“ANALYSIS OF TEXT DOCUMENT SUMMARIZATION USING NEAREST NEIGHBOUR AND AGGLOMERATIVE CLUSTERING”

[1] Ganesh Jorvekar, [2] Manish Rai, [3] Dr. Mohit Gangwar
[1]Student, [2]Head of Department , [3]Principal
[1]Computer Science and Engineering,
[1]Sanjivani K.B.P. Polytechnic, Kopargaon, India

Abstract- World Wide Web is a huge collection of data of different file formats. With the coming of the information revolution, electronic documents are becoming a principle media of business and academic information. In order to fully utilize these on-line documents effectively, it is crucial to be able to extract the gist of these documents. It is not the case that a particular clustering algorithm is best suited for clustering of documents of different file formats.Having a Text Summarization system would thus be immensely useful in serving this need. In order to generate a summary, we have to identify the most important pieces of information from the document, omitting irrelevant information and minimizing details, and assembling them into a compact Coherent report. A particular Clustering algorithm is best suited for query dependent text document summarization. As every document we can convert into text, this strategy is much needful for the end users. The conclusion is drawn by using and comparing two different clustering algorithm namely Nearest Neighbour And Agglomerative Hierarchical Clustering Algorithm.

Keywords-Summarization, Clustering, Query Dependent, Text Summarization, Nearest Neighbour, Agglomerative Hierarchical

I. INTRODUCTION

The present research work focuses on analytical study of different document clustering and summarization techniques currently the most research is focused on Query-Independent summarization. The main aim of this research work is to combine both approaches of document clustering and query dependent summarization. This mainly includes applying different clustering algorithms on a text document. Create a weighted document graph of the resulting graph based on the keywords. And obtain the optimal tree to get the summary of the document. The performance of the summary using different clustering techniques will be analyzed and the optimal approach will be suggested [1,2,5].

As we know WWW is the largest source of Information and huge amount of data is available on web. So to get accurate data or Relevant Data from the raw data is the challenging task in today’s era. By using different Clustering Algorithms we can retrieve the relevant information for a specific query. As we know there are different clustering algorithms for grouping of similar data element. In this approach we are using two different

Clustering algorithm for retrieving the information and we compare them with respect to space and time complexity [5]. Document understanding techniques such as document summarization have been receiving much attention these years. Current document clustering methods usually represent documents as a term document matrix and perform clustering algorithm on it. Although these clustering methods can group the documents satisfactorily, it is still hard for people to capture the meanings of the documents since there is no satisfactory interpretation for each document cluster.

I. PROBLEM DEFINATION

“To find best suited query dependent clustering algorithm for text document summarization “

The purpose of this project is to suggest better query dependent clustering algorithm for text document summarization. Our present task aims at developing a query dependant single-document summarizer using Nearest Neighbour clustering and Agglomerative Hierarchical clustering techniques. We hope it will add another dimension towards solving the seemingly complex task of document summarization and presentation of the gist of documents [15].

Goal: The goal is to use nearest neighbor, agglomerative hierarchical clustering algorithm for query dependent clustering of nodes in text document, and finding query dependent summary. The summary will be compared and best algorithm will be suggested for query dependent clustering using different clustering techniques. This technique will help end users to prepare query dependent summary of text document s in which they are interested[15].

- The proposed work will be mainly focused on summarization of text files (i.e. .txt)

- The proposed work will be limited to clustering of text files of standard files related to the topic popular amongst researchers will be used.
- Only hierarchical clustering and nearest neighbor method clustering are considered for generating cluster based graph.
- Standard performance evaluation metrics will be used to validate performance.

What is a summary?
"An abbreviated, accurate representation of the content of a document preferably prepared by its author(s) for publication with it." Such abstracts are also useful in access publications and machine-readable databases (American National Standards Institute Inc., 1979).[15]
The process of producing a summary from a source
text consists of
the following steps:

1. The interpretation of the text;
2. The extraction of the relevant information which ideally includes the "topics" of the source;
3. The condensation of the extracted information and construction of a summary representation;
4. The presentation of the summary representation to the reader in natural language.

While some techniques exist for producing summaries for domain independent texts (Luhn, 1958; Marcu, 1997) it seems that domain specific texts require domain specific techniques [1] (DeJong, 1982; Paice and Jones, 1993). In order to address the issue of topic identification, content selection and presentation, we have studied alignments (manually produced) of sentences from professional abstracts with sentences from source documents.

II. SYSTEM DESIGN

A specialist has to check for the dataflow and have to manually where the data flows. Analysts have proved that it would take more time for an experienced specialist to note the dataflow.
We are accepting text file only. Newline contents are forming a node, hence a single cluster. If there is no newline content then only one node will be there, hence only on cluster. This will degrade the performance of the algorithms, as the cluster size is very big.

Software Architecture

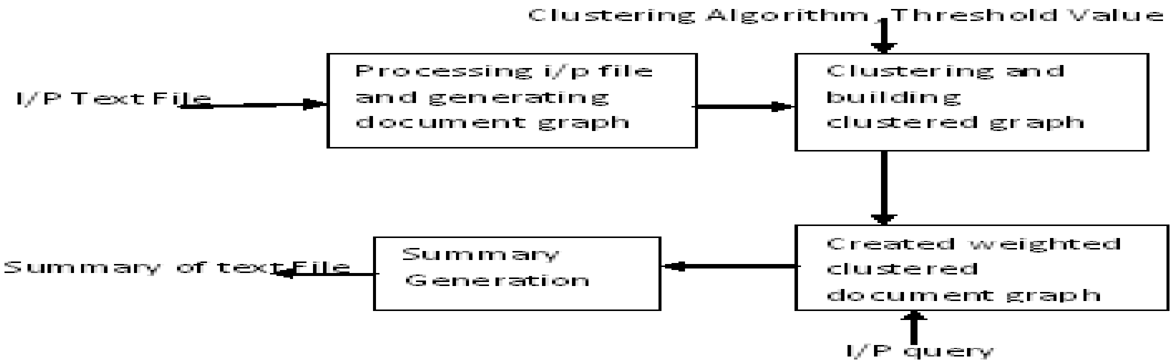

Clustering Algorithm, Threshold Value
I/P Text File
Processing i/p file and generating document graph
Clustering and building clustered graph
Summary of text File
Summary Generation
Created weighted clustered document graph
I/P query

Figure 1 Architecture Diagram

Figure shows the architecture diagram of the system. As shown in figure there are four main blocks : a block for uploading and processing text file and making document graph, a block for clustering and making clustered graph, a block for making weighted clustered document graph., the last block for generating summary for fired query.

