International Journal of Scientific & Engineering Research Volume 4, Issue 6, June 2013
ISSN 2229-5518

An Automated Approach to Categorize the Web Documents through Text Mining

Sajjan Kumar, Priyanka Nervariya, Dr. Deepak Singh Tomar
Department of Computer Science & Engineering
Maulana Azad National Institute of Technology
Bhopal, India
Emails: sajjan.nitb@gmail.com, priyankanervariya07@gmail.com, deepaksingh@manit.ac.in

Abstract - With the increased access of the internet, it has become obvious for all small and big organizations to have an efficacious web presence to acquaint users with the identity of the enterprise. Now a day's daily routine work of large organization such as communication, document distribution, tender declaration such as notices circular etc is done via websites. Web pages of a website are divided into groups on the basis of similarity among documents in that section of website or user interest. One of the common tasks performed by the web manager is to upload documents in different groups on a website. Large number of documents are available on a website domain, thus it's a difficult task for web manager to decide manually the group of an upcoming document for uploading. In this paper, an approach is developed to automate the process of deciding the group of an upcoming document on a website.

Index Terms: Data clustering, Data Preprocessing, Maximal Cliques, Maximum clique, Overlapped Cluster, Text mining, Weighted Clique

1 Introduction

Text mining is an automated technique of extracting the hidden or previously unknown information from a large amount of heterogeneous unstructured textual resources [1]. With the rapid increase of the data generated by economic, academic and social activities, text mining provides us the potential to support innovation and development of new knowledge to make sense of these vast information available [2]. Text mining is having wide application in the field of Marketing, industries etc. It deals with large textual databases which are high dimensional in nature i.e. where each word and phrase has a dimension moreover the input to text mining is generally from multiple nodes, whether its user logs, system generated files or web logs etc.

Document clustering is considered as subset of data clustering .It borrows its concept from the fields of information retrieval (IR), natural language processing (NLP), and machine learning (ML). Clustering of Documents involves the use of descriptors and descriptor extraction.[3] Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. Examples of document clustering include web document clustering for search users. Clustering of Documents on a web site can assignment to a group by using document clustering approach to make web pages cluster.

2 Motivation

With the increased access of the internet, it has become obvious for all small and big organizations to have an efficacious web presence to acquaint users with the identity of the enterprise. Moreover organizations also carry out their daily routine work such as communication, document distribution, and tender declaration such as notices circular etc is done via websites. Web pages of a website are divided into groups on the basis of similarity among documents in that section of website or user interest. One of the common tasks which a web manager performs on regular basis is the uploading of documents on a website. Manually deciding the target group to which a particular document belongs is a tedious and a hectic process. The approach developed in this paper is to automate document assignment to a group by using document clustering approach to make web pages cluster.

3 Background

Document assignment is heavily used in information retrieval systems to enhance their performance. Several researchers conducted their research work in these field and proposed different clustering methods for browsing documents or organizing the retrieved results for easy viewing [4]. Agglomerative clustering proposed by [5] starts with all the documents as a separate cluster. At each step, the two most similar clusters are merged and this can be repeated until the desired number of clusters is obtained. But this method does not consider special properties of individual clusters thus it suffers from wrong merging decisions when noise is present.

Dragomir R. Radev. [7] Discussed algorithm for information fusion, which merges similar sentences across documents to create new sentences based on language generation technologies. Portability is mere limitation of this approach.

Several works has been done in partitioning graph into sub graphs. In this paper graphical approach is followed. Alba, Richard D in 1973 [8] discussed about cliques. Several approaches have been done for the fast execution of cliques. L. Babel [9] depicts the fast approach to solve Maximum weight clique problem.

4 PROPOSED ARCHITECTURE

Applying document assignment procedure on Web pages involves preprocessing of the web pages to make their format compatible to the proposed algorithm. For this the web pages has to undergo different modules. Preprocessor module process the document and make it compatible for further text mining process by applying several operations on document such as cleaning, stemming, filtering etc. Clustering is the main part of the architecture where Document clustering algorithm is implemented. Here Inverse document term frequency matrix, similarity matrix along with sparse matrix is calculated. Document is further assigned to their respective clusters.

Fig1.Depicts the proposed architecture and overview of Document Assignment system based on text mining. For this work, our institute website www.manit.ac.in is explored. At first instance following group of web pages are assumed.

Here G1→ Student Group
G2→ Faculty / Staff Group
G3→ Circulars for Account Section
G4→Others

Brief discussion about the different modules of the proposed architecture is elaborated below:-

4.1 Preprocessing Module

Documents provided as input to the clustering module are not consistent in nature. They may contain out of range values, missing values, impossible data combinations etc. [11] Irrelevant or missing values can lead to poor results moreover output will be highly garbled. Thus preprocessing makes the data compatible for further algorithm. Preprocessing involves Filtering, cleaning, Stemming and Pruning [12].

