International Journal of Science Engineering and Advance Technology

An Approach to Text Documents Clustering with {n, n-1, ….., 1}-Word(s) Appearance Using Graph Mining Techniques

Bapuji Rao, Saroja Nanda Mishra
Research Scholar, CSE, BPUT, Rourkela, India
CSE&A, IGIT, Sarang, Dhenkanal, India
bapuji.research@gmail.com, sarose.mishra@gmail.com

Abstract— **This paper is about text document clustering with an input of n words. Initially a cluster of all text documents with extension name ".Txt" from m-documents of various types is formed. Then on an input of n-words, the proposed algorithm starts *n, n-1, n-2,.....,1* sets of cluster. Each cluster of text documents with the presence of *n, n-1, n-2,......,1* word(s) respectively. These *n*-forms of clustering are treated as documents-words relation and in memory it is represented as un-oriented documents-words incidence matrix. Finally these un-oriented documents-words incidence matrices are represented as bi-partite graphs, since the bi-partite graph has two sets of nodes namely document and word. The proposed algorithm using graph mining techniques was implemented using C++ programming language and the result was satisfactory.**

Keywords—document cluster; bi-partite graph; document sub-graph; un-oriented documents-words incidence matrix

I. Introduction

Document clustering are extensively used in the 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 is related to the 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 unsupervised learning it means there is no need of human interference for clustering of documents. 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.

II. Literature Survey

The 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. So that text clustering is done in more efficient way. These two features are proposed by [2]. The common and easy way to apply in text clustering is the Feature selection in which supervision is available is proposed by [9]. Text clustering is highly dependent on document similarity. The concept of term contributed proposed by [8] is applied for document clustering. 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 [3, 6] on the original representation of the documents. Each document can be represented as a vector. So the vector representation of document make more 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 proposed by [5]. To represent documents word (text) the method word-clusters were used and proposed by [7]. An algorithm for clustering of text documents for a given set of words using graph mining techniques has been proposed by [1]. It clusters text documents having for a given set of words (texts). Finally the document-word relation as a cluster is represented as a bi-partite graph.

III. Proposed Algorithms

A. *Algorithm for Text Document Clustering for n-Words with n-Clusters of {n, n-1, n-2, ...,1}-Words*

```
Algorithm nCluster( )
// Global Declarations [Algorithm convention [4]]
Document.Txt: Text file to hold m-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.
DocWord[m][n]: Un-Oriented Documents-Words incidence matrix of order mXn.
{
 // read document names from file
 open("Document.Txt");
 read(m); // total number of documents
  // clustering of text documents with extension name
".Txt"
  // and store in the array Document[m]
   i:=0;
   while(not EOF())
   do
   {
     // read text documents name from file
        read(DocumentName);
        Extension:=Get_Extension_Name(DocumentName);
        if(Extension=".Txt")
               then { i:=i+1;  Document[i]:=DocumentName;
}
    }
     close("Document.Txt");
  // assign the actual number of ".Txt" documents i.e. 'i' in
'm'
     m:=i;
  // 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");
      WUIMatrix("Imatrix.Txt");
      CUIMatrix( );
      WUIMatrix("Rmatrix.Txt");
      ClusterFormation( );
}
```

B. Procedure for Detection of Documents

Procedure DetectDocuments(DocWord, words, twords)

DocWord [m][n]: Un-Oriented Documents-Words matrix of order mXn.
words: Actual number of words.
twords: Total number of words used for cluster.
Cluster[m]: To assign the text document index.

```
{
 for i:=1 to m do
  {
     count:=0;
     for j:=1 to words do
          if(DocWord[i][j]=1) then count:=count+1;

     if(count=twords)
    {
       k:= k+1;
       //to assign text document index
       Cluster[k]:=i;
       flag:=1;
     }
  }
 if(flag=1)
 {
   // write cluster of documents with 'tword' word(s)
presence
   for i:=1 to words do
           write(Word[i]);
     for i:=1 to k do
     {
        write(Document[Cluster[i]]);
             for j:=1 to words do
               write(DocWord[Cluster[i]][j]);
     }
 }
 else
      write("No Documents Clustering");
}
```