Block 1: Processing input file and generating document graph:
This block is needed to accept the text file only. It is responsible to upload text file, to process the file i.e. to form nodes for every newline contents. It is also responsible for generating weight from each node to very other node

Block 2: Clustering node and building clustered graph:
This block is responsible for choosing a clustering algorithm out of two. It also accepts the threshold, so that can check the similarity between the clusters up to that level. It is responsible for making clusters.[11,13]
Block 3: Creating weighted document clustered graph:
This block is responsible to accept the fired query. It is responsible to check the similarities between the query a contents and the contents in the clusters. It then build weighted clustered document graph[15].

Block 4.Summary generation:

This block is responsible for generating the summary of the clusters we formed, as a response for fired query. It generated the minimal clusters and after finding the weight of the node for fired query, it gives top most summaries.

III. COMPARISION

Comparison of different techniques used for text document summarization [16].

Type of summarization methods	Subtype	Concept	Advantages	Disadvantages	Application/Work Done
1.Approaches Figures	Abstractive	It is the process of reducing a text document in order to create a summary that is semantically related	Good compression ratio. More reduced text and semantically related summary	Difficult to compute	SUMMRIST
	Extractive	It consists of selecting important sentences from original document based on statistical features	Easy to compute because it does not deal with the semantics and more successful	Suffers from inconsistencies, lack of balance, results in lengthy summary	Summ-It applet,designed by Surrey University
2.Details	Indicative	It only presents main idea of text to user. They can be used to quickly decide whether a text is worth reading	Encourages the users to read the main document in depth. Used for quick categorization and easier to produce	Detailed information is not present	Information present on the back of the movie pack or novels Length 5 to 10%
	Informative	Gives concise information of the main text	Serves as a substitution for the main document	Does not provide quick overview	SumUM Length 20 to 30%
3.Content	Generic	Generalized summary irrespective of the type of user. Information is at same level of importance	Can be used by any type of user	It provides an author's view not user specific	SUMMARIST
	Query based	User has to determine the topic of original text in the form of query and system only extract that information	Specific information can be searched. It reflects user’s interest	Not used by any type of user. It is based on type of user	Mitre's WebSumm
	Domai	Summarize the text which their	They are aware of the special	Limited to the	

4.Limitation	n dependent	subject can be defined in the fixed domain	domain on which they are dependent	subject of the document	TRESTLE
	Genre specific	Accept only special type of text as input.	Overcomes the problem of summarizing heterogeneous document	Limitation on template of the text	Newsblaster
	Domain Independent	Can accept any type of text.	Any type of text input is accepted. It is not domain dependent	Difficult to implement	Copy and Paste system
5.Number of input document	**Single document**	**Can accept only one input document**	**Less overhead**	**Cannot summarize multiple documents of related topics**	**Copy and paste system**
	Multi-document	Can accept multiple input documents	Multiple documents of same topic can be summarized to single document	Difficult to implement	SUMMONS Designed by Columbia university
6.Language	Mono-Lingual	Can accept input only with specific language and output is based on that language	Need to work with only one language	Cannot handle different language	FarsiSum
	Multi-Lingual	Can accept documents in different language	Can deal with multiple language	Difficult to implement.	SUMMARIST(English, Japanese,Spanish)

IV. IMPLEMENTATION

System Implementation

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users, which it will work efficiently and effectively. It involves careful planning, investigation of the current System and its constraints on implementation, design of methods to achieve the change over, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.[14]

Implementation is very important phase, the most critical stage in achieving a successful new system and in giving the users confidence. That the new system will work is effective. After the system is implemented the testing can be done. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.[5,11,13]

The input is a text file contains new line keyword. The contents are separated by new line are the contents of the node which are the paragraph. If there are no new line in the file, then whole file contents becomes a single node and hence a single cluster, which can degrade the performance of the result.

The total workflow is divided into following modules:

Module 1: Processing the input text file and creating the document graph

Functions Used:

Split ()

The system accepts input text file. The file is read and stored into a string. The string is then split by the newline keyword. The split file is assigned to the string array as the split function returns the string array. The array contains paragraphs which are further treated as nodes.

```
            string [] nodeList = null;
NodeList = File.ReadAllLines (txtInputFile.Text);
```

The next stage is to find the similarity between the nodes that means finding the similarity edges between nodes and finding their similarity or weight.

Each paragraph becomes a node in the document graph.

The document graph G (V, E) of a document d is defined as follows:

d is split to a set of non-overlapping nodes *t (v), v € V*.

An edge *e (u, v) € E* is added between nodes *u, v € V* if there is an association between *t (u)* and *t (v)* in d.

Hence, we can view G as an equivalent representation of d, where the associations between text fragments of d are depicted.

Module 2: Adding Weighted Edges to Document Graph
(Note: Adding weighted edge is query independent)

A weighted edge is added to the document graph between two nodes if they either correspond to adjacent node or if they are semantically related, and the weight of an edge denotes the degree of the relationship. Here two nodes are considered to be related if they share common words (not stop words) and the degree of relationship is calculated by “Semantic parsing”. Also notice that the edge weights are query-independent, so they can be pre-computed.[3,7,9]

The following input parameters are required at the pre computation stage to create the document graph:

1. Threshold for edge weights: Only edges with weight not below threshold will be created in the document graph. (A threshold is user configurable value that controls the formation of edges)

Adding weighted edge is the next step after generating document graph. Here for each pair of nodes u, v we compute the association degree between them, that is, the score (weight) EScore (e) of the edge e (u, v). If Score (e) ≥threshold, then e is added to E. The score of edge e (u, v) where nodes u, v have text fragments t(u), t(v) respectively is: [1]

$$\mathrm{EScore} = \frac{\sum ((tf(t(u),w) + tf(t(v),w)) .idf(w)))}{\mathrm{size}(t(u)) + \mathrm{size}(t(v))}$$

Where t f (d, w) is the number of occurrences of w in d,

id f (w) is the inverse of the number of documents containing w, and

size(d) is the size of the document (in words).That is, for every word w appearing in both text fragments we add a quantity equal to the tf□idf score of w. Notice that stop words are ignored.

Functions Used:

Remove Common Words ()

The common words are eliminated from the nodes as they can degrade the performance of calculating the similarity between two nodes also they can degrade the system performance because of number of computational loops increases. E.g. a,an,the,he,she,they,as,it,and,are,were,there etc.