Filtering of data aims to filter out data that is not relevant for processing it involves removal of stop words such as is ,an ,the etc. **Data cleaning** refers to the process of detecting and correcting inaccurate data. It removes inconsistent data and makes the data consistent [13]. **Stemming** of data conflates the data to its stem or root. [14]. Ex- Abate, Abated, abatement, abates reduced to "abat". Many approaches for stemming involves Brute force look up , Suffix – Affix stripping, Part of speech recognition, Porter Stemmer etc. Porter Stemmer utilizes suffix stripping approach to stem data [15]. **Pruning** of data involves removal of all those words which are irrelevant for document clustering. For Pruning we calculate term frequency [16] of each word in a document, and remove all those words whose occurrence is less than the decided threshold. Output of Pruning is feed as input to the document clustering module. After completion of

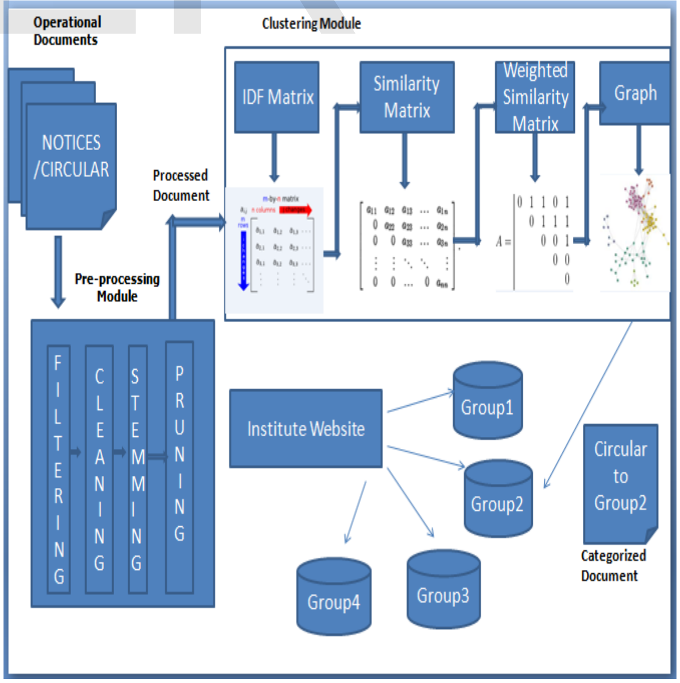


Figure 1 Showing Proposed Architecture for document assignment to website section through text mining.

Pre-processing phase a suitable format is used to represent the document as follows.

4.1.a Document Representation/Document Transformation

In this work documents are transformed through the vector-space model. In this model, each document, d, is considered to be a vector, d, in the term-space (set of document "words"). In its simplest form, each document is represented by the (TF) vector.

Document 1 = term1 $\hat{a}_1$ + term2 $\hat{a}_2$ + ………..+ term n $\hat{a}n$ (1)

Where $\hat{a}1.....\hat{a}n$ are unit vector M Dimension.

Where Term Frequency is, say D = {d1, . . . , dn} be a set of documents and T = {t1, . . . ,tm} the set of distinct terms occurring in D. Let tf(d, t) denote the frequency of term t in document D[17]. Then the vector representation of a document d is given as

Td = (tf(d,t1),………tf(d.tm)) (2)

The vector representation of the document is the basis for the following cluster module.

4.2 Clustering Module

The input to this module is the transformed document. Clustering module comprises several sub modules within it.. Different sub modules of clustering module are explained below:-

4.2.a Inverse Document Term Frequency Matrix

Terms that appears frequently in a small number of documents but rarely in the other documents tend to be more relevant and specific for that particular group of documents and therefore more useful for finding similar documents [17]. In order to capture these terms and reflect their importance, in this work transformation of the basic term frequencies tf(d, t) into the Tf-idf (term frequency and inversed document frequency) weighting scheme. Tf-idf weighs the frequency of a term t in a document d with a factor that discounts its importance with its appearances in the whole document collection, which is defined as:

Tfidf(d,t) = tf(d,t) × log (|D| / df(t)) (3)

Inverse Document term frequency matrix is N × M dimensional matrix i.e.
IDF = $[\ A\ ]_{N\times M}$ where
N= Number of Document.
M= words whose frequency need to be counted.
An element in Aij is defined as the Number of times that term occurs in the respective document.

4.2.b Similarity Matrix

To calculate the proximity between two documents here concept of similarity matrix is explored. Similarity matrix[18] is a square matrix i.e. N × N where N represents the number of documents among which similarity is to be measured. It can be an upper triangular matrix or lower triangular matrix, it is because the value Aij == Aji , where A is Similarity Matrix and i and j represents rows and columns respectively. Similarity Matrix comprises the similarity value among documents.

Similarity measure is the metric used to measure the degree of closeness or separation among the target documents. It corresponds to the characteristics that are believed to distinguish the clusters embedded in the data. Quality of the cluster formed also depends on the similarity measure. Thus the similarity measure map the distance of the two objects into a single numerical value, which depends on two factors that are the qualities of the data object and the similarity measure. Some of the common methods to calculate similarity among two documents are:-

Cosine Similarity

The cosine similarity computes the similarity by multiplying the Document Vectors Values and Finding their Sum and then dividing it by their Unit Vector values. Given two documents Ta and Tb, their cosine similarity is given by

$$SIMC(\overrightarrow{\mathrm{T_a}},\overrightarrow{\mathrm{T_b}}) = \frac{\overrightarrow{\mathrm{T_a}}.\overrightarrow{\mathrm{T_b}}}{|\overrightarrow{\mathrm{T_a}}| \times |\overrightarrow{\mathrm{T_a}}|} \qquad (4)$$

Given two documents as vectors, the cosine angle between the vectors gives the cosine similarity. If Ta and Tb, both are same, the similarity value tends to 1.