C. Procedure for Formation of Clusters

Procedure ClusterFormation ( )

```
{
   // call procedure DetectDocuments() for n-times for 'n'
no. of
   // clusters
      for i:=0 to n-1 do
      {
        DetectDocuments(DocWord, n, n-i);
       }
}
```

D. Procedure for Writing Un-Oriented Documents-Words Incidence Matrix in Text File

Procedure WUIMatrix(FileName)

FileName: To hold the file name for writing the output of Un-Oriented Documents-Words Incidence Matrix.

```
{
  open(FileName); // open FileName for writing purpose
  for i:=1 to n do
      write(Word[i]); // write in FileName
  for i:=1 to m do
   {
     write(Document[i]); //write in FileName
      for j:=1 to n do
            write(DocWord[i][j]); //write in FileName
   }
}
```

E. Procedure for Creation of Un-Oriented Documents-WordsIncidence Matrix

Procedure CUIMatrix( )

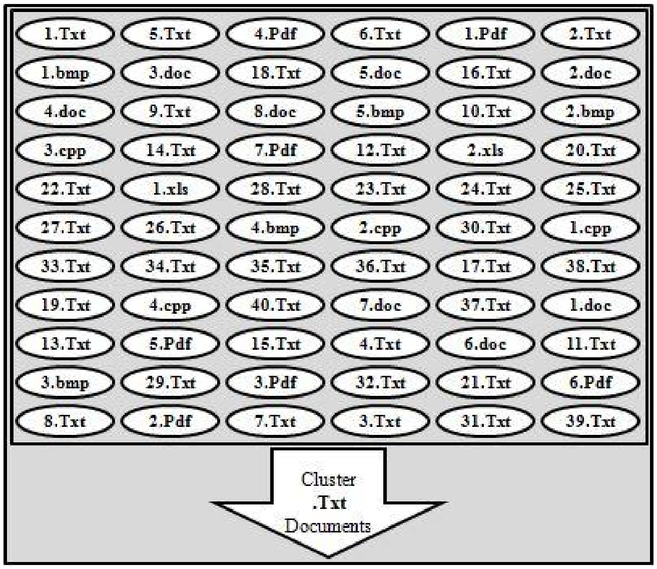
flag[m]: To assign the files status i.e. availability or not in the disk

```
{
  //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 or not
               for j:=1 to n do
           {
             if (Word[j]=st) then DocWord[i][j]:=1;
            else DocWord[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 reads two text files as datasets namely "Document.Txt" and "Word.Txt" which contains *m*-document names and *n*-words respectively. Using "Document.Txt", it clusters only text documents whose extension name is ".Txt". These cluster of text documents are assigned 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]. Now *n*-numbers of cluster of text documents are formed with the help of input of *n*-words.

The procedure WUIMatrix("Imatrix.Txt") is called to write the initial form of un-oriented documents-words incidence matrix [1, 8], DocWord[m][n] in a text file "Imatrix.Txt". Then the procedure CUIMatrix( ) is called 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 values 1 are assigned in the matrix, DocWord[m][n] of order *m*-text documents X *n*-words. Then the procedure WUIMatrix("Rmatrix.Txt") is called to write the resultant form of un-oriented documents-words incidence matrix, DocWord[m][n] in the text file "Rmatrix.Txt".



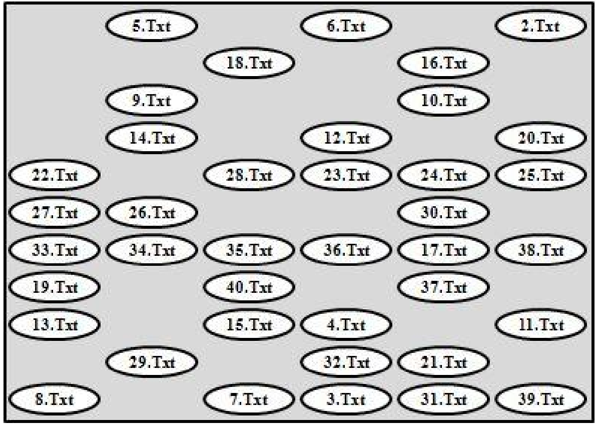
The procedure ClusterFormation( ) for creation of *n* number of clusters from *n* number of words having with each cluster of text documents with *n*, (*n-1*), (*n-2*), ….., *1* number of word(s) appearances in *m* number of text documents. The procedure DetectDocuments(DocWord, n, n-i) is called *n* times to create *n* number of text document clusters.

IV. EXAMPLE

Fig. 1. Sixty six different types of documents for clustering of text documents

"Fig. 1" consists of sixty six numbers of documents comprising of five types of documents such as ".Txt", ".Pdf", ".doc", ".cpp", and ".bmp". The authors aim is to clustering of only text documents with extension name ".Txt". First the authors cluster all text documents whose extension names are ".Txt". Now it clusters of text documents comprising of forty text documents and is depicted in "Fig. 2".

Fig. 2. Text documents clustering comprising of forty ".Txt" documents