The filtered two nodes are passed as parameters to the Relation Manager Class for finding the similarity between them.

Table 1. Nodes and Node weights

First_Node	Second_Node	Edge Weight
1	2	0.5
1	3	0.7
. .	. .	. .
30	31	0.8
30	32	0.6

Module 3: Document Clustering

Clustering is grouping of similar nodes(The nodes which shows degree of closure greater than or equal to the Cluster Threshold specified by the user) into a group. The following approaches of clustering are used

a) Nearest Neighbour
b) Hierarchical Agglomerative

Algorithm for Nearest Neighbour Clustering:

1. Set i = 1 and k = 1. Assign pattern x_1 to cluster C_1.
2. Set i = i + 1. Find nearest neighbor of x_i among the patterns already assigned to clusters. Let d_m denote the distance from x_i to its nearest neighbor. Suppose the nearest neighbor is in cluster m.
3. If d_m greater than or equal to t then assign x_i to C_m where t is the threshold specified by the user. Otherwise set k = k+1 and assign x_i to a new cluster C_k.
4. If every pattern has been considered then stop else go to step 2.

Algorithm for Agglomerative Hierarchical Clustering:

1) Start by assigning each node to a cluster, so that if you have N nodes, you now have N clusters, each containing just one node. Let the distances (similarities) between the clusters the same as the distances (similarities) between the nodes they contain.
2) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3) Compute distances (similarities) between the new cluster and each of the old clusters.
4) Repeat steps 2 and 3 until the maximum number of nodes in the cluster (User definable value) is reached.

Functions Used:

FindMaxWeight ()

FindMaxWeight returns the pair of nodes having maximum edge weight with their weight from document graph. e.g.

Table 2.Nodes and the max weight

First Node	Second Node	Max Weight
1	22	2.5
2	19	1.2
3	31	3.5
. .	. .	. .
31	12	2.7

NearestNeighborCluster ()

1. The first pair of nodes in the above table is added in first Cluster because they have maximum weight. Here Node 1 and 22 are closely related hence added to the first cluster. So Cluster_1 contains 2 nodes 1 and 22. Cluster_1 :- 1,22
2. Next node – node 2 shows maximum weight with node 19 but none of the node (node 2 and node 19)are in previous clusters so they forms new cluster – Cluster_2 Cluster_2:- 2, 19
3. Similarly Node 3 and 13 are forming new cluster. Cluster_3 :- 3,31
4. Now next pair (node 31 and 12) contains node 31 which is already in cluster_3 hence node 12 is added into cluster_3, so cluster_3 now becomes Cluster_3:-3, 31, 12.
5. The above procedure is repeated till the end of the node pairs.

Hierarchical Clustering ()

1) Every node is considered as a cluster.
2) Start by the first pair of nodes in the above table and add in the first cluster because they have maximum weight. Here Node 1 and 22 are closely related hence added to the first cluster. So Cluster_1 contains 2 nodes 1 and 22. Cluster_1 :- 1,22
3) The newly formed cluster Cluster_1 is compared with all other clusters (Nodes are clusters). Here the Cluster_1 and every other cluster are sent to the relation manager and maximum weight is calculated. The cluster which is having maximum weight greater than or equal to Cluster Threshold (Specified by user) is added into the first cluster Cluster_1. If the value is less than Cluster Threshold Value then that cluster is kept as it is.
4) The step 3 is repeated till all the clusters are processed.

Table 3.Cluster number and nodes in cluster

Cluster Number	**Nodes**
Cluster_1	1,22,13,25
Cluster_2	2,9,13,20

Module 4: Creating Clustered Document Graph

After the clusters are formed either by Nearest Neighbour or agglomerative hierarchical, the similarity edges between two similar clusters are calculated. This is same as creating document graph and adding the similarity edges between two similar nodes. Every cluster is split into individual nodes and this grouping of nodes is passed to the relation manager in order to find the weight between two set of nodes or Clusters.[5,7,10]

Adding Weight to Nodes In Clustered Document Graph

When a query *Q* arrives, the nodes in *V* are assigned query-dependent weights according to their relevance to *Q*. In particular, we assign to each node *v* corresponding to a text fragment *t(v)* node score *NScore(v)* defined by the Okapi formula as given below.[1]

$$\mathbf{NScore\ (v)} = \sum_{t\in Q.d} \ln\frac{N-df+0.5}{df+0.5}\cdot\frac{(k_1+1)tf}{(k_1(1-b)+b\frac{dl}{avdl})+tf}\cdot\frac{(k_3+1)qtf}{k_3+qtf}$$

tf is the term’s frequency in document,
qtf is the term’s frequency in query,
N is the total number of documents in the collection,
df is the number of documents that contain the term,
dl is the document length (in words),
avdl is the average document length and
k1 (between 1.0–2.0), b (usually 0.75), and k3 (between 0–1000) are constants.

Functions Used:

CalculateClusterWeight ()
All the values mentioned above are computed and passed as parameters to the okapi formula.
The returned Node Weight is stored in the table. e.g.[1]

Table 4.Cluster no and weight of cluster

Cluster No	**Nodes**	**Cluster Weight**
Cluster_1	1,22,13,32	2.4
Cluster_2	9,17,24	2.5
Cluster_3	34,12,10	0
Cluster_4	4,14,23	0

Module 5: Generating Closure Graph and Finding Minimal Spanning Tree.

