International Journal of Advanced Research Methodology in Engineering & Technology,
Volume 1, Issue1, December 2016

An Approach to Clustering of Text Documents with n-Texts Input Using Graph Mining Techniques

Bapuji Rao[1] and Sabari Giri Murugan[2]
1 Sr. Faculty-IT (Gr-3), Department of ITIMS, iNurture Education Solutions Pvt. Ltd., Bangalore, India.
Mail ID: rao.bapuji@gmail.com
2 Faculty (Gr-2), Department of IT, iNurture Education Solutions Pvt. Ltd., Bangalore, India.
Mail ID: sabarimtech08@gmail.com

ABSTRACT

The authors propose a simple approach of clustering of text documents with a given n-words input using graph mining techniques. This proposed approach clusters text documents in three forms based on identifying of n-words, (n-1)-words, and (n-2)-words respectively from text documents. These three forms of clustering of text documents with identifying all the n-words, (n-1)-words, and (n-2)-words from n-words input forms from set of text documents. These three forms of clustering are treated as document-word relation and finally represented as bi-partite graphs. For this the authors have proposed an algorithm for three forms of clustering of text documents for a given n-words input using graph mining techniques. Finally, the paper concludes with the result analysis of the proposed algorithm by implementing the proposed technique using C++ programming language and observed satisfactory results.

Key Words: Bi-partite graph, Clustering, Sub-graph, Un-oriented incidence matrix.

1. INTRODUCTION

A clustering can be defined as grouping of similar objects in the data of similar characteristics. The problem of clustering can be very useful in the text domain. Document clustering has been extensively used in a number of different areas of text mining and information retrieval. Clustering especially helps of organizing documents in a structural way to improve retrieval and browsing those documents. The study of the clustering problem precedes its applicability to the text domain. Text document clustering is a selection of text documents with the particular word(s)/text(s) present. So each group of text documents called cluster of text documents of a particular word's presence. Clustering is the most common form of unsupervised learning and no supervision means that there is no human expert who has assigned documents to cluster. In text document clustering, it is the distribution and makeup of the text documents as a group based on a particular word present in all the grouped text documents. Clustering is sometimes referred to as automatic classification.

In text document clustering, a group of words (texts) are used on a set of text documents for discovering such text documents having with the given set of words (texts). Further such discovered text documents for the given set of words (texts) are grouped into that many cluster of text documents.

2. LITERATURE SURVEY

According to Aggarwal & Zhai [1] both feature selection and feature transformation methods such as Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Analysis (PLSA), and Non-negative Matrix Factorization (NMF) are used to improve the quality of the document representation and make it more efficient to text clustering. Feature selection is more common and easy to apply in text clustering in which supervision is available for the feature selection process proposed by [2]. Since the results of text clustering are highly dependent on document similarity. Such cases the concept of term contributed by [3] is applied. So the contribution of a term can be viewed as its contribution to document similarity.

The technique of concept decomposition uses any standard clustering technique has been studied in [4, 5] on the original representation of the documents. The frequent terms in the centroids of these clusters are used as basis vectors which are almost orthogonal to one another. The documents can then be represented

in a much more concise way in terms of these basis vectors. So the condensed conceptual representation allows for enhanced clustering as well as classification of text documents. Therefore, a second phase of clustering can be applied on this condensed representation in order to cluster the documents much more effectively [6]. Such a method is tested in [7] by using word-clusters in order to represent documents.

An algorithm for clustering of text documents for a given set of words has been proposed by [9]. It clusters text documents having for a given set of words (texts). Finally the document-word relation is represented as a bi-partite graph.

3. PROPOSED ALGORITHMS

Algorithm *n*-Words_DocumentsClustering()

```
Algorithm Convention [8]
// Global Declarations
Document.Txt: Text file to hold m text documents name.
Word.Txt: Text file to hold n words (texts).
Document [m]: To assign m text documents name.
Word [n]: To assign n words.
Frequency [m][n]: Bit-matrix of order mXn which contains 0s and 1s.
{

 // read text document names from file
 open("Document.Txt");
read(m); // total number of text documents
i:=1;
while(not EOF())
do
{
   // read text documents name from file
     read(Document[i]);
     i:=i+1;
}
close("Document.Txt");

  // read words from file
     open("Word.Txt");
     read(n); // total number of words
     i:=1;
     while(not EOF())
     do
     {
       // read words from file
          read(Word[i]);
          i:=i+1;
     }
      close("Word.Txt");
UnorientedIncidenceMatrixCreation();
UnorientedIncidenceMatrixDisplay();
Cluster_Formation();
}
```