The authors to cluster of these forty text documents into six numbers of clusters by applying six numbers of words (texts) on it and is depicted in "Fig. 3". The six words are {E-Mail, Internet, HTML, Protocol, TCP/IP, Web}. When these six words are applied on "Fig. 2", it starts formation of six types of cluster since there are six types of words. Each cluster is a set of documents with a particular number of words (texts) appearances in it. Each cluster is treated as one set of text documents with the presence of a six-words, five-words, four-words, three-words, two-words, and one-word respectively. Hence it is treated as document sub-graph with a particular number of words relationships. The authors successfully forms six document sub-graphs from "Fig. 3". These six document sub-graphs are depicted from "Fig. 4" to "Fig. 9".

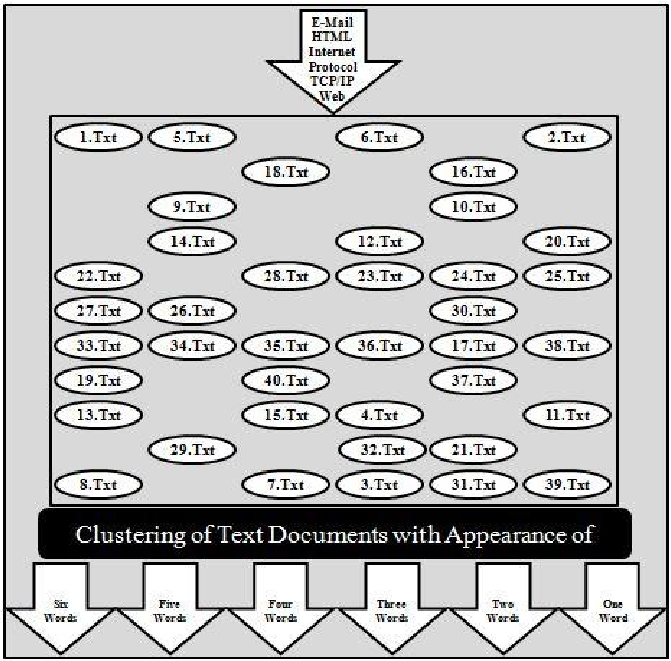


Fig. 3. Clustering of text documents with appearance of six, five, four, three, two, and one word(s)

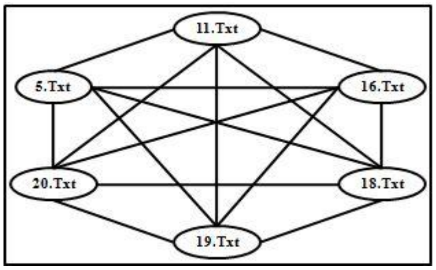


Fig. 4. Clustering of documents with 6-words appearance

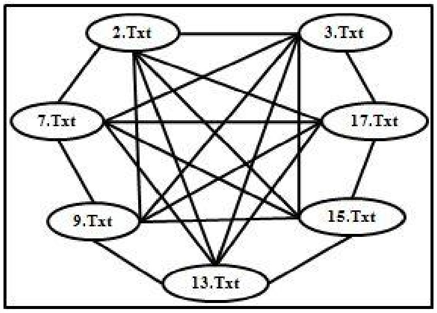


Fig. 5. Clustering of documents with 5-words appearance

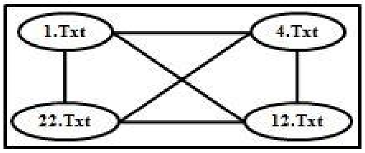


Fig. 6. Clustering of documents with 4-words appearance

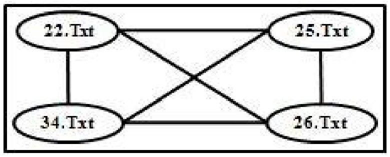
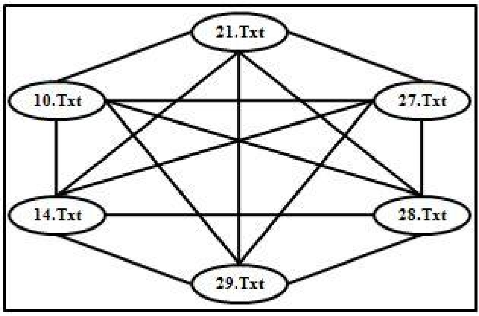
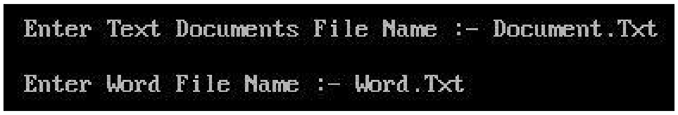


Fig.7. Clustering of documents with 3-words appearance





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

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

Fig. 8. Clustering of documents with 2-words appearance

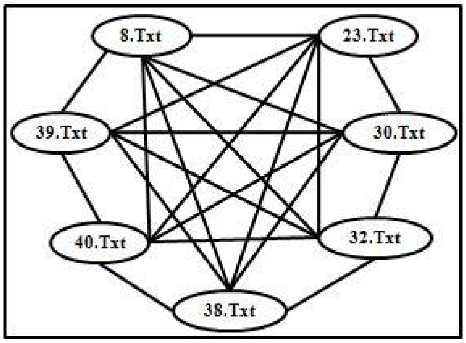


Fig. 9. Clustering of documents with 1-word appearance

Further the authors successfully represent all the six document sub-graphs into documents-words bi-partite graph [4] and are depicted from "Fig. 10" to "Fig. 15" respectively. These six bi-partite graphs have two types of nodes such as document node and word node respectively. The relationship between the document node and word node is an edge which is the indication of presence of word in that document.

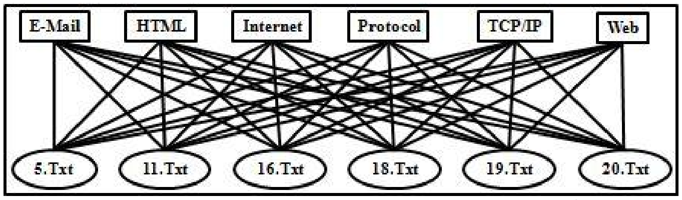


Fig. 10. Documents-Words Bi-Partite Graph with occurrence of six words

In memory the bi-partite graph can be represented as un-oriented documents-words incidence matrix which is only consists of 0s and 1s. The indication of 1 is an edge between the document node and word node. Similarly the indication of 0 means, there is no edge between the document node and word node.

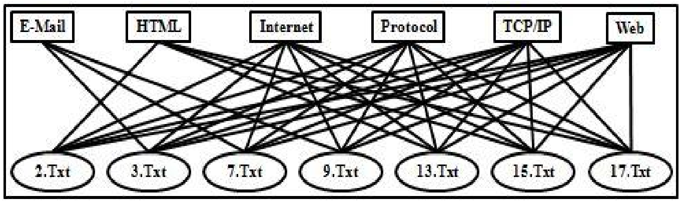


Fig. 11. Documents-Words Bi-Partite Graph with occurrence of any five words

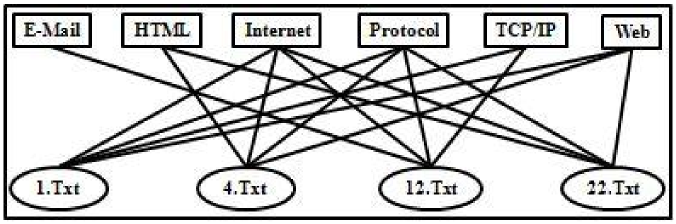


Fig. 12. Documents-Words Bi-Partite Graph with occurrence of any four words

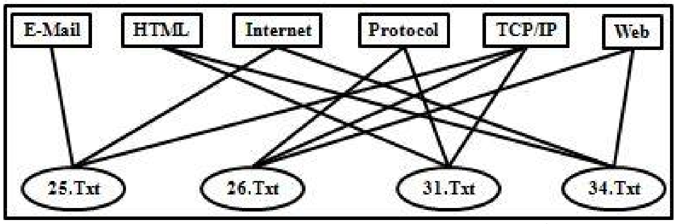


Fig. 13. Documents-Words Bi-Partite Graph with occurrence of any three words

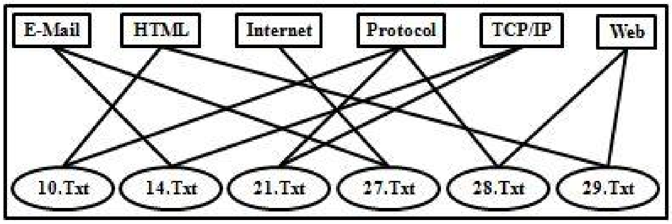


Fig. 14. Documents-Words Bi-Partite Graph with occurrence of any two words

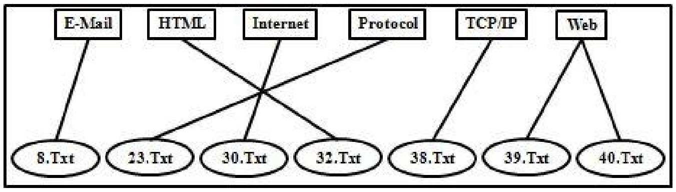
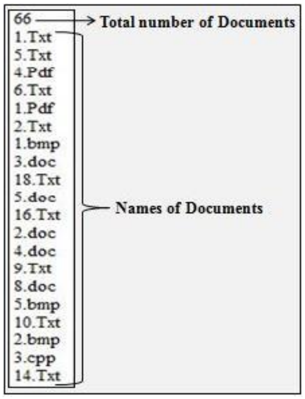


Fig. 15. Documents-Words Bi-Partite Graph with occurrence of any one word

V. Experimental Results

The number of documents and the 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 "Document.Txt" and "Word.Txt" are dataset to the proposed algorithm and depicted in "Fig. 16" and "Fig. 17".

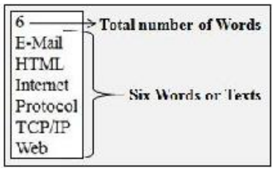
The algorithm was written in C++ programming. It was compiled with TurboC++ and run on Intel Core I5-3230M CPU + 2.60 GHz Laptop with 4GB memory running MS-Windows 7.



```
66
1.Txt
5.Txt
4.Pdf
6.Txt
1.Pdf
2.Txt
1.bmp
3.doc
18.Txt
5.doc
16.Txt
2.doc
4.doc
9.Txt
8.doc
5.bmp
10.Txt
2.bmp
3.cpp
14.Txt
```

Total number of Documents
Names of Documents

Fig. 16. Dataset File Document.Txt

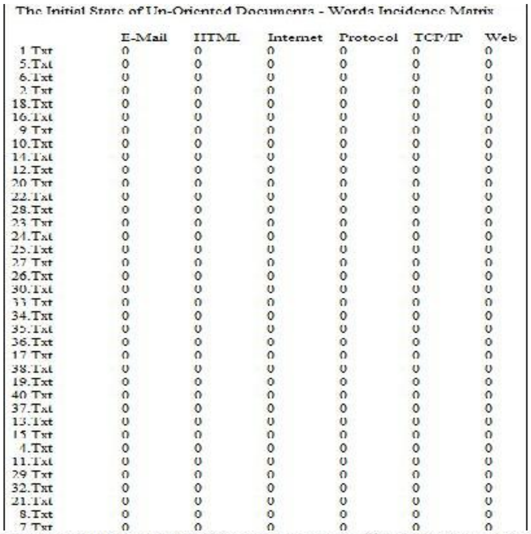


```
6
E-Mail
HTML
Internet
Protocol
TCP/IP
Web
```

Total number of Words
Six Words or Texts

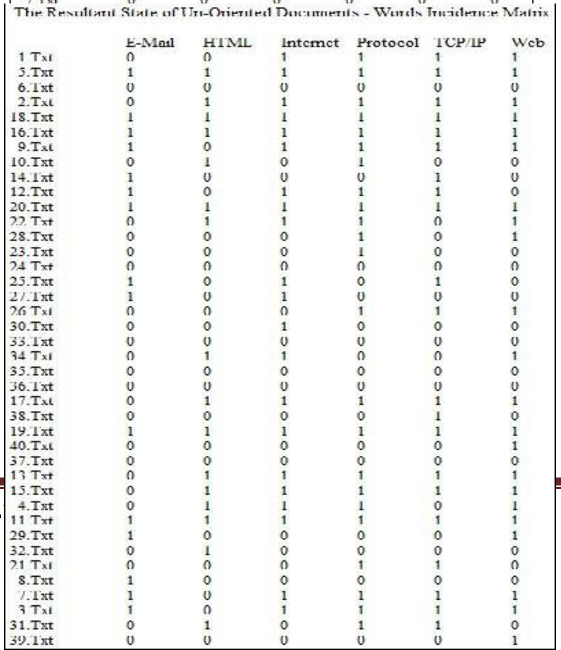
Fig. 17. Dataset File Word.Txt

Fig.18. Input of Document and Word File Name



```
The Initial State of Un-Oriented Documents - Words Incidence Matrix

          E-Mail   HTML   Internet   Protocol   TCP/IP   Web
 1.Txt    0        0      0          0          0        0
 5.Txt    0        0      0          0          0        0
 6.Txt    0        0      0          0          0        0
 2.Txt    0        0      0          0          0        0
18.Txt    0        0      0          0          0        0
16.Txt    0        0      0          0          0        0
 9.Txt    0        0      0          0          0        0
10.Txt    0        0      0          0          0        0
14.Txt    0        0      0          0          0        0
12.Txt    0        0      0          0          0        0
20.Txt    0        0      0          0          0        0
22.Txt    0        0      0          0          0        0
28.Txt    0        0      0          0          0        0
23.Txt    0        0      0          0          0        0
24.Txt    0        0      0          0          0        0
25.Txt    0        0      0          0          0        0
27.Txt    0        0      0          0          0        0
26.Txt    0        0      0          0          0        0
30.Txt    0        0      0          0          0        0
33.Txt    0        0      0          0          0        0
34.Txt    0        0      0          0          0        0
35.Txt    0        0      0          0          0        0
36.Txt    0        0      0          0          0        0
17.Txt    0        0      0          0          0        0
38.Txt    0        0      0          0          0        0
19.Txt    0        0      0          0          0        0
40.Txt    0        0      0          0          0        0
37.Txt    0        0      0          0          0        0
13.Txt    0        0      0          0          0        0
15.Txt    0        0      0          0          0        0
 4.Txt    0        0      0          0          0        0
11.Txt    0        0      0          0          0        0
29.Txt    0        0      0          0          0        0
32.Txt    0        0      0          0          0        0
21.Txt    0        0      0          0          0        0
 8.Txt    0        0      0          0          0        0
 7.Txt    0        0      0          0          0        0
```

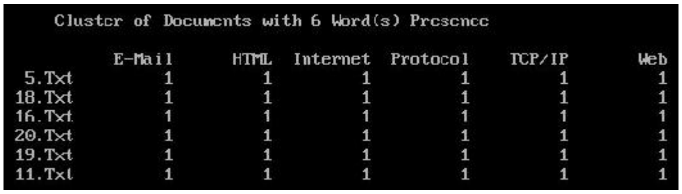