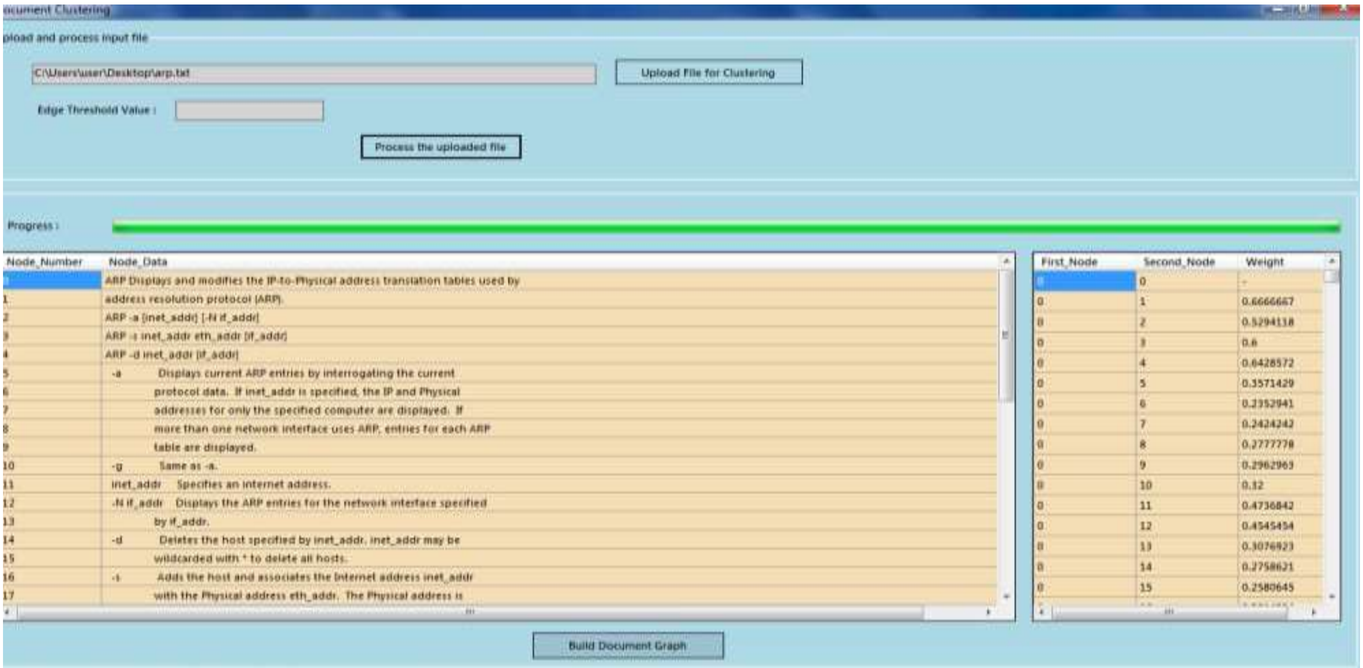
Closure graph contains minimal clusters. Minimal clusters are the clusters which shows non zero weight with the in out query. In Above example (Tab 3) only Cluster_1 and Cluster_2 are the minimal clusters. The minimal clusters are the clusters which appear in the result.

Module 6: Result

After getting the minimal clusters, the result can be displayed in two ways:
Top 1 Result Summary
Multi-Result Summary
In top 1 result summary, the minimal cluster having highest weight with the input query is returned, and in multi-result summary all the minimal clusters are returned as result.

V. RESULT

Initially user needs to give text file as input. Then he will click on 'process the input file'. Every newline contains will form a single node and description of the node data will be displayed. Along with this the weight between each node to every other node is calculated. The weight of a node with itself is null. This process is demonstrated by following examples.


ocument Clustering
pload and process input file
Upload File for Clustering
Process the uploaded file
Progress :
Build Document Graph

C:\Users\user\Desktop\arp.txt

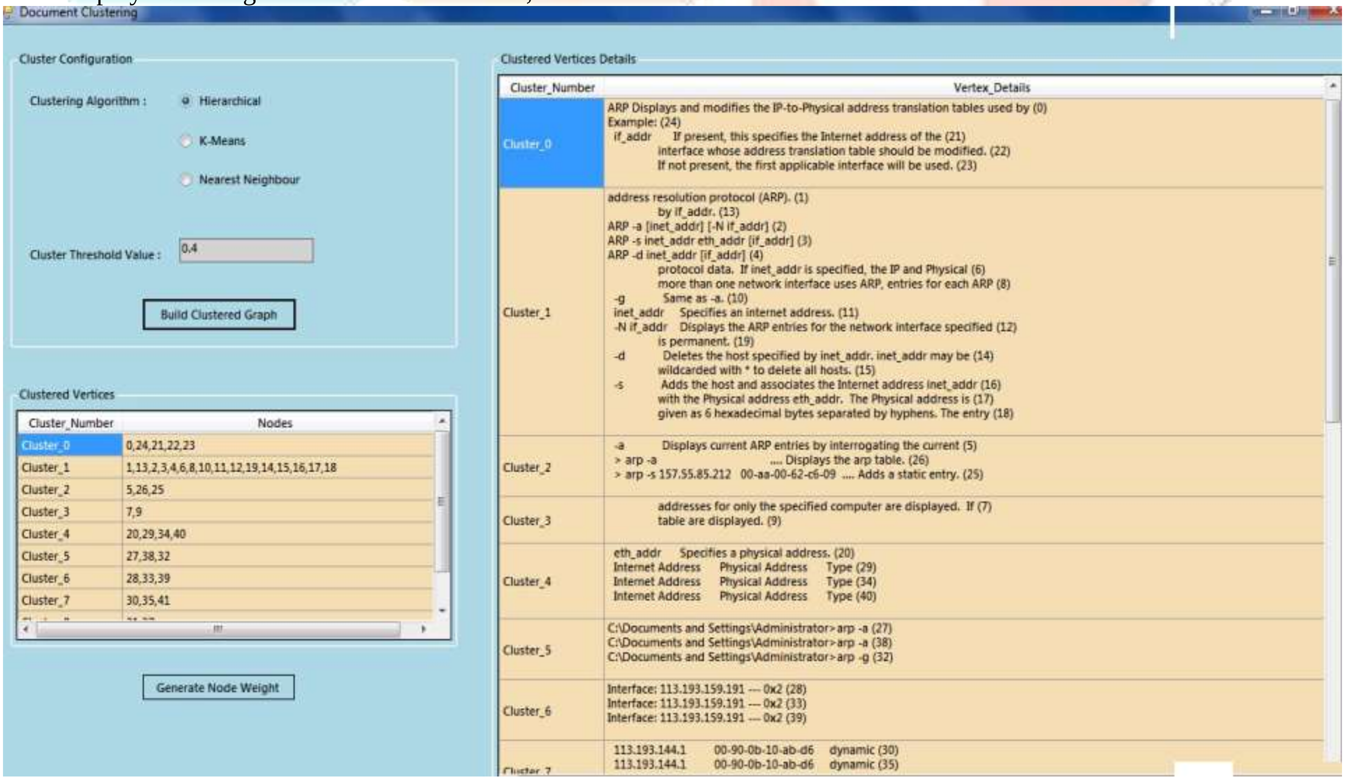
Edge Threshold Value :