Jaccard Coefficient

Jaccards coefficient measure is a metric to measure the similarity between data sets. It is calculated as the intersection divided by the union of the objects [19].

In text document, this similarity metric coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but are not the shared terms.

$$SIMJ(\overrightarrow{\mathrm{T_a}}, \overrightarrow{\mathrm{T_b}}) = \frac{\overrightarrow{\mathrm{T_a}}.\overrightarrow{\mathrm{T_b}}}{|\overrightarrow{\mathrm{T_a}}|^2 + |\overrightarrow{\mathrm{T_b}}|^2 - \overrightarrow{\mathrm{T_a}}.\overrightarrow{\mathrm{T_b}}} \qquad (5)$$

The range of the coefficient is from [0 , 1]. Value is 1, when Ta = Tb and 0 when Ta and Tb , both are Disjoint. 0 shows both are dissimilar and 1 terms them as similar.

4.2.c Graph Formation

To visualize the similarity among documents in this work graph is formed. Adhering to the Indicator function F(x), defined as:

$$F(x) \ = \begin{cases} 1 & X >= Th \\ 0 & X < \ Th \end{cases} \qquad (6)$$

The above Indicator function follows discrete function. As a result, the values above threshold value are only set to 1. This method hinders the similarity value between the vertexes (object) which are either nearer to threshold value or much more than threshold.

Hence, in this paper a WEIGHTED SIMILARITY GRAPH is drawn, that clearly captures the strong or weak relationship between the vertices or objects. Hence the new equation (7), is formalized to represent the Step Similarity Function.

$$F(x) \ = \begin{cases} x & X >= Th \\ 0 & X < \ Th \end{cases} \qquad (7)$$

The function shows that two nodes are connected by an edge if and only if the similarity value is greater than the threshold value. Here the threshold value is defined as the mean of the values in Jaccards similarity Matrix and similarly for cosine Similarity matrix. This approach is continued further to draw the connected graph which results to the Clique formation.

The above mechanism results to Weighted Graph. The main feature of this is to find the set of all maximal set of vertices that sums to a maximum value. This concept will provide the strong relationship between document clustering through fuzzy**.**

4.2.d Clustering Algorithm

One of the common used approaches for clustering is Clique solving in which set of vertices that are densely connected are grouped together. There are two concepts in clique, maximal cliques and maximum clique. Maximal cliques are the set of vertices which are not a subset of the vertices of larger clique. Maximum cliques are the largest set of all the vertices of cliques in a graph.

The approach followed here is to find the maximal cliques in the graph. This work is enhanced by Carraghan and Pardalos[21] by finding the maximum clique of the Graph. In their approach the maximal clique is discovered by repeatedly and simultaneously removing the vertices from the universal set of vertices.

Higher the similarity value strong is the connectivity; Novi Quadrianto [22] proposed the approach to find the most persistent clique. That clique is subset of vertices, that 1) is almost fully or at least densely connected, 2) has the maximum weight.

The pseudo code of the algorithm to find the dense cliques along with the Weights are presented in Fig.2 This algorithm explores Densely Cliques. Also, the cases where document belong to multiple groups, then that particular document is added which sums to a definite value above the threshold values. This would result to the cliques that are heavily inter-connected and also those cliques' edges weight sums to the maximum value.

Notations:-

The undirected Graph is denoted by G=(V , E) , where V is the set of vertices and E is the set of edges. N(v) represents the set of vertices adjacent to the vertex v. the total number of vertices is denoted by n. U is the set of all the vertices of the graph.

Two vertices are said to be adjacent if they are connected by an edge max is the global variable that gives maximum weighted sum of the clique.

Set[v] is pointer to array : This points to set of vertices that are directly connected to vertex v. It is the neighborhood matrix.

Intersection_Matrix[i][j] : Depicts the count of common nodes in Row i and j of Set[ ].

Function Generate_neighbour_matrix( ) : Function takes weighted similarity matrix as argument and returns Set[v].

Function Generate_Intersection_Matrix( ) : At each level or phase function takes Set[ ], number of document and level number. The mutual intersection of set[ ] array between row and column is found. The row corresponds to previous Cluster_set[ ] entries and column as num of documents. This function Returns the Intersection matrix and absolute average that is average of non zero elements.

Algorithm:

```
Static int Cluster_set[ ];
Static int Counter;
Function Generate_neighbour_matrix( );
Function Generate_Intersection_Matrix( );
Function Process( );
Int Main()
{
Integer  Num_clique = 0, level_no = 1, Counter ;
Cluster_set[ ] =
Cluster_formation(weighted_Similarity_Matrix[ ][ ],
Num_doc);

While( Counter )
{
Final_cluster[ ] = Remove_Duplicacy(Cluster_set[ ],
Counter );
Counter--;
}
}

Function Cluster_formation( W_S_M[ ][ ], Num_doc )
{
Set[v]=Generate_neighbour_matrix(weighted_Similarity_Matrix[ ][ ] );

Counter = Num_doc;
do
{
(Intersection_Matrix,
abs_avg)=Generate_Intersection_Matrix(Set[v],
num_doc , level_no );

Level_no++;

(Cluster_set[ ], Counter) = Process(Intersection_Matrix);
}While( abs_avg  > Th);
}
```

Figure 2 Algorithm to find dense cliques

Function Process( ) : This function accepts argument as intersection matrix and returns the Cluster_set and Number of element in cluster_set[ ]. Counter shows the number of elements in Cluster_set. It is incrementd only if the common elements in intersection matrix[i][j] is greater than threshold value.
Also for such case a new entry is made into Cluster_set[] of next level by merging the row and the column.

Finally, the duplicate elements are removed from the last Cluster_set to get the Cluster.

After the execution of proposed algorithm clusters are visualized by bit wise representation for better understanding. For this generated sample data from proposed algorithm is shown in Fig 3 given below.

4.2.e Cluster visualization Through Bit-Wise Representation

For overlapped clusters where any document might be present in multiple groups, there comes a need to represent such clusters. The above mechanism leads to a matrix (N*M), where N(rows) denotes the document and M(cols) denotes the clusters. If 200000 documents and the total number of groups into which they need to be clustered is 20, then total space required to store this table is

16*200000*sizeof(float)(let, 8 bytes). This total amounts to 25,000 Kb. Using bitwise scheme and Boolean Algebra, the total size to store the information is 2*200000*sizeof(float)(let, 8 bytes) i.e. 3125 Kb.
There comes a need to reduce this space. George Boole[23] in 1854 introduced the concept of Boolean Algebra. Hence, we use bits to store the cluster details, whether the document is present in any particular cluster or not. Value 0000100110010000 in (Xth Row) denotes, document X is present in 5th, 8th, 9th, 12th cluster.

Row -----> Document number
Col -----> Cluster Group number

Representation:

- Let, N be the total number of documents to be clustered. So, Size of array is N i.e. Array [N].
- Let, M be the total number of Cluster Groups. So, total number of bits to represent is M bits.

For Overlapped clusters using Jaccards Similarity matrix, the Bit-wise matrix representation is as follows:-

$$\begin{array}{ll|ccccccc}
 & & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} \\
\text{doc1} & : & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
\text{doc 2} & : & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{doc 3} & : & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{doc 4} & : & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{doc 5} & : & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{doc 6} & : & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{doc 7} & : & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{doc 8} & : & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{doc 9} & : & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{doc 10} & : & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
\text{doc 11} & : & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{doc 12} & : & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{doc 13} & : & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{doc 14} & : & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{doc 15} & : & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{doc 16} & : & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\text{doc 17} & : & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{doc 18} & : & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{doc 19} & : & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\text{doc 20} & : & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}$$

Figure 3 Sample data of 20 documents mapped into 7 cluster groups generated from clustering Algorithm.

Sample data generated using bit wise approach is recorded in fig 3. The following algorithm given below stores the cluster group details as bitwise. Here in the algorithm SET function ON the bit at given position of a document. Read value reads the bit at given Position of the document x.

The operations given below are being used in the algorithm in Fig. 4. Left shifts, shifts all the bits in the integer to one left. Hence, the vacant position created to the rightmost bit is filled by bit 0.

// << : (Left Shift)
// >> : (Right Shift)
// col : (Num_classes)
// row : (Document _ number)

Pseudo code :- For Bit Wise Representaion

```
// Initialisation
int array[No_of_doc] = {0};
static int column = 16; // let integer be of 2byte

// To set the value (0/1) of row(document) and in col(Cluster/Class number).
void Set_Value(int array[ ] , int row , int col)
{           array[row] = array [row]  OR  (1 << col );
}
//print the classes in which Document_num is present
void Print_Value( int array[ ] , int row)
{
            For (int j = 1 ; j <  col ; j++)
            {
                        Bool  temp = 0;
            // right shifting bits of MSB(Most Significant Bit)
            and printing till LSB(Least Significant Bit) is got.
                        Temp = array[row] >> (column – j) ;
                        If(temp == 1)
                        {
                        printf("Doc present in  Jth cluster ");
                        }
            }
}
```

Figure 4 Pseudo code for Bitwise Representation.

Document number	**Binary Number**

array[0]	0000000000011000
array[2]	0000000000000100
array[3]	0000000001100000
.	.
.	.
.	.
array[20]	0000000000000010

Figure 5 Bit-Wise Representation of Documents.

5 Experimental Setup

The work presented in this paper has been conducted in the web security lab of MANIT. MANIT website (www.manit.ac.in) is based on web content management and developed on Joomla2.0. More than 1200 web articles are developed for www.manit.ac.in and more than 7000 end users of the institute availing the web based services through www.manit.ac.in. On daily basis the web team receives document for uploading from more than 30 departments/sections. For experimental purpose the web environment has been set up by hosting the web site resources. The experiments are conducted using language C++. Sample documents which need to be uploaded on MANIT website are taken on offline.

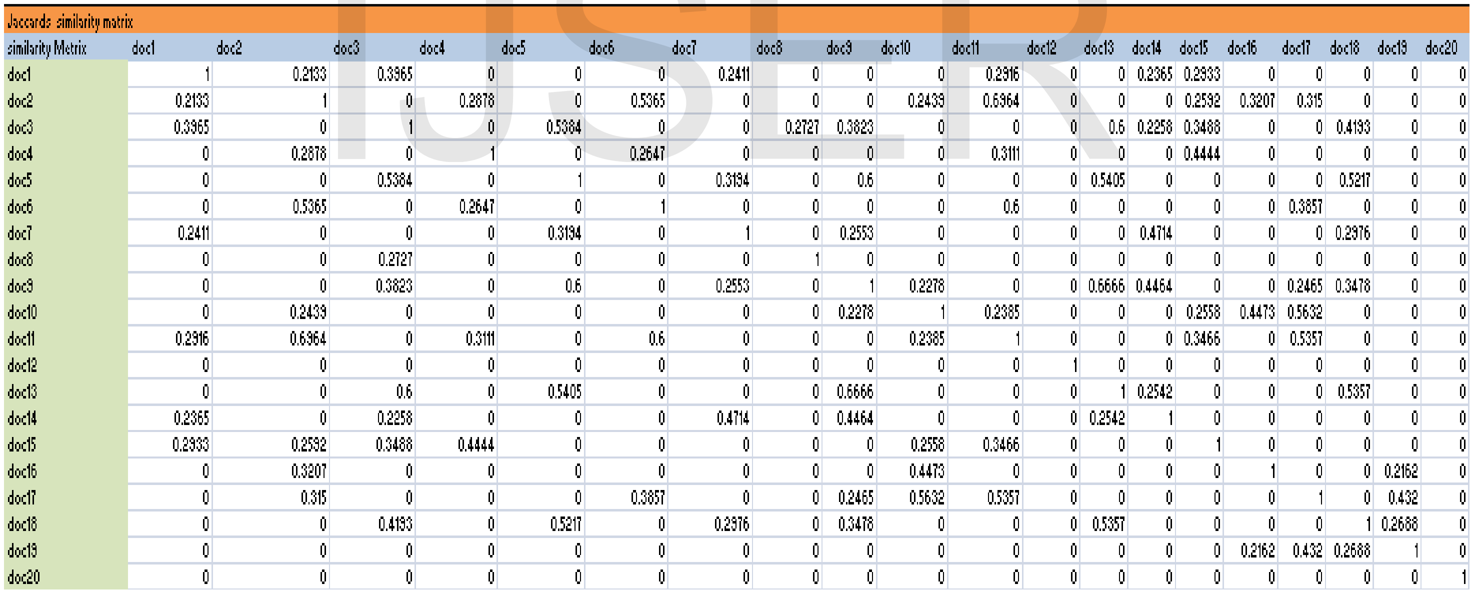
6 Experimental Results

Initially the experiment is done on 20 documents. After performing feature analysis and accepting the words above the Word count gives us Bag of Words. Bag of Words can be defined as the words which are most frequent in a particular document and also they are present in other documents.

$$\text{Term1} = (f1\ \hat{a}1) + (f2\ \hat{a}2) + (f3\ \hat{a}3) + \ldots\ldots\ (fn\ \hat{a}n) \qquad (8)$$

Where,

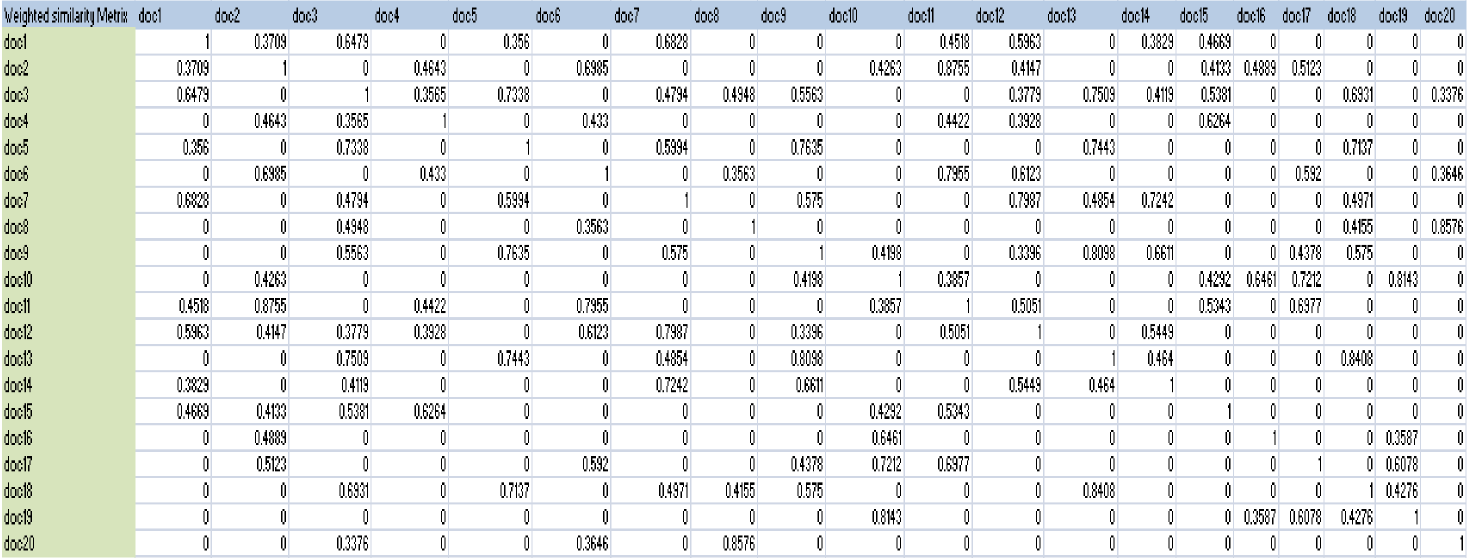
f1 â1 is the frequency of term in doc1 and so on.
â1, â2, â3… are unit vectors of doc1, doc2, doc3.
The threshold value of the word count is the mean of word count in the document as well as mean of its occurrence in different documents so as to generate the inverse document matrix.



Jaccards similarity matrix

similarity Metrix	doc1	doc2	doc3	doc4	doc5	doc6	doc7	doc8	doc9	doc10	doc11	doc12	doc13	doc14	doc15	doc16	doc17	doc18	doc19	doc20
doc1	1	0.2133	0.3965	0	0	0	0.2411	0	0	0	0.2916	0	0	0.2365	0.2933	0	0	0	0	0
doc2	0.2133	1	0	0.2878	0	0.5365	0	0	0	0.2439	0.6964	0	0	0	0.2592	0.3207	0.315	0	0	0
doc3	0.3965	0	1	0	0.5384	0	0	0.2727	0.3823	0	0	0	0.6	0.2258	0.3488	0	0	0.4193	0	0
doc4	0	0.2878	0	1	0	0.2647	0	0	0	0	0.3111	0	0	0	0.4444	0	0	0	0	0
doc5	0	0	0.5384	0	1	0	0.3194	0	0.6	0	0	0	0.5405	0	0	0	0	0.5217	0	0
doc6	0	0.5365	0	0.2647	0	1	0	0	0	0	0.6	0	0	0	0	0	0.3857	0	0	0
doc7	0.2411	0	0	0	0.3194	0	1	0	0.2553	0	0	0	0	0.4714	0	0	0	0.2976	0	0
doc8	0	0	0.2727	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
doc9	0	0	0.3823	0	0.6	0	0.2553	0	1	0.2278	0	0	0.6666	0.4464	0	0	0.2465	0.3478	0	0
doc10	0	0.2439	0	0	0	0	0	0	0.2278	1	0.2385	0	0	0	0.2558	0.4473	0.5632	0	0	0
doc11	0.2916	0.6964	0	0.3111	0	0.6	0	0	0	0.2385	1	0	0	0	0.3466	0	0.5357	0	0	0
doc12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
doc13	0	0	0.6	0	0.5405	0	0	0	0.6666	0	0	0	1	0.2542	0	0	0	0.5357	0	0
doc14	0.2365	0	0.2258	0	0	0	0.4714	0	0.4464	0	0	0	0.2542	1	0	0	0	0	0	0
doc15	0.2933	0.2592	0.3488	0.4444	0	0	0	0	0	0.2558	0.3466	0	0	0	1	0	0	0	0	0
doc16	0	0.3207	0	0	0	0	0	0	0	0.4473	0	0	0	0	0	1	0	0	0.2162	0
doc17	0	0.315	0	0	0	0.3857	0	0	0.2465	0.5632	0.5357	0	0	0	0	0	1	0	0.432	0
doc18	0	0	0.4193	0	0.5217	0	0.2976	0	0.3478	0	0	0	0.5357	0	0	0	0	1	0.2688	0
doc19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2162	0.432	0.2688	1	0
doc20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 6 : Similarity matrix using Jaccard’s Coefficient.



Weighted similarity Metrix	doc1	doc2	doc3	doc4	doc5	doc6	doc7	doc8	doc9	doc10	doc11	doc12	doc13	doc14	doc15	doc16	doc17	doc18	doc19	doc20
doc1	1	0.3709	0.6479	0	0.356	0	0.6828	0	0	0	0.4518	0.5963	0	0.3829	0.4669	0	0	0	0	0
doc2	0.3709	1	0	0.4643	0	0.6985	0	0	0	0.4263	0.8755	0.4147	0	0	0.4133	0.4889	0.5123	0	0	0
doc3	0.6479	0	1	0.3565	0.7338	0	0.4794	0.4948	0.5563	0	0	0.3779	0.7509	0.4119	0.5381	0	0	0.6931	0	0.3376
doc4	0	0.4643	0.3565	1	0	0.433	0	0	0	0	0.4422	0.3928	0	0	0.6264	0	0	0	0	0
doc5	0.356	0	0.7338	0	1	0	0.5994	0	0.7635	0	0	0	0.7443	0	0	0	0	0.7137	0	0
doc6	0	0.6985	0	0.433	0	1	0	0.3563	0	0	0.7955	0.6123	0	0	0	0	0.592	0	0	0.3646
doc7	0.6828	0	0.4794	0	0.5994	0	1	0	0.575	0	0	0.7987	0.4854	0.7242	0	0	0	0.4971	0	0
doc8	0	0	0.4948	0	0	0.3563	0	1	0	0	0	0	0	0	0	0	0	0.4155	0	0.8576
doc9	0	0	0.5563	0	0.7635	0	0.575	0	1	0.4198	0	0.3396	0.8098	0.6611	0	0	0.4378	0.575	0	0
doc10	0	0.4263	0	0	0	0	0	0	0.4198	1	0.3857	0	0	0	0.4292	0.6461	0.7212	0	0.8143	0
doc11	0.4518	0.8755	0	0.4422	0	0.7955	0	0	0	0.3857	1	0.5051	0	0	0.5343	0	0.6977	0	0	0
doc12	0.5963	0.4147	0.3779	0.3928	0	0.6123	0.7987	0	0.3396	0	0.5051	1	0	0.5449	0	0	0	0	0	0
doc13	0	0	0.7509	0	0.7443	0	0.4854	0	0.8098	0	0	0	1	0.464	0	0	0	0.8408	0	0
doc14	0.3829	0	0.4119	0	0	0	0.7242	0	0.6611	0	0	0.5449	0.464	1	0	0	0	0	0	0
doc15	0.4669	0.4133	0.5381	0.6264	0	0	0	0	0	0.4292	0.5343	0	0	0	1	0	0	0	0	0
doc16	0	0.4889	0	0	0	0	0	0	0	0.6461	0	0	0	0	0	1	0	0	0.3587	0
doc17	0	0.5123	0	0	0	0.592	0	0	0.4378	0.7212	0.6977	0	0	0	0	0	1	0	0.6078	0
doc18	0	0	0.6931	0	0.7137	0	0.4971	0.4155	0.575	0	0	0	0.8408	0	0	0	0	1	0.4276	0
doc19	0	0	0	0	0	0	0	0	0	0.8143	0	0	0	0	0	0.3587	0.6078	0.4276	1	0
doc20	0	0	0.3376	0	0	0.3646	0	0.8576	0	0	0	0	0	0	0	0	0	0	0	1

Figure 7: Similarity matrix using Cosine Similarity Formula

This would also be termed as the mean of the magnitude. The value calculated is 5.264.

For instance the set of Bag of Words are given as:-

BoW = {Experience, HOD, Promotion, Register, Relevant, Student}

Bag of Words are further used here to calculate the Inverse term frequency matrix. Similarity Matrix is derived by using different similarity coefficient metric. The similarity matrix depicts the similarity value that rather tells , how much the documents are inter-related to each other.

Figure 8, Showsvalues calculated using Cosine Similarity Coefficient method given in equation (4).

Figure 9, comprises values calculated using Jaccards Similarity Coefficient method given in equation (5). The threshold value is calculated that is the mean of all the edge weight in the similarity matrix. The threshold value is used to find the weight of the edges of the graph.

The threshold value of cosine similarity matrix of table 2 is got to be 0.3334044.

The threshold value of Jaccards similarity matrix of table 3 is got to be 0.211856.

Taking documents as vertex, weighted graph is drawn. The step function in equation (7) is used to determine whether any edge exists between the document and the corrosponding weight between documents as edge weight.

Graph Representation of the Clustered Documents is done using the Algorithm explained above.

Total number of clusters formed using the Jaccard's coefficient Similarity matrix is 6 , that is depicted as:

Overlapped clusters Using Jaccard's Similarity matrix:-

{ G1(13, 10, 9, 7, 3, 8) , G2(18, 13, 10, 9, 3) G3(18, 14, 9, 5, 1) , G4(19, 17, 10, 6, 4, 1) ,G5(16, 15, 11, 2) ,G6(20) , G7(12) }

The Weighted clusters formed by using the Jaccards Similarity matrix give better clusters as compared to the cosine similarity matrix. Fig. 9 describes plot of the performance by using both similarity coefficient used above.

The algorithm used in this paper gives a dense cluster that has the maximum weight in that clique.

7 Performance Evaluation

Performance of the algorithm is determined by calculating F Measure. F measure is basically the harmonic of precision and recall.

Precision is given by following formula:-

$$\mathbf{P} = \frac{(\mathbf{Cm} \cap \mathbf{Ca})}{\mathbf{Ca}} \qquad \textbf{(8)}$$

Whereas Recall is calculated using equation (9) given below. Hence, it may be defined as the ratio of common cluster documents manually and automated machine generated cluster to the manually performed cluster.

$$\mathbf{R} = \frac{(\mathbf{Cm} \cap \mathbf{Ca})}{\mathbf{Cm}} \qquad \textbf{(9)}$$

Where Ca & Cm are defined as,
Ca – Cluster set derived automated machine generated
Cm – Cluster set derived manually
The F-measure "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision"[24].

$$F(\beta) = (1 + \beta^2)\frac{P.R}{\beta^2.P+R} \qquad (10)$$

Efficiency is calculated on the cumulative grounds of F-Measure and Number of cluster formed for n number of documents. The approach in this paper depicts that the Jaccards coefficient driven weighted similarity graph produce efficient clusters than the cosine coefficient driven.

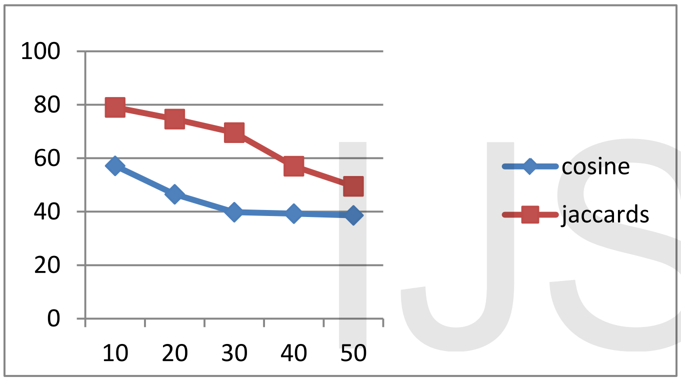


Figure 9: Efficient Cluster Formation : Comparison Graph

X-axis : Number of documents
Y-axis : Efficiency of cluster formation

8 Conclusion

Automated document assignment is developed recently as a wide research topic. This paper presented an approach in which an upcoming document can be automatically assigned to a group on a website. To make the algorithm more effective on real website data, we used weighted graph approach to identify the clusters formed. Experimental results show that approach in this paper using the Jaccards coefficient similarity method and weighted similarity graph and the Clique solving approach leads to efficient clusters. Although, there is lot of scope for further improvement in the proposed. One major challenge is to make the algorithm time efficient for large domain websites where the number of pages and the groups are very large. Further enhancement can be done by optimizing the code for clique algorithm.

Acknowledgment

We would like to express our heartfelt gratitude to all the persons who made this research work successful. Especially to our college MANIT, Bhopal without which supports this paper will not be possible. Also very special thanks to the reviewers for their valuable suggestions.

REFERENCES

[1] Munyaradzi Chiwara, Mahmoud Al-Ayyoub, Mohammad Sajjad Hussain,Rajan Gupta, Prof. Anita Wasilewska presented on the topic Data mining: Text Mining.

[2] Dr Diane McDonald, Intelligent Digital Options and Ursula Kelly, View forth Consulting compiled a report .The Value and Benefit of Text Mining to UK Further and Higher Education for JISC(2012).

[3] Michael Steinbach, George Karypis, Vipin Kumar "A Comparison of Document Clustering Techniques. "Department of Computer Science and Engineering, University of Minnesota ,Technical Report #00-034.

[4] Ying Zhao and George Karypis Department of Computer Science, University of Minnesota, Minneapolis, MN 55455.

[5] Dragomir R. Radev a, Hongyan Jing b and Małgorzata Stys, "Centroid-Based Summarization of Multiple Documents", International Journal of Information Processing and Management(2004).

[6] P. Willet. "Recent Trends in Hierarchical Document Clustering: a Critical Review." Information Processing and Management, 24:577-97, 1988.

[7] B. Larsen and C. Aone. "Fast and Effective Text Mining using Linear-Time Document Clustering." In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.

[8] Alba, Richard D. "A Graph-Theoretic denition of a sociometric clique. Journal of Mathematical Sociology, 3(1):113-126, 1973.

[9] L. Babel , A fast algorithm for maximum weight clique problem , computing 52 (1994) 31-38.

[11] Pyle, D., 1999. *Data Preparation for Data Mining.* Morgan Kaufmann Publishers, Los Altos, California.

[12] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Data Preprocessing for Supervised Leaning", *International Journal of Computer Science*, 2006, Vol 1 N. 2, pp 111–117.

[13] Han, J., Kamber, M. *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001. ISBN 1-55860-489-8.

[14] Lovins, Julie Beth (1968). "Development of a Stemming Algorithm". *Mechanical Translation and Computational Linguistics* **11**: 22–31.

[15] The Porter Stemmer, Daniel Waegel CISC899/ FALL 2011.

[16] Miki yamamoto, Kenneth W. Church AT & T Labs – Research," Using Suffix Array to Compute Term Frequency and Document Frequency for all Substrings in a Corpus.

[17] Anna Huang ,"Similarity Measures for Text Document Clustering", NZCSRSC 2008, April 2008, Christchurch, New Zealand

[18] An Introduction to Cluster Analysis for Data Mining.

[19] Jaccard, Paul. _Etude comparative de la distribution orale dans une portion des alpes et des jura. Bul- letin del la Soci_et_e Vaudoise des Sciences Naturelles,37:547-579, 1901.

[21]R. Carraghan, P.M. Pardalos," An Exact Algorithm for the Maximum Clique Problem", Oper. Res. Lett. 9 (1990) 375–382.

[22] Novi Quadrianto , Chao Chen , Christoph H. Lampert, "On The Most Persistent Soft-Clique in a Set of Sampled Graphs. In Proceedings of the 29 th International Conference on Machine Learning", Edinburgh, Scotland, UK, 2012.

[23] Boole, George [1854]. " An Investigation of the Laws of Thought."Prometheus Books.

[24] Van Rijsbergen, C. J. (1979) " Information Retrieval (2nd ed.)" Butterworth.