```
The Resultant State of Un-Oriented Documents - Words Incidence Matrix

          E-Mail   HTML   Internet   Protocol   TCP/IP   Web
 1.Txt    0        0      1          1          1        1
 5.Txt    1        1      1          1          1        1
 6.Txt    0        0      0          0          0        0
 2.Txt    0        1      1          1          1        1
18.Txt    1        1      1          1          1        1
16.Txt    1        1      1          1          1        1
 9.Txt    1        0      1          1          1        1
10.Txt    0        1      0          1          0        0
14.Txt    1        0      0          0          1        0
12.Txt    1        0      1          1          1        0
20.Txt    1        1      1          1          1        1
22.Txt    0        1      1          1          0        1
28.Txt    0        0      0          1          0        1
23.Txt    0        0      0          1          0        0
24.Txt    0        0      0          0          0        0
25.Txt    1        0      1          0          1        0
27.Txt    1        0      1          0          0        0
26.Txt    0        0      0          1          1        1
30.Txt    0        0      1          0          0        0
33.Txt    0        0      0          0          0        0
34.Txt    0        1      1          0          0        1
35.Txt    0        0      0          0          0        0
36.Txt    0        0      0          0          0        0
17.Txt    0        1      1          1          1        1
38.Txt    0        0      0          0          1        0
19.Txt    1        1      1          1          1        1
40.Txt    0        0      0          0          0        1
37.Txt    0        0      0          0          0        0
13.Txt    0        1      1          1          1        1
15.Txt    0        1      1          1          1        1
 4.Txt    0        1      1          1          0        1
11.Txt    1        1      1          1          1        1
29.Txt    1        0      0          0          0        1
32.Txt    0        1      0          0          0        0
21.Txt    0        0      0          1          1        0
 8.Txt    1        0      0          0          0        0
 7.Txt    1        0      1          1          1        1
 3.Txt    1        0      1          1          1        1
31.Txt    0        1      0          1          1        0
39.Txt    0        0      0          0          0        1
```

Fig. 19. Initial form of Un-oriented Documents-Words Incidence Matrix

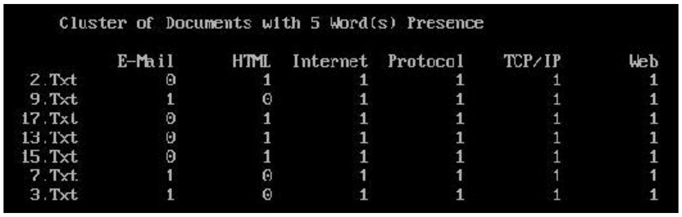
Fig. 20. Resultant State of Un-oriented Documents-Words Incidence Matrix



```
    Cluster of Documents with 6 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
  5.Txt         1         1         1         1          1          1
 18.Txt         1         1         1         1          1          1
 16.Txt         1         1         1         1          1          1
 20.Txt         1         1         1         1          1          1
 19.Txt         1         1         1         1          1          1
 11.Txt         1         1         1         1          1          1
```

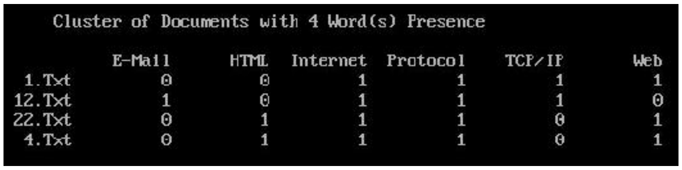
Fig. 21. Un-oriented Documents-Words Incidence Matrix with Six Words Presence



```
    Cluster of Documents with 5 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
  2.Txt         0         1         1         1          1          1
  9.Txt         1         0         1         1          1          1
 17.Txt         0         1         1         1          1          1
 13.Txt         0         1         1         1          1          1
 15.Txt         0         1         1         1          1          1
  7.Txt         1         0         1         1          1          1
  3.Txt         1         0         1         1          1          1
```

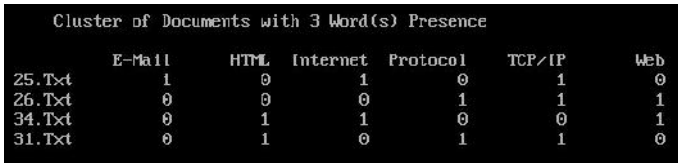
Fig. 22. Un-oriented Documents-Words Incidence Matrix with Five Words Presence



```
    Cluster of Documents with 4 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
  1.Txt         0         0         1         1          1          1
 12.Txt         1         0         1         1          1          0
 22.Txt         0         1         1         1          0          1
  4.Txt         0         1         1         1          0          1
```

Fig. 23. Un-oriented Documents-Words Incidence Matrix with Four Words Presence

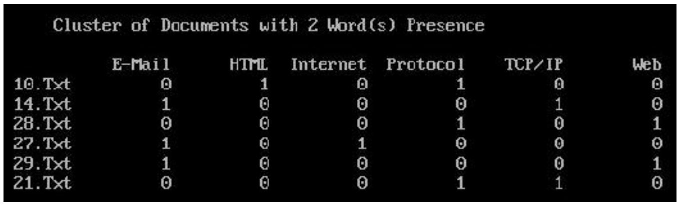


```
    Cluster of Documents with 3 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
 25.Txt         1         0         1         0          1          0
 26.Txt         0         0         0         1          1          1
 34.Txt         0         1         1         0          0          1
 31.Txt         0         1         0         1          1          0
```

Fig. 24. Un-oriented Documents-Words Incidence Matrix with Three Words Presence

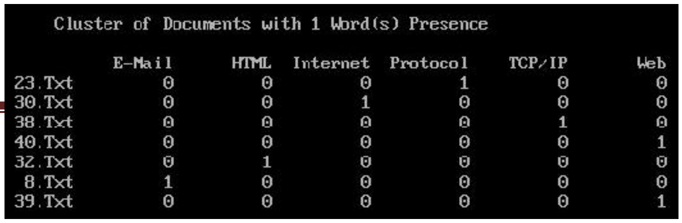
Finally the authors have drawn successfully the bi-partite graphs from un-oriented Documents-Words incidence matrix depicted from "Fig. 21" to "Fig. 26". All the six bi-partite graphs are depicted from "Fig. 10" to "Fig. 15" respectively.



```
    Cluster of Documents with 2 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
 10.Txt         0         1         0         1          0          0
 14.Txt         1         0         0         0          1          0
 28.Txt         0         0         0         1          0          1
 27.Txt         1         0         1         0          0          0
 29.Txt         1         0         0         0          0          1
 21.Txt         0         0         0         1          1          0
```

Fig. 25. Un-oriented Documents-Words Incidence Matrix with Two Words Presence



```
    Cluster of Documents with 1 Word(s) Presence

           E-Mail      HTML  Internet  Protocol     TCP/IP        Web
 23.Txt         0         0         0         1          0          0
 30.Txt         0         0         1         0          0          0
 38.Txt         0         0         0         0          1          0
 40.Txt         0         0         0         0          0          1
 32.Txt         0         1         0         0          0          0
  8.Txt         1         0         0         0          0          0
 39.Txt         0         0         0         0          0          1
```

Fig. 26. Un-oriented Documents-Words Incidence Matrix with One Word Presence

VI. Conclusion

The proposed algorithm using graph mining techniques which clusters of text documents from various types of documents upon inputting *n*-words. The algorithm successfully forms n numbers of cluster having the presence of words *n*, *n-1*, ………, *1* in those clusters of text documents. These *n* numbers of clusters are treated as documents-words graphs which are termed as bi-partite graphs. Finally the algorithm was implemented using C++ programming language and observed satisfactory results.

References

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

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

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

[4] E. Horowitz, S. Sahani, & D. Mehta, Fundamentals of Data Structures in C++ (2nd Edition), University Press (India) Private Limited, Himayat Nagar, Hyderabad, AP-500029, India, 2013.

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

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

[7] N. Slinim & N. Tishby, "Document Clustering Using Word Clusters via the Information Bottleneck Method", 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, Pp. 208-215, 2000.

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

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