Node_Number	Node_Data
0	ARP Displays and modifies the IP-to-Physical address translation tables used by
1	address resolution protocol (ARP).
2	ARP -a [inet_addr] [-N if_addr]
3	ARP -s inet_addr eth_addr [if_addr]
4	ARP -d inet_addr [if_addr]
5	-a Displays current ARP entries by interrogating the current
6	protocol data. If inet_addr is specified, the IP and Physical
7	addresses for only the specified computer are displayed. If
8	more than one network interface uses ARP, entries for each ARP
9	table are displayed.
10	-g Same as -a.
11	inet_addr Specifies an internet address.
12	-N if_addr Displays the ARP entries for the network interface specified
13	by if_addr.
14	-d Deletes the host specified by inet_addr. inet_addr may be
15	wildcarded with * to delete all hosts.
16	-s Adds the host and associates the Internet address inet_addr
17	with the Physical address eth_addr. The Physical address is

First_Node	Second_Node	Weight
0	0	-
0	1	0.6666667
0	2	0.5294118
0	3	0.6
0	4	0.6428572
0	5	0.3571429
0	6	0.2352941
0	7	0.2424242
0	8	0.2777778
0	9	0.2962963
0	10	0.32
0	11	0.4736842
0	12	0.4545454
0	13	0.3076923
0	14	0.2758621
0	15	0.2580645

Figure2:-Output screen of uploaded text file, processed file, document graph

In this step a clustering algorithm is selected .Threshold for clustering is selected, then the cluster no. and the nodes in that cluster will be displayed. If we give lower threshold value, no of clusters will decrease and the size of cluster will increase.



Clustering Algorithm : ● Hierarchical ○ K-Means ○ Nearest Neighbour

Cluster Threshold Value : 0.4

Cluster_Number	Nodes
Cluster_0	0,24,21,22,23
Cluster_1	1,13,2,3,4,6,8,10,11,12,19,14,15,16,17,18
Cluster_2	5,26,25
Cluster_3	7,9
Cluster_4	20,29,34,40
Cluster_5	27,38,32
Cluster_6	28,33,39
Cluster_7	30,35,41

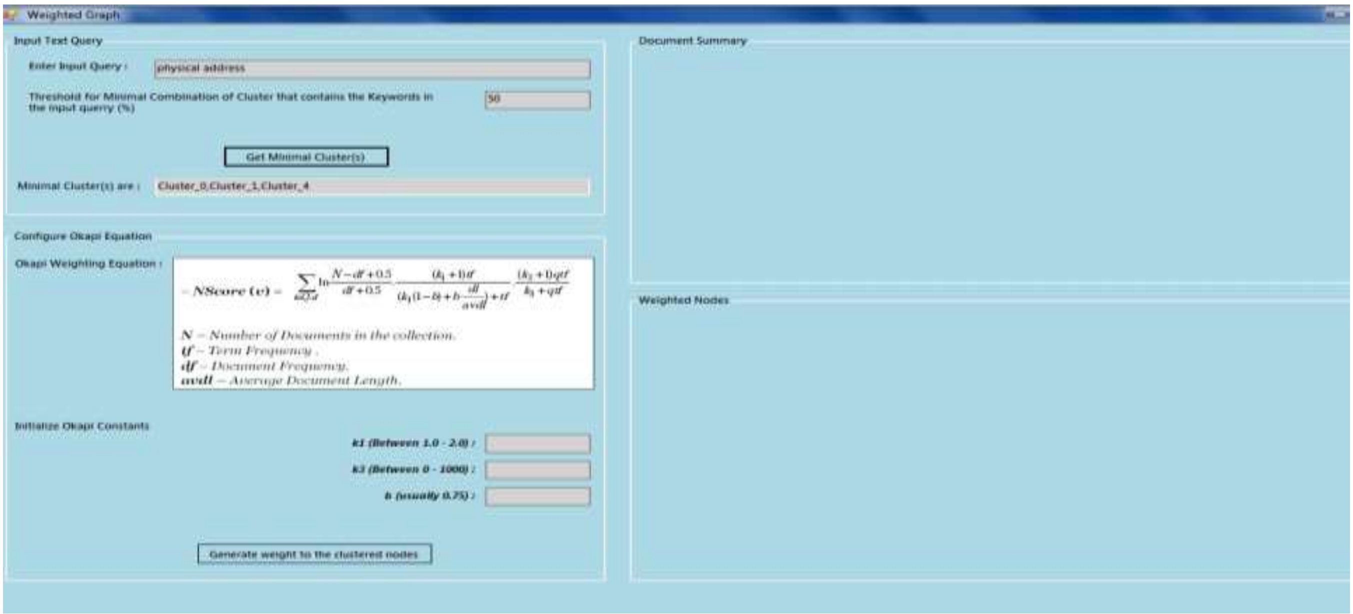
Cluster_Number	Vertex_Details
Cluster_0	ARP Displays and modifies the IP-to-Physical address translation tables used by (0) Example: (24) if_addr If present, this specifies the Internet address of the (21) interface whose address translation table should be modified. (22) If not present, the first applicable interface will be used. (23)
Cluster_1	address resolution protocol (ARP). (1) by if_addr. (13) ARP -a [inet_addr] [-N if_addr] (2) ARP -s inet_addr eth_addr [if_addr] (3) ARP -d inet_addr [if_addr] (4) protocol data. If inet_addr is specified, the IP and Physical (6) more than one network interface uses ARP, entries for each ARP (8) -g Same as -a. (10) inet_addr Specifies an internet address. (11) -N if_addr Displays the ARP entries for the network interface specified (12) is permanent. (19) -d Deletes the host specified by inet_addr. inet_addr may be (14) wildcarded with * to delete all hosts. (15) -s Adds the host and associates the Internet address inet_addr (16) with the Physical address eth_addr. The Physical address is (17) given as 6 hexadecimal bytes separated by hyphens. The entry (18)
Cluster_2	-a Displays current ARP entries by interrogating the current (5) > arp -a .... Displays the arp table. (26) > arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry. (25)
Cluster_3	addresses for only the specified computer are displayed. If (7) table are displayed. (9)
Cluster_4	eth_addr Specifies a physical address. (20) Internet Address Physical Address Type (29) Internet Address Physical Address Type (34) Internet Address Physical Address Type (40)

Cluster_5	C:\Documents and Settings\Administrator>arp -a (27)
	C:\Documents and Settings\Administrator>arp -a (38)
	C:\Documents and Settings\Administrator>arp -g (32)

Cluster_6	Interface: 113.193.159.191 --- 0x2 (28) Interface: 113.193.159.191 --- 0x2 (33) Interface: 113.193.159.191 --- 0x2 (39)
Cluster_7	113.193.144.1 00-90-0b-10-ab-d6 dynamic (30) 113.193.144.1 00-90-0b-10-ab-d6 dynamic (35)

Figure 3:- Output screen of clustered document graph.

After clustering end user will fire the query and give % to correlate the cluster with fired query. The clusters containing the part of query called as minimal cluster will b displayed.



Input Text Query

Enter Input Query : physical address

Threshold for Minimal Combination of Cluster that contains the Keywords in the input querry (%) 50

Minimal Cluster(s) are : Cluster_0,Cluster_1,Cluster_4

Configure Okapi Equation

Okapi Weighting Equation :

$$NScore(v) = \sum_{t \in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

N – Number of Documents in the collection.
tf – Term Frequency .
df – Document Frequency.
avdl – Average Document Length.

Initialize Okapi Constants

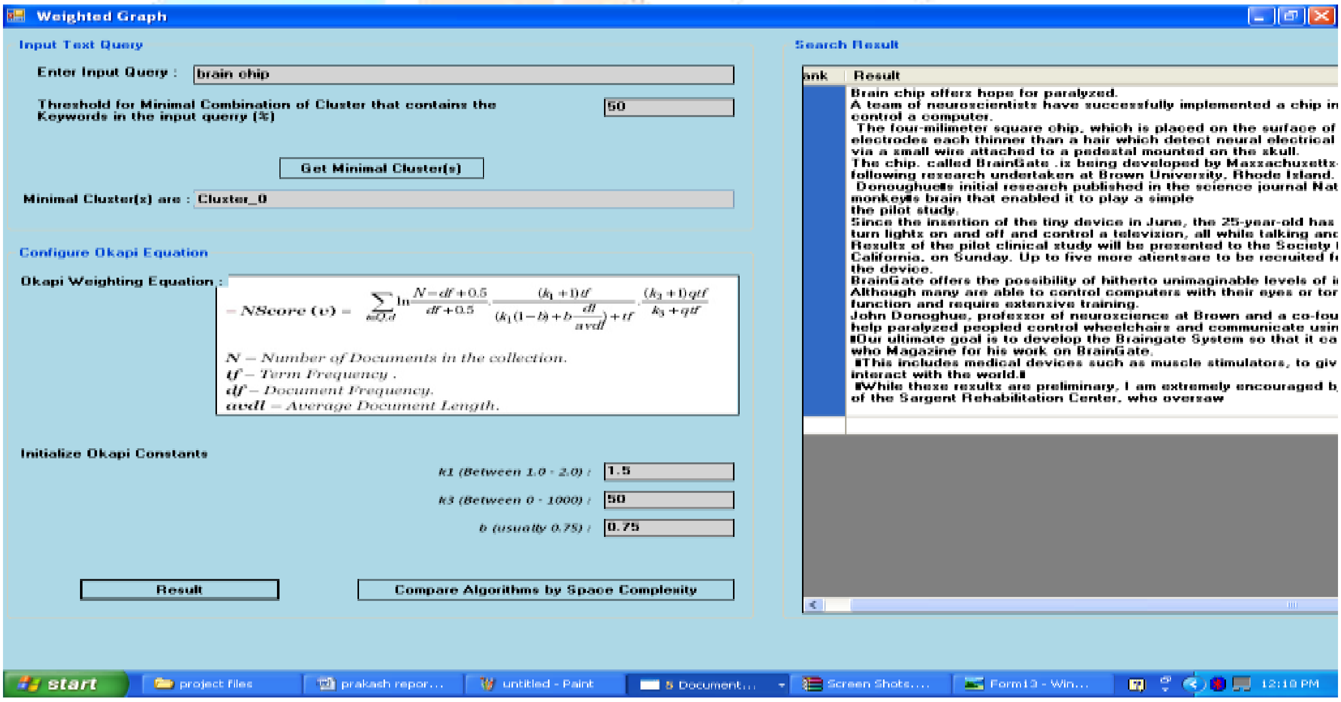
k1 (Between 1.0 - 2.0) :

k3 (Between 0 - 1000) :

b (usually 0.75) :

Figure 4 :-Output screen of getting minimal clusters after arrival of query

.



Input Text Query

Enter Input Query : brain chip

Threshold for Minimal Combination of Cluster that contains the Keywords in the input querry (%) 50

Minimal Cluster(s) are : Cluster_0

Configure Okapi Equation

Okapi Weighting Equation :

$$NScore(v) = \sum_{t \in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

N – Number of Documents in the collection.
tf – Term Frequency .
df – Document Frequency.
avdl – Average Document Length.

Initialize Okapi Constants

k1 (Between 1.0 - 2.0) : 1.5

k3 (Between 0 - 1000) : 50

b (usually 0.75) : 0.75

Search Result

ank Result

Brain chip offers hope for paralyzed.
A team of neuroscientists have successfully implemented a chip in
control a computer.
The four-milimeter square chip, which is placed on the surface of
electrodes each thinner than a hair which detect neural electrical
via a small wire attached to a pedestal mounted on the skull.
The chip, called BrainGate ,is being developed by Massachusetts-
following research undertaken at Brown University, Rhode Island.
Donoughue■s initial research published in the science journal Nat
monkey■s brain that enabled it to play a simple
the pilot study.
Since the insertion of the tiny device in June, the 25-year-old has
turn lights on and off and control a television, all while talking and
Results of the pilot clinical study will be presented to the Society
California, on Sunday. Up to five more atientsare to be recruited f
the device.
BrainGate offers the possibility of hitherto unimaginable levels of i
Although many are able to control computers with their eyes or tor
function and require extensive training.
John Donoghue, professor of neuroscience at Brown and a co-fou
help paralyzed peopled control wheelchairs and communicate usin
■Our ultimate goal is to develop the Braingate System so that it ca
who Magazine for his work on BrainGate.
■This includes medical devices such as muscle stimulators, to giv
interact with the world.■
■While these results are preliminary, I am extremely encouraged b
of the Sargent Rehabilitation Center, who oversaw

Figure 5:-Output screen of clustered summary result

Now the summary we have got should be compared to suggest better algorithm out of two for this work .So we have checked the results on the basis of space complexity, time for execution, length of summary. For this purpose we have taken some special files downloaded from www.scibd.com,www.4shared.com, www.docstoc.com.these are popular websites where current affairs file in every format is available.

Comparison of Nearest Neighbour and Agglomerative Hierarchical Clustering algorithm in terms of Space Complexity

Table5:- Space Complexity

File Name(size)/Clustering algorithm	Nearest Neighbour clustering	Agglomerative Hierarchical Clustering

World Economics(25 kb)	133	700
Global Computing (70 kb)	1417	15674
Brain Chip Research(4 kb)	357	231
Engineering(38 kb)	1321	17118
Stock Market(51 kb)	1556	39097
Nuclear Test(41 kb)	316	99856

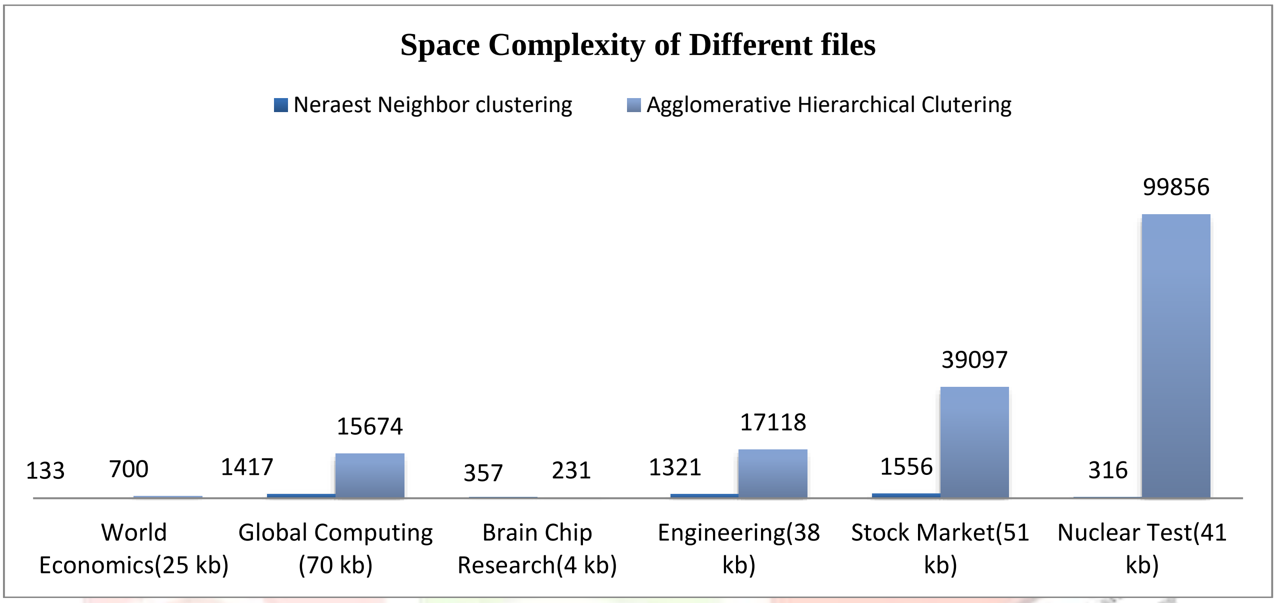


Figure 6:-Chart for Comparing space complexity

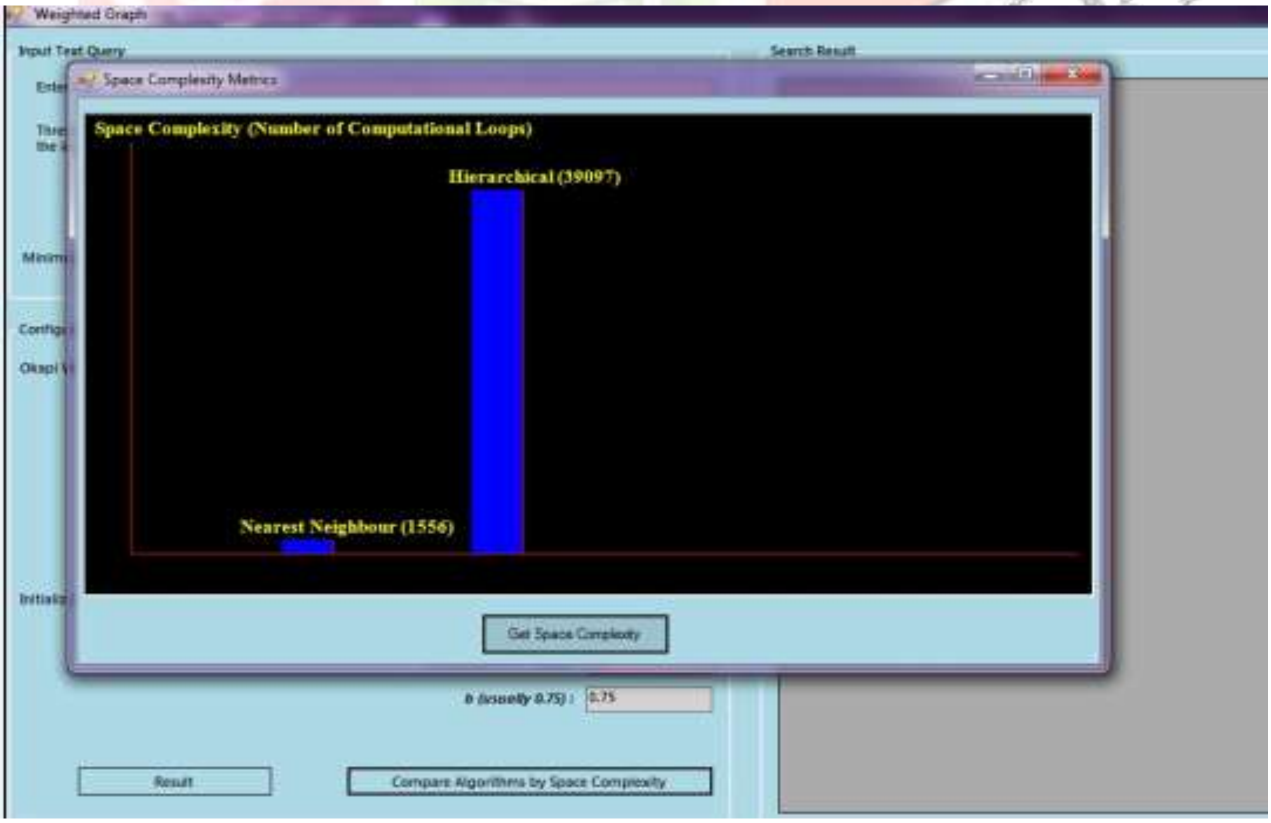


Figure 7:-Comparison of space complexity for a file.

Comparison of Nearest Neighbour and Agglomerative Hierarchical Clustering algorithm in terms of Time Complexity

Table 7:- Time for execution (in seconds)

File Name(size)/Clustering algorithm	Nearest Neighbour clustering	Agglomerative Hierarchical Clustering
World Economics(25 kb)	5	8
Global Computing (70 kb)	45	87
Brain Chip Research(4kb)	2	4
Engineering(38 kb)	11	13
Stock Market(51 kb)	22	80
Nuclear Test(41 kb)	15	22

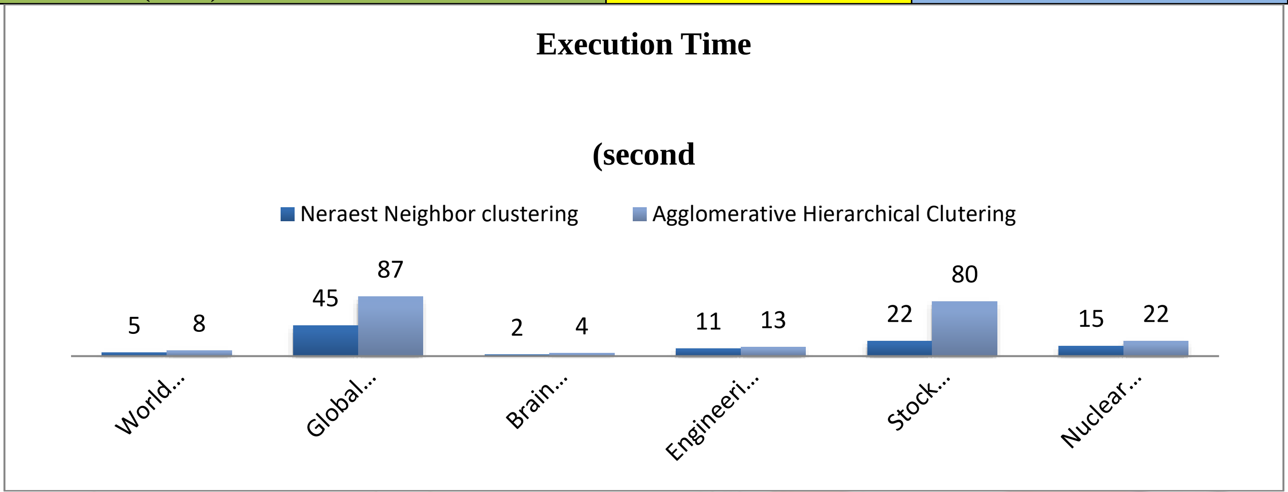


Figure 8:- Chart for comparing execution time

If we observe the table or graph for execution time of those files, it is observed that the execution time required for nearest neighboring clustering algorithm is less than agglomerative hierarchical clustering algorithm.

Comparison of Nearest Neighbour and Agglomerative Hierarchical Clustering algorithm in terms of Length of Summary

Table8:- Length of summary (in no of words)

File Name(size)/Clustering algorithm	Nearest Neighbour clustering	Agglomerative Hierarchical Clustering
World Economics(25 kb)	3425	3467
Global Computing (70 kb)	6628	11348
Brain Chip Research(4 kb)	457	483
Engineering(38 kb)	5324	5324
Stock Market(51 kb)	6146	12354
Nuclear Test(41 kb)	5325	5325

If we observe the table or graph for length of the summary of those files, it is observed that mostly length of the summary for nearest neighbouring clustering algorithm is less than agglomerative hierarchical clustering algorithm.

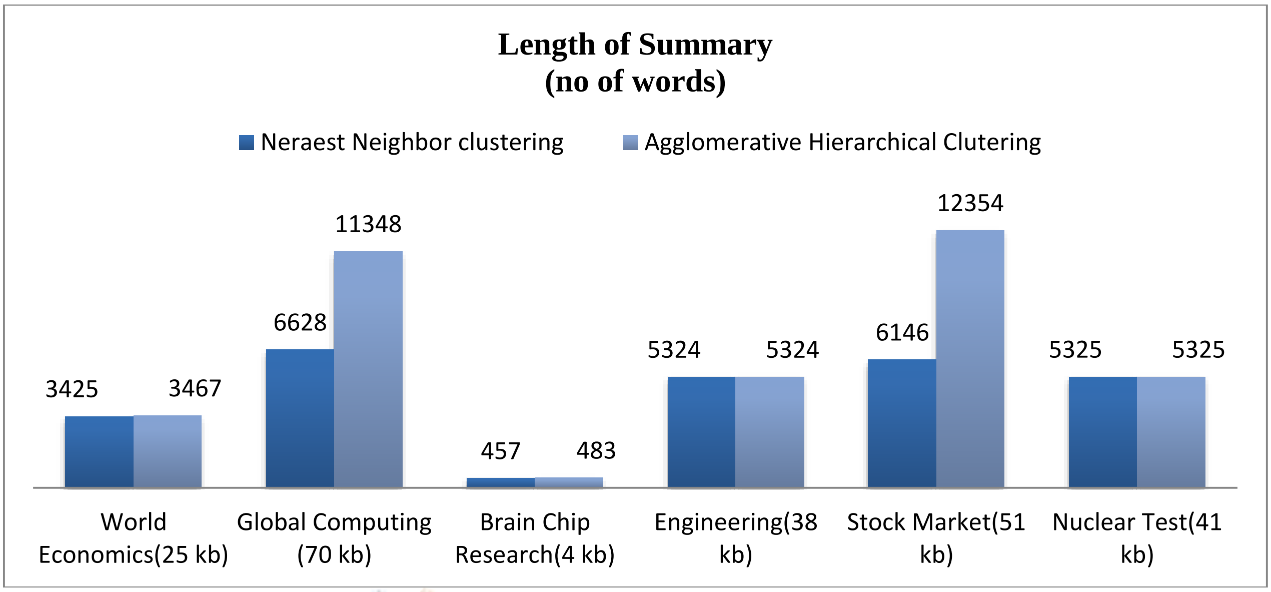


Figure 9:- Chart for comparing length of summary

VI. FUTURE WORK

In this system only two clustering algorithