particularly targeting sparce matrices representing power-law graphs. They show their representation scheme, coupled with a novel tiling algorithm that can yield significant benefits over the current state of the art GPU and CPU efforts on a number of core data mining algorithms such as Page Rank, HITS and Random Walk with Restart [17].

Graphs became increasingly important in modeling complicated structures, such as circuits, images, chemical compounds, protein structures, biological networks, social networks, the web, workflows, and XML documents. Many graph search algorithms have been developed in chemical informatics, computer vision, video indexing and text retrieval with the increasing demand on the analysis of large amount of structured data; graph mining has become an active and important theme in data mining. Bogdanov [31] in 2008 studied on Graph searching, indexing, mining and modeling for Bioinformatics, chemoinformatics and Social network.

Lam and Chan [32] in 2008 studied on graph data mining algorithm which is increasingly applied to biological graph data set. In this paper they proposed graph mining algorithm MIGDAC (Mining graph data for classification) that applies on graph theory and an interesting measure to discover interesting sub graphs which can be both characterized and easily distinguished from other classes.

A graph transaction is represented by adjacency matrices and the frequent patterns appearing in matrices are mined through the extended algorithm. These are modelled as attribute graph in which each vertex represents an atom and each edge a bond between atoms. Each vertex carries attribute that indicates the atom type.

3. BASIC DEFINITIONS

Definition 3.1: Document Set: - A document set, denoted D= { D_1, D_2, D_3........... D_n}, also called a document collection, is a set of documents, where n is the total number of documents in D.

Definition 3.2: Term Set: - The term set of a document set D= { D_1, D_2, D_3........... D_n} denoted by TD={t_1, t_2,…, t_n}, is the set of terms appeared in D.

Definition 3.3: Directed Graph :- A directed Graph or digraph G consists of a set of vertices V={ v_1,v_2........v_n} ,and a set of edges E={e_1,e_2,---------------e_n},each edge in the graph G is assigned a direction and is identified with an ordered pair (u, v) where u is the initial vertex and v is end vertex.

Definition 3.4: Adjacency Matrix: - let G be a directed graph consist of n vertices .then the adjacency matrix of graph is an n*n matrix A= [a_{ij}] defined by

$$A_{ij} = \left\{ \begin{array}{ll} 1 & \text{if there is an edge between } v_i \text{ and } v_j \text{ and} \\ & \text{if } v_i \text{ is initial vertex and } v_j \text{ is final vertex} \\ 0 & \text{if there is no edge between } v_i \text{ and } v_j \end{array} \right\}$$

Benefits of Matrix Representation of Graph:-

There are two standard way of maintaining a Graph G in the memory of a computer. One way is called the Sequential Representation of G that is by mean of its Adjacency Matrix A. The other way is called type Linked Representation or Adjacency structure of G that uses linked lists of neighbor.

Matrix Representation of Graph to computer could be viewed as a very convenient and useful way, because the Graph and Matrix have their own important role in basic science, Network Analysis and other Research Problems. The main advantages of the Matrix Representation of the graph are:

1 Matrix can be easily stored and manipulated in computer.

2 Simple basic knowledge of the operation of matrix algebra waded to evaluate the characteristics of Graph.

3 Matrix Representation of a Graph depends y upon the Order of Vertices.

4. PROPOSED METHOD

In this paper we combine two approach binary matrix and graph based techniques. We are generating rule from textual data by using graph based technique. The proposed method for extracting Rules from Text consists of three phases which is shown by figure: 1.

1. Text Preprocessing phase (transformation, filtration, stemming and indexing of the documents)
2. Rule Generation Mining (RGM) (using graph based approach)
3. Result phase (visualization of results).

Suppose we have data set D= (D_1, D_2, D_3, D_4, D_5, D_6, D_7).

Phase 1: Text Pre-processing

The most important procedure in the pre-processing of documents is to convert the word forms into meaning combination. The goal of text preprocessing phase is to optimize the performance of the next phase: i.e. Association Rule Mining. Each document set have numerous stop words, special marks, punctuation marks and spaces. This process includes various sub processes like stop word elimination, stemming, feature selection etc.

a) Stop words Elimination:-

First we remove all stop words. Stop words are the words which don't have meaning with respect to the classification. So these words are removed when the term matrix is created for the classification purpose. In short the words are removed from the documents which are not necessary for the next stage. Stop words are 'a', 'an', 'the', 'was', 'were' etc. [6, 12, 13, 20, 30], along with all removed prepositions, conjunction and articles from the data set D.

b) Stemming:-

After stop words elimination, the stemming process will be applied. The stemming process is elimination of prefixes and suffixes, [6, 12, 14, 28, 30]. The objective is to remove the variation that arises from the amount of different grammatical forms of the similar word. The stemming process helps to decrease the size of the data dictionary file.

c) Feature Term Selection:-

In text classification applications, selection is a critical task for the classifier performance. With increasing number of documents, the number of features also increases. To reduce the size of the dictionary, the threshold term selection method is used. In this method, the upper and lower thresholds are decided according to the number of words in the dictionary [23, 24]. After that the term which exceeds the upper threshold and the terms below lower threshold are extracted from the document. This helps to reduce the size of the dictionary.

The weighting scheme TF-IDF (Term Frequency, Inverse Document Frequency)[23, 24] is used to assign higher weights to distinguish terms in a document, and it is the most widely used weighting scheme which is defined as [23, 24, 25].

Once text pre-processing is applied over the document, it will be converted into form of binary matrix. To convert all documents in the form of binary matrix we have used BMM Model [6, 18, 19].

Binary matrix model [BMM]

After selection process we have limited terms in each document. Suppose we have n documents and maximum m steem words in a document. The binary matrix M is represented as [6, 18, 19] .

$$M\,[d_i * w_j] = \left\{\begin{array}{ll} 1 & \text{if wi in present in di} \\ 0 & \text{otherwise} \end{array}\right\}$$

Where i=1,2,3..............n
j= 1,2,3..........m
and di = document list
wj = word list

In binary matrix model each row represents a vector. This means that each document can be represented as a vector. In given model document D_1-> [1, 1, 0, 0, 1, 1, 1].The result of first phase is shown in Table. 1.

Phase 2: Rule Generation Mining (RGM)

In this phase association rule mining will be done using graph based algorithm [4] and generate various frequent k-itemsets. The Binary Term Matrix will be used as input dataset in this phase.

Algorithm for Graph based approach

Input: The set of different Textual document D.

Output: Frequent Patterns.

1. Scan document table D and create directed graph G.
2. Create Adjacency Matrix A according to definition 4.
3. Update Value of each element A_{ij}.list and A_{ij}.count of matrix A.
4. Delete corresponding row and column of a element A_{ij}.count=0 only for diagonal elements.
5. Read each element A_{ij} of matrix A if A_{ij}.count<minimum Support then set A_{ij}.Count =0
6. Find 1- Frequent itemset and 2- frequent itemset from matrix.
7. Calculate other K-itemsets from each column using logical AND operator.
8. END

Implementation of algorithm:-

The first phase performed text preprocessing on textual dataset D= (D_1, D_2, D_3, D_4, D_5, D_6, D_7) and generate binary matrix table, which is shown in Table.1.

Document id	1	2	3	4	5	6	7
D_1	1	1	0	0	1	1	1
D_2	0	1	0	1	0	0	1
D_3	0	0	0	1	1	0	0
D_4	0	1	1	0	0	1	1
D_5	0	0	0	0	1	1	0
D_6	0	1	1	1	0	0	1
D_7	1	1	0	0	0	1	1

Table 1: Binary Matrix M

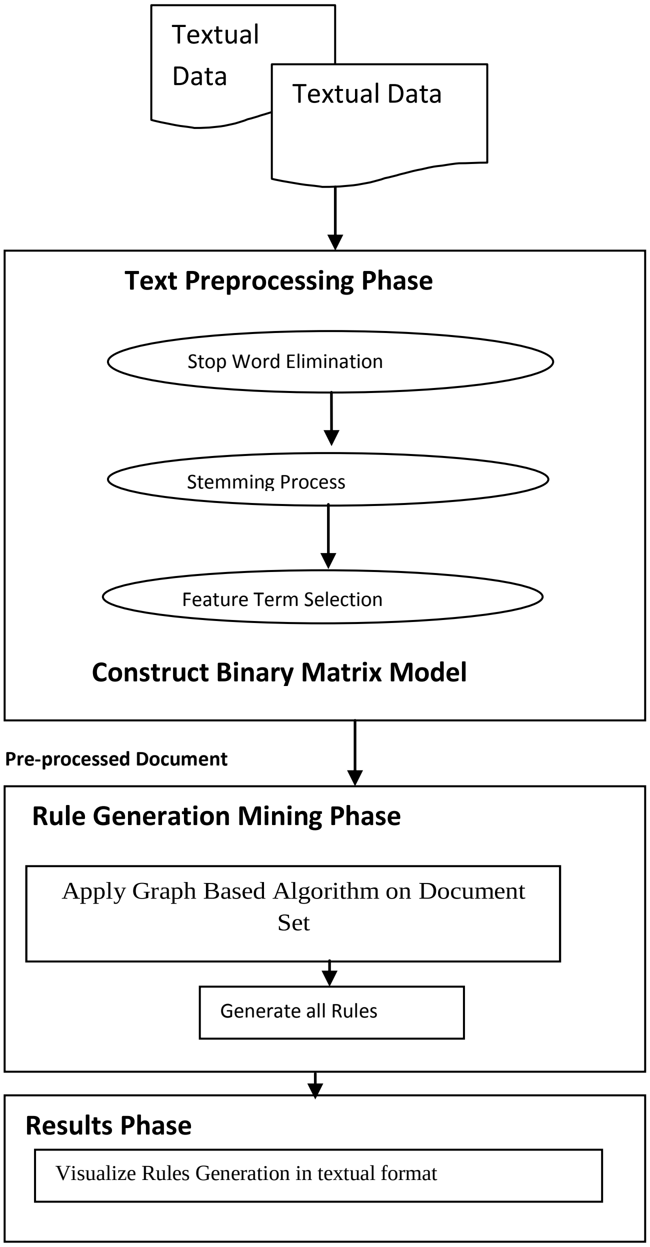

Figure 1: Text Mining System Architecture

To simplify the working of graph based algorithm we have given unique word to each term in the document. For example.

1=Compiler, 2=Interpreter, 3=Sound, 4=Picture, 5=Marker, 6= Assembler, 7= Program etc.

In the first phase of this architecture, scan the binary matrix dataset and construct document matrix table.2.

Document id	Word Sequence
D_1	1,2,5,6,7
D_2	2,4,7
D_3	4,5
D_4	2,3,6,7
D_5	5,6
D_6	2,3,4,7
D_7	1,2,6,7

Table 2: Document Matrix for BMM

In first step, the algorithm will scan the document matrix D and create the directed graph G, which is shown in figure.2

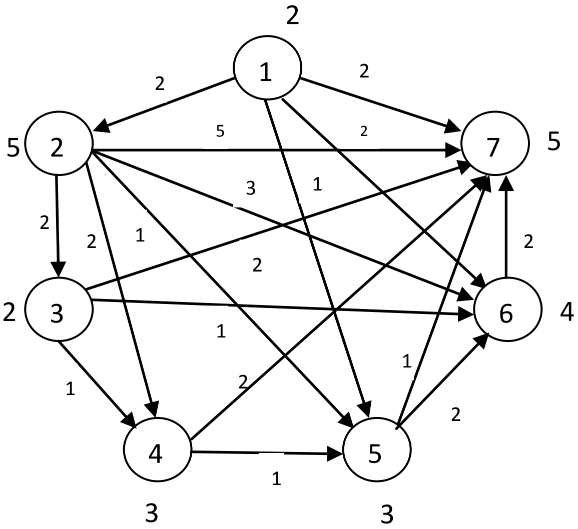

Figure 2: Graph Representation of Document Matrix

This graph G will be stored in matrix into the form of adjacency matrix which is shown in figure.3 because in this paper we have used modified definition of adjacency matrix for a graph with parallel edges, which is defined as in [4].

Suppose A is symmetric matrix where

$$A_{ij}= \begin{cases} m, & \text{and m>0 if there are m directed edges Between vertices i to j} \\ 0 & \text{if there is no edge between them} \end{cases}$$

$$\begin{array}{c|ccccccc}
1 & A_{11} & & & & & & \\
2 & A_{12} & A_{22} & & & & & \\
3 & A_{13} & A_{23} & A_{33} & & & & \\
4 & A_{14} & A_{24} & A_{34} & A_{44} & & & \\
5 & A_{15} & A_{25} & A_{35} & A_{45} & A_{55} & & \\
6 & A_{16} & A_{26} & A_{36} & A_{46} & A_{56} & A_{66} & \\
7 & A_{17} & A_{27} & A_{37} & A_{47} & A_{57} & A_{67} & A_{77} \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}$$

Figure 3: Lower Triangular Adjacency Matrix of Graph

Each element A_{ij} of matrix A has two fields; one is list and second is count. The list fields contain Document id of corresponding items {1,2,3,4,5,6,7} of the matrix element A_{ij} and count field stored an integer value, which is equivalent to number of Document id in list field. Each matrix element contains the following information:

A_{11}.list =[D_1,D_7] and A_{11}.Count =2 , contain Document id in which {1} occurred.

A_{12}.list =[D_1,D_7] and A_{12}.Count= 2 , contain Document id in which {1,2} occurred.

A_{13}.list = [ϕ] and A_{13}. Count=0 then there is no path between the vertices in Graph.

A_{14}.list= [ϕ] and A_{14}. Count=0 then there is no path between the vertices in Graph.

A_{15}.list = [D_1] and A_{15} .Count = 1, contain Document id in which {1, 5} occurred.

A_{16}.list = [D_1, D_7] and A_{16}.Count =2, contain Document id in which {1, 6} occurred.

A_{17}.list = [D_1, D_7] and A_{17}.count =2, contain Document id in which {1, 7} occurred.

Similarly, we can show the rest of the elements of the matrix. In further step, we check count value of each element of the matrix A, if any diagonal element A_{ij}.count<minimum support(Suppose Minimum support =2) then, delete row and column of corresponding element from the matrix, because the superset of any infrequent item set will never be frequent, and other than diagonal elements (for which i≠j). If A_{ij}.count< minimum support then Aij is to assign zero value in the matrix. For example A_{13}=0 because A_{13}.count 0< minimum support, similarly do for others. As shown in figure 4.

$$\begin{array}{c|ccccccc}
1 & A_{11} & & & & & & \\
2 & A_{12} & A_{22} & & & & & \\
3 & 0 & A_{23} & A_{33} & & & & \\
4 & 0 & A_{24} & 0 & A_{44} & & & \\
5 & 0 & 0 & 0 & 0 & A_{55} & & \\
6 & A_{16} & A_{26} & 0 & 0 & A_{56} & A_{66} & \\
7 & A_{17} & A_{27} & A_{37} & A_{47} & 0 & A_{67} & A_{77} \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}$$

Figure 4: Filtered Adjacency Matrix of Graph G

Now filtering adjacency matrix we will find all k- itemset. All diagonal elements of matrix shows the frequent 1- itemset. Such as

A_{22}=2, A_{33}=3, A_{44}=4, A_{55}=5, A_{66}=6, A_{77}=7

Other element of matrix shows the frequent 2-itemset those where A_{ij}.count≠0.

A_{12}= {1, 2} A_{16}= {1, 6} A_{17}= {1, 7}

A_{23}= {2, 3} A_{24}= {2, 4} A_{26}= {2, 6} A_{27}= {2, 7}

A_{37}= {3, 7} A_{47}= {4, 7} A_{56}= {5, 6} A_{67}= {6, 7}

For extracting frequent k- itemsets we apply AND operation in between each column elements of matrix.

Frequent 3-itemset is calculated as

A_{12}.list AND A_{16}.list= {1, 2, 6} = [D_1, D_7]

A_{12}.list AND A_{17}.list= {1, 2, 7} = [D_1, D_7]

A_{16}.list AND A_{17}.list= {1, 6, 7} = [D_1, D_7]

A_{23}.list AND A_{24}.list= {2, 3, 4} = [D_6] = 1 is less then minimum support thus it is not frequent.

A_{23}.list AND A_{26}.list= {2, 3, 6} = [D_4] = 1 is less then minimum support thus it is not frequent.

A_{23}.list AND A_{27}.list= {2, 3, 7} = [D_4, D_6]

A_{24}.list AND A_{26}.list= {2, 4, 6} = [ϕ] =0 is less then minimum support thus it is not frequent.

A_{24}.list AND A_{27}.list= {2, 4, 7} = [D_2, D_6]

A_{26}.list AND A_{27}.list= {2, 6, 7} = [D_1, D_4, D_7]

Frequent 4 -item set calculate from first Column

A_{12}.list AND A_{16}.list AND A_{17}.list = {1, 2, 6, 7} = [D_1, D_7]

Similarly calculate from second column

A_{23}.list AND A_{24}.list AND A_{26}.list= {2, 3, 4, 6} = [ϕ]

A_{23}.list AND A_{24}.list AND A_{27}.list= {2, 3, 4, 7} = [D_6] = 1 is less then minimum support thus it is not frequent.

A_{24}.list AND A_{26}.list AND A_{27}.list= {2, 4, 6, 7} = [ϕ] is less then minimum support thus it is not frequent.

Phase 3: Results phase (Visualize the Results)

In This phase extracted association rules can be viewed in textual format or in graphical format. In this phase, the system is designed to visualize the extracted association rules in textual format or tables.

For rule generation we can consider one of the frequent itemsets{1,2,6,7} which is frequent in document D_1 and D_7.The considered itemsets {1,2,6,7} can also be presented in its original form like(Compiler, interpreter, assembler, program). Where 1 represents to compiler, 2 represent to interpreter, 6 represents to assembler and 7 represents to program. Now from this frequent itemset we can generate possible association rule like:

Compiler & Interpreter & Assembler→ Program

Interpreter→ Program

Program→Assembler & Interpreter

Program→Compiler

Program→ Assembler etc.

Similarly, we can generate the maximum rule from all frequent itemsets. It is noticed that the extracted rules will reflect the most important features and informative news of the domain in the document collection. Some of the text clustering algorithms use frequent word sets to compare the distance between documents.

5. EXPERIMENTAL RESULT

To explore the behavior of proposed approach, we used various selected sample of 200 web pages, collect news that is related to computer. There are many sources to news such as Reuters, Computer News today, Email..etc. There are multiple features in the document and they are scattered widely in the text such as “Compiler”, “Interpreter”, “Program”, “Sound”, “Assembler”, “Image”, “Picture” etc. The aim of this work is to find relation between the features and represents them in the form of association rules which will be useful to the end users or people to get correct information about the computer.

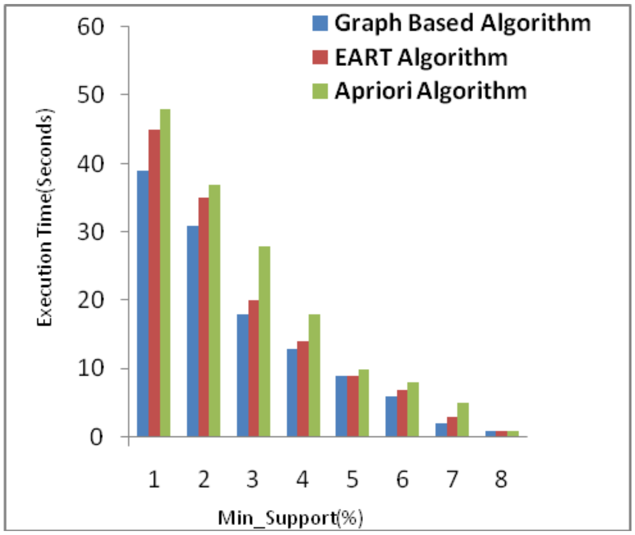

Figure 5: Runtime Comparison among Apriori, EART and Graph Based algorithm

Figure 5 shows the performance of graph based algorithm with Apriori and EART using news_20 dataset and generated various size rules. Finally, we have observed that the difference in execution time among the three systems is in Seconds. However, this difference will increase and goes in minutes especially when documents are available in large amount. In this case Apriori based system and EART system will runs in minutes.

The experiments were performed on an Intel core 2 Duo, 2.94 GHz system running Windows 7 professional with 2 GB of RAM.

6. CONCLUSION AND FUTURE WORK

In this Paper we propose new method which stands for rule generation from collection of documents based on the keyword features by using graph based technique. Frequent word discovered from the document set can represent the topic covered by the documents very well and it measures the closeness between the documents. The proposed system use the concept features to represent text and to extract the more useful association rules that have more meaning. This is a very efficient method because the advantage of graph based approach is; the matrix of Graph maintenance and manipulation is easy in computer. There is a need of only basic knowledge of matrix algebra. It reduces the CPU time because we can scan only two times the document dataset and we don’t need to generate candidate set. Due to increasing size and complexity of Textual data in computer science there is a need for efficient graph rule generation algorithm. Future possibilities of this approach, we will use this for text document clustering and to get working ability with specific sequences of word.

7. ACKNOWLEDGMENT

This work is supported by research grant from MANIT, Bhopal, India under Grants in Aid Scheme 2010-11, No. Dean(R&C)/2010/63 dated 31/08/2010.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between sets of items in large Databases. ACM SIGMOD Records, 1993 pp 207-216.

[2] L. Singh, B. Chen, R. Haight, and P. Scheuermann. An algorithm for constrained association rule: Mining in semistructured data. In Proceedings of the third Pacific-Asia Conference, PAKDD '99, Beijing, China, 1999.

[3] R. Agrawal and R. Skirant. “Fast algorithms for mining association rules”. In Proceedings of the 20th Int 'I Conference on Very Large Databases, June 1994 pp. 478-499.

[4] R.S. Thakur, R. C. Jain, K.R. Pardasani “Graph Theoretic Based Alogorihtm for mining frequent Pattern” International Joint Conference on Neural Networks (IJCNN 2008),pp 628-632.

[5] Ch. Cherif Latiri, S. BenYahia “Generating Implicit Association Rules from Textual Data” IEEE, 2001 pp 137-143.

[6] S. Ghanshyam Thakur, Rekha Thakur and R.C. Jain, “Association Rule Generation from Textual Document” International Journal of Soft Computing, 2: 2007 pp. 346-348.

[7] B. Ganter and R. Wille. Formal Concept Analysis. Springer-Verlag, Heidelberg, 1999.

[8] Hany Mahgoub, Dietmar Rösner, Nabil Ismail and Fawzy Torkey, “Text Mining Technique Using Association Rules Extraction” International Journal of Information and Mathematical Sciences, 2008 pp. 21-28.

[9] J.Hen ,J. Pei, and Y. Yin,“ Mining Frequent patters without candidate generation,” Prod. SIGMOD 2002.

[10] J.Pei, J. Han, H. LU,S. Nishio, S. Tang and D. Yang,” H-Mine: Hyper-structure Mining of frequent Patterns in large database,” in Proc. The IEEE international conference on data mining, ,2001 pp. 441-448.

[11] J.Pei, J. Han and Lakshmanan “ Mining frequent itemsets with Convertible Constraints”, in ICDE 2001.

[12] Han I and Kamber M, “Data Mining concepts and Techniques,”Morgar Kaufmann Publishers,2000, pp.335–389.

[13] Zhang Yuhang,Wang Yue, Yang Wei, “Research on Data Cleaning in Text Clustering” International Forum on Information Technology and Applications 2010.

[14] N. Manerikar, T. Palpanas, Frequent items in streaming data: an experimental evaluation of the state-of-the-art, Data and Knowledge Engineering 68 (4), 2009 pp. 415–430.

[15] Vijender Singh, Deepak Garg, “Survey of Finding Frequent Patterns in Graph Mining: Algorithms and Techniques” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-3, July 2011.

[16] J. Huan, W. Wang, J. Prins and J. Yang,”Spin: mining maximal frequent subgraphs from graph Databases”, KDD04 Seattle,Washington, USA, 2004.

[17] Yang, Parthasarthy and Sadayappan, “Fast Mining Algorithms of Graph data on GPUs.” ACM, KDD-LDMTA’10, 2010.

[18] Tao Li, “A General Model for Clustering Binary Data” KDD’05, August 21–24, 2005, Chicago, Illinois, USA pp. 188-197.

[19] Mickey, M. R., Mundle, P., & Engelman, L. (1988). Boolean factor analysis. In Bmdp statistical software manual, vol. 2, University of California Press , pp. 789–800..

[20] Tin Kam Ho, “Stop word location and identification for adoptive text reorganization.”Brisbane, Australia, Augest17-20,1998, pp. 605-609.

[21] R. Feldman and H. Hirsh, “Mining Associations in Text in the Presence of Background Knowledge,” Knowledge Discovery and Data Mining, 1996 pp. 343-346, http://citeseer.ist.psu.edu/feldman96mining.html.

[22] J.D. Holt and S.M. Chung, “Multipass Algorithms for Mining Association Rules in Text Databases,” Knowledge Information System, vol. 3, no. 2, 2001 pp. 168-183.

[23] C. Manning and H Schütze, Foundations of statistical natural language processing (MIT Press, Cambridge, MA, 1999).

[24] J. Paralic and P. Bednar, “Text mining for documents annotation and ontology support (A book chapter in: "intelligent systems at service of Mankind,” ISBN 3-935798-25-3, Ubooks, Germany, 2003).

[25] M. Rajman and R. Besancon, “Text mining: natural language techniques and text mining applications”, in Proc. 7th working conf. on database semantics (DS-7), Chapan &Hall IFIP Proc. Series. Leysin, Switzerland Oct. 1997, 7-10.

[26] K. Wang, C. Xu, B. Liu, Clustering transactions using large items, in: Proceedings of the 8th International Conference on Information and Knowledge Management, 1999, pp. 483–490.

[27] B.C.M. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: Proceedings of SIAM International Conference on Data Mining, 2003.

[28] E. Ukkonen, On-line construction of suffix trees, Algorithmica 14 (1994), pp. 249–260.

[29] W.-L. Liu and X.-S. Zheng, "Documents Clustering based on Frequent Term Sets", Intelligent Systems and Control, 2005.

[30] Zhitong Su ,Wei Song ,Manshan Lin ,Jinhong Li, "Web Text Clustering for Personalized Elearning Based on Maximal Frequent Itemsets", Proceedings of the 2008 International Conference on Computer Science and Software Engineering ,2008, Vol: 06, Pages: 452-455.

[31] A.M. Fahim, G. Saake, A.M. Salem, F. A. Torkey, M.A. Ramadan,“ K-mens for spherical clusters with large variance in sizes.” World Academy of science, Engineering & Tech., 45, 2008, pp.177-182.

[32] Winnie W.M. Lam, Keith C. C. Chan, “Analyzing Web Layout Structures using Graph Mining”, Granular Computing, 2008. GrC 2008. IEEE International Conference 2008 pp.361-366.

9. AUTHORS BIOGRAPHY

Dharmendra Singh Rajput received his B.Sc. in Computer Science from BU Bhopal in 2005. After then M.C.A. from SATI Vidisha in 2008. and then joined the Faculty of Department of Computer Applications, UIT-RGPV Bhopal on Nov. 2008 to July 2010. Then joined MANIT, Bhopal from July 2010 to Dec. 2011. Currently, he is a Pursuing Ph.D. (Research fellow) of the Department of Computer Applications in MANIT, Bhopal India. His research interests in Text mining, data mining, Information Retrieval. He is a member of IAENG, IACSIT, and APCBEES.

Dr. Ramjeevan Singh Thakur received his B.Sc. from Sagar University in 1995. After then M.C.A. from SATI Vidisha in 1999. And then Ph.D. degree From RGPV, Bhopal in 2008 in Computer Science and Applications. He joined the faculty of the Department of Computer Applications, RGPV University, Bhopal on July 2000 to July 2007, after then joined NIT, Trichy from July 2007 to june 2010.Currently, he is a Associate Professor of the Department of Computer Application, Maulana Azad National Institute of technology, Bhopal, India. His research interests include data/document warehousing, and data/text mining. He is a member of the CSI, IAENG, and IACSIT.

Dr. Ghanshyam Singh Thakur received his B.Sc. from Sagar University in 2000. After then M.C.A. from RaviShankar Shukal University in 2003. And then Ph.D. degree From BU, Bhopal in 2009 in Computer Science. He joined the faculty of the Department of Computer Applications, SATI Vidisha on July 2004 to Jan. 2008, after then joined Polytechnic College, Bhopal from Jan. 2008 to May 2010.Currently, he is a Asistant Professor of the Department of Computer Applications, Maulana Azad National Institute of technology, Bhopal, India. His research interests include Text Mining, Document clustering, Information Retrieval data/document warehousing. He is a member of the CSI, IAENG, and IACSIT.