3.1. Procedure for Document Detection

Procedure Detect_Documents (Frequency, nw, tnw)

```
Frequency [m][n]: Bit-matrix of order mXn which contains 0s and 1s.
nw: Total number of words.
tnw: Total number of words used for cluster.
Cluster[m]: Array to hold the text document index.
{
for i:=1 to m do
{
   count:=0;
   for j:=1 to nw do
      if(Frequency[i][j]=1) then count:=count+1;

     if(count=tnw)
     {
       k:= k+1;
      //assignment of document index
       cluster[k]:=i;
       flag:=1;
      }
}
if(flag=1)
{
//Display cluster of documents with 'tnw' Word's
//Presence
  for i:=1 to nw do
       write(Word[i]);

   for i:=1 to k do
   {
     write(Document[cluster[i]]);
     for j:=1 to nw do
        write(Frequency[cluster[i]][j]);
    }
}
else
     write("No Documents Clustering");
}
```

3.2. Procedure for Cluster Formation

Procedure Cluster_Formation( )

```
{
//call procedure Detect_Documents() for 3
//times for 3 no. of clusters
  for i:=0 to 2 do
```

```
  {
    Detect_Documents(Frequency, n, n-i);
   }
}
```

3.3. Procedure for Unoriented Incidence Matrix Display

```
Procedure UnorientedIncidenceMatrixDisplay( )
{
  for i:=1 to n do
       write(Word[i]);

  for i:=1 to m do
  {
     write(Document[i]);
     for j:=1 to n do
          write(Frequency[i][j]);
  }
}
```

3.4. Procedure for Unoriented Incidence Matrix Creation

```
Procedure UnorientedIncidenceMatrixCreation()
flag[m]: To assign the files status i.e. availability
or not in the disk.
{
  //Check every file names' availability
    for i:=1 to m do
    {
       flag[i]:=0;
     //open i^th text document for reading
       open(Document[i]);
     if(Not_Found(Document[i])) then flag[i]:=1;
       close(Document[i]);
    }
       status:=0;
       for i:=1 to m do
       {
          if(flag[i]=1) then
          {
            write(Document[i],"Not Found");
            status:=1;
          }
           flag[i]:=0;  //reassignment
       }

       if(status=1) then
         write("No Text Document Clustering");
       else // status is OK
       {
         for i:=1 to m do
         {
            open(Document[i]);
            st:=" ";
//read a character from Document[i] and assign to
//ch
            read(ch);
            while(not EOF())
            do
    {
      if(ch=space or ch='\n')
      {
//check st word is present in Word[] array
        for j:=1 to n do
        {
         if (Word[j]=st) then Frequency[i][j]:=1;
         else Frequency[i][j]:=0;
         }
          st:=" ";
      }
     else st:= st + ch;

//read a character from Document[i] and assign to
//ch
            read(ch);
          } //while close
            close(Document[i]);
        }// for
     }//else
}
```

The proposed algorithm has three phases. Phase-I reads two text files as datasets namely "*Document.Txt*" and "*Word.Txt*" which contains *m*-text document names and *n*-words respectively. First it opens the text file "*Document.Txt*" and make available of *m*-text document names in the array called Document[*m*]. Secondly it opens the text file "*Word.Txt*" and make available of *n*-words in the array called Word[*n*]. So Phase-I is all about reading of data i.e. *m*-text document names and *n*-words, and assign to the arrays Document[*m*] and Word[*n*] respectively.

Phase-II calls the procedure UnorientedIncidenceMatrixCreation() to searching of *n*-words, which is in Word[*n*] from *m*-text documents, which is in Document[*m*]. So the presence of *n*-words, Word[*n*] are searched from *m*-text documents, Document[*m*] and the result i.e. bit value 1 is assigned in the matrix, Frequency[*m*][*n*] of order *m*-text documents X *n*-words. Then the procedure UnorientedIncidenceMatrixDisplay( ) is called to display the matrix, Frequency[*m*][*n*]. So Phase-II is about creation of Document-Word un-oriented incidence matrix [8, 9] and display.

Phase-III calls the procedure Cluster_Formation( ) for creation of three number of clusters from *n* number of words having with each cluster of documents with *n*, (*n-1*), and (*n-2*) number of words (texts) presence in *m* number of documents. The procedure Detect_Documents (Frequency, nw, tnw) is called three times to create three number of text document clusters. So Phase-III is about creation and display of cluster of text documents.

4. EXAMPLE

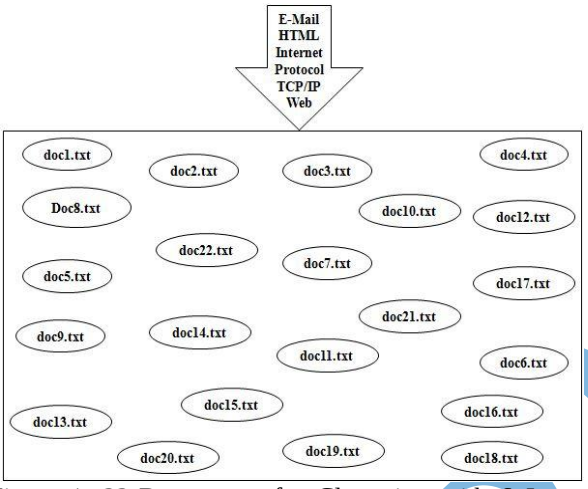


Figure 1: 22-Documents for Clustering with 6 Input Words (Texts)

The authors have considered twenty two number of text documents namely from *doc1.txt* to *doc22.txt* for clustering of three sets of clusters with the help of six input words (texts) namely {*E-Mail*, *HTML*, *Internet*, *Protocol*, *TCP/IP*, *Web*} and depicted in "Figure 1".

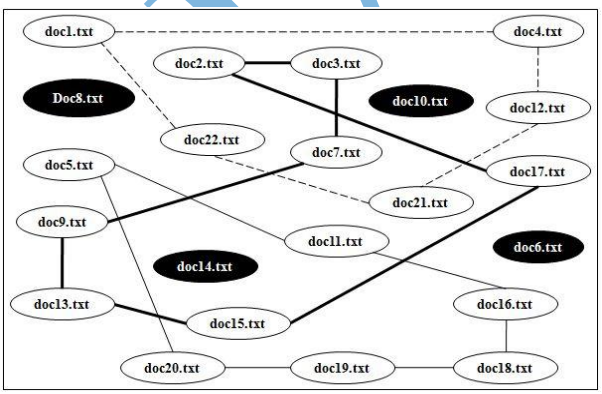
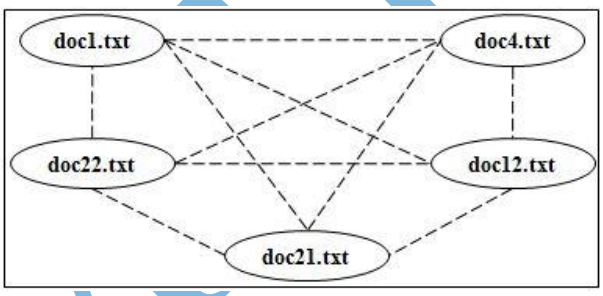
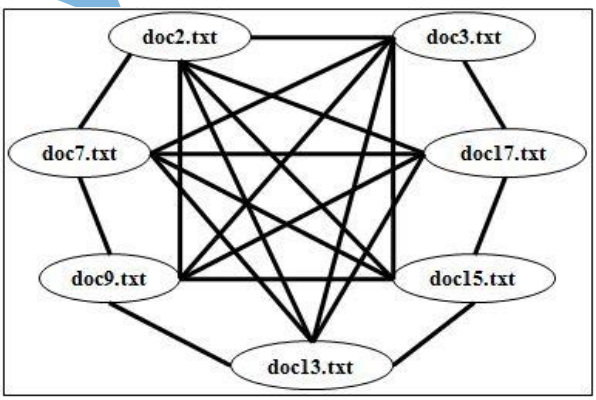


Figure 2: Three Sets of Clustering's Comprising of 6-Words, 5-Words, and 4-Words Appearance

After the verification process of all the six words (texts) on all the twenty two text documents, it successfully creates three relations with appearance of *six words*, *five words*, *and four words* respectively. The **thin line** is the indication of relation of text documents with *six words* appearance. The **thick line** is the indication of relation of text documents with any *five words* appearance. Similarly the **dashed line** is the indication of relation of text documents with any *four words* appearance. These lines of connectivity are depicted in "Figure 2".



From "Figure 2", these three relation graphs are separated and represented individually as

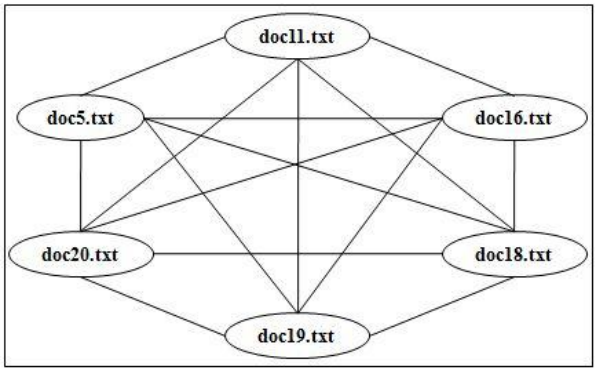


clustering of documents with appearance of *four*, *five*, and *six* words respectively. These clustering are depicted from "Figure 3" to "Figure 5".

Figure 3: Clustering of Documents with Any 4-Words Appearance

Figure 4: Clustering of Documents with Any 5-Words Appearance

Figure 5: Clustering of Documents with 6-Words Appearance



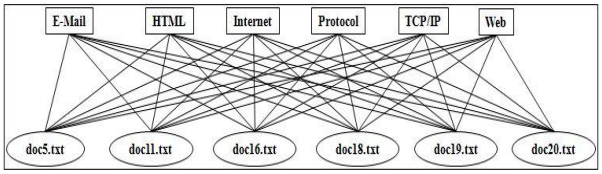


Figure 6: Bi-Partite Graph of Document-Word with Occurrence of Any Four Words

Further using "Figure 3" to "Figure 5", the document-word relationships can be represented as bi-partite graphs [8], and are depicted from "Figure 6" to "Figure 8".

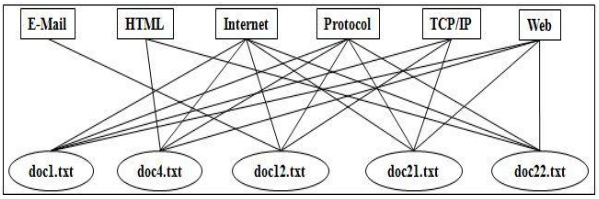


Figure 7: Bi-Partite Graph of Document-Word with Occurrence of Any Five Words

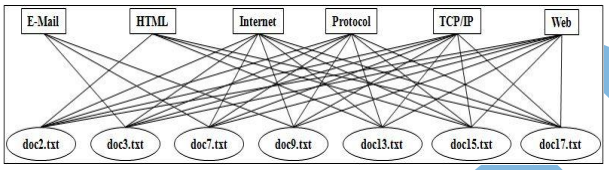


Figure 8: Bi-Partite Graph of Document-Word with Occurrence of Six Words

5. EXPERIMENTAL RESULTS

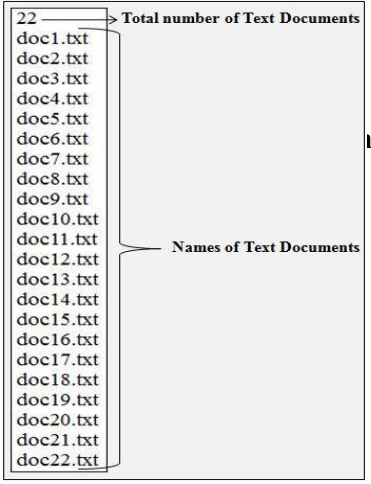


Figure 9: Dataset File Document.Txt

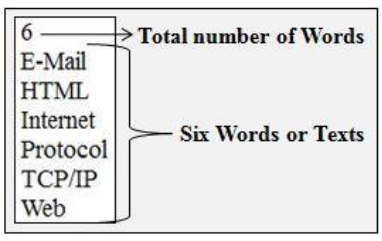


Figure 10: Dataset File Word.Txt

The details of text documents such as number of text documents and the text document names are stored in a text file called "*Document.Txt*". The 1st row indicates total number of document names. The 2nd row onwards is the indication of document names. Similarly the details of words such as number of words and unique words are stored in a text file called "*Word.Txt*". The 1st row indicates total number of words. The 2nd row onwards is the indication of unique word names. These two text files are considered as dataset to the proposed algorithm and depicted in "Figure 9" and "Figure 10". When these datasets are input to the experiment depicted in "Figure 11", the algorithm open both the data files and read into the arrays Document[*m*] and Word[*n*] respectively. Then the procedure **UnorientedIncidenceMatrixCreation( )** is called for creation of frequency matrix of Document-Word, Frequency[*m*][*n*]. The initial form and the resultant form of un-oriented incidence matrix are depicted in "Figure 12" and "Figure 13" respectively. In the example the total number of documents are twenty two i.e. {*doc1.txt*, *doc2.txt*, *doc3.txt*, *doc4.txt*, *doc5.txt*, *doc6.txt*, *doc7.txt*, *doc8.txt*, *doc9.txt*, *doc10.txt*, *doc11.txt*, *doc12.txt* , *doc13.txt*, *doc14.txt*, *doc15.txt*, *doc16.txt*, *doc17.txt*, *doc18.txt*, *doc19.txt*, *doc20.txt*, *doc21.txt*, *doc22.txt* } and the total number of words are six i.e. {*E-Mail*, *HTML*, *Internet*, *Protocol*, *TCP/IP*, *Web*}. Hence the order of *Document-Word un-oriented incidence matrix*, Frequency[*m*][*n*] is *22X6*.

Finally, the procedure **Cluster_Formation ( )** forms three number of clusters i.e. appearance of ***any four words***, ***any five words***, and ***six words*** in a set of 22 text documents. These three clusters are depicted from "Figure 14" to "Figure 16".

```
Enter Text Documents File Name :- Document.Txt

Enter Word File Name :- Word.Txt
```

Finally the authors have represented these three clusters as bi-partite graphs and are depicted from "Figure 6" to "Figure 8".

The algorithm was written in *C++ programming* and compiled with *TurboC++*. The experiment was run on Intel Core I5-3230M CPU + 2.60 GHz Laptop with 4GB memory running MS-Windows 7.

Figure 11: Input of Document and Word File Name

The Initial State of Frequency Table of Document - Words

	E-Mail	HTML	Internet	Protocol	TCP/IP	Web
doc1.txt	0	0	0	0	0	0
doc2.txt	0	0	0	0	0	0
doc3.txt	0	0	0	0	0	0
doc4.txt	0	0	0	0	0	0
doc5.txt	0	0	0	0	0	0
doc6.txt	0	0	0	0	0	0
doc7.txt	0	0	0	0	0	0
doc8.txt	0	0	0	0	0	0
doc9.txt	0	0	0	0	0	0
doc10.txt	0	0	0	0	0	0
doc11.txt	0	0	0	0	0	0
doc12.txt	0	0	0	0	0	0
doc13.txt	0	0	0	0	0	0
doc14.txt	0	0	0	0	0	0
doc15.txt	0	0	0	0	0	0
doc16.txt	0	0	0	0	0	0
doc17.txt	0	0	0	0	0	0
doc18.txt	0	0	0	0	0	0
doc19.txt	0	0	0	0	0	0
doc20.txt	0	0	0	0	0	0
doc21.txt	0	0	0	0	0	0
doc22.txt	0	0	0	0	0	0

Figure 12: Initial form of Un-Oriented Incidence Matrix of Documents-Words

The Resultant Frequency Table of Document - Words

	E-Mail	HTML	Internet	Protocol	TCP/IP	Web
doc1.txt	0	0	1	1	1	1
doc2.txt	0	1	1	1	1	1
doc3.txt	1	0	1	1	1	1
doc4.txt	0	1	1	1	0	1
doc5.txt	1	1	1	1	1	1
doc6.txt	0	0	1	1	1	0
doc7.txt	1	0	1	1	1	1
doc8.txt	0	0	0	0	0	1
doc9.txt	1	0	1	1	1	1
doc10.txt	0	1	0	0	0	1
doc11.txt	1	1	1	1	1	1
doc12.txt	1	0	1	1	1	0
doc13.txt	0	1	1	1	1	1
doc14.txt	0	1	0	0	0	1
doc15.txt	0	1	1	1	1	1
doc16.txt	1	1	1	1	1	1
doc17.txt	0	1	1	1	1	1
doc18.txt	1	1	1	1	1	1
doc19.txt	1	1	1	1	1	1
doc20.txt	1	1	1	1	1	1
doc21.txt	0	0	1	1	1	1
doc22.txt	0	1	1	1	0	1

Figure 13: Resultant form of Un-oriented Incidence Matrix of Documents-Words

Cluster of Documents with 4 Word's Presence

	E-Mail	HTML	Internet	Protocol	TCP/IP	Web
doc1.txt	0	0	1	1	1	1
doc4.txt	0	1	1	1	0	1
doc12.txt	1	0	1	1	1	0
doc21.txt	0	0	1	1	1	1
doc22.txt	0	1	1	1	0	1

Figure 14: Un-oriented Document-Word Incidence Matrix of Any Four Words Presence

Cluster of Documents with 6 Word's Presence

	E-Mail	HTML	Internet	Protocol	TCP/IP	Web
doc5.txt	1	1	1	1	1	1
doc11.txt	1	1	1	1	1	1
doc16.txt	1	1	1	1	1	1
doc18.txt	1	1	1	1	1	1
doc19.txt	1	1	1	1	1	1
doc20.txt	1	1	1	1	1	1

Figure 15: Un-oriented Document-Word Incidence Matrix of Any Five Words Presence

Cluster of Documents with 5 Word's Presence

	E-Mail	HTML	Internet	Protocol	TCP/IP	Web
doc2.txt	0	1	1	1	1	1
doc3.txt	1	0	1	1	1	1
doc7.txt	1	0	1	1	1	1
doc9.txt	1	0	1	1	1	1
doc13.txt	0	1	1	1	1	1
doc15.txt	0	1	1	1	1	1
doc17.txt	0	1	1	1	1	1

Figure 16: Un-oriented Document-Word Incidence Matrix of Six Words Presence

6. CONCLUSION

The authors have proposed an approach of three sets of clustering of text documents on an input of *n*-words (texts) using graph mining techniques. Initial portion of the work is a brief review of the literature on clustering of text documents in data mining. The cluster of document-word pair is represented as a bi-partite graph since the technique is graph theoretic. In memory the bi-partite graph is represented as an un-oriented incidence matrix. Finally with the help of un-oriented incidence matrix, the algorithm is started formation of three sets of clustering of text documents for *n*, *(n-1)*, and *(n-2)* numbers of words (texts) appearances. The proposed algorithm was implemented using C++ programming language with the datasets in the "Figure 9" and "Figure 10", and observed satisfactory results.

REFERENCES

[1] A Survey of Text Clustering Algorithms, C. Aggarwal & C. Zhai, Mining Text Data, Springer US, Pp. 77-128, 2012.

[2] A Comparative Study on Feature Selection in Text Categorization, Y. Yang & J. O. Pedersen, ICML, Volume No. 97, Pp. 412-420, 1997.

[3] An Evaluation on Feature Selection for Text Clustering, T. Liu, S. Liu, Z. Chen, & W. Y. Ma, ICML, Volume No. 3, Pp. 488-495, 2003.

[4] On Effective Conceptual Indexing and Similarity Search in Text Data, C. Aggarwal & P. S. Yu, IEEE International Conference on Data Mining, San Jose, CA, Unites States, Pp. 3-10, 2001.

[5] Concept Decompositions for Large Sparse Text Data Using Clustering, I. S. Dhillon & D. S. Modha, Machine Learning, Volume No. 42(1-2), Pp. 143-175, 2001.

[6] An Introduction to Modern Information Retrieval, G. Salton, McGraw-Hill, Inc. New York, NY, USA, 1983.

[7] Document Clustering Using Word Clusters via the Information Bottleneck Method, N. Slinim & N. Tishby, 23rd Annual International ACM SIGIR Conference on Research and Development

in Information Retrieval, New York, NY, USA, Pp. 208-215, 2000.

[8] Fundamentals of Data Structures in C++ (2nd Edition), University Press (India) Private Limited, 3-6-747/1/A & 3-6-754/1, Himayat Nagar, Hyderabad, AP-500029, India, 2013.

[9] An Approach to Clustering of Text Documents Using Graph Mining Techniques, Bapuji Rao & B. K. Mishra, IJRSDA, IGI Publishing, New York, Volume No. 4, Issue 1, Article 5, 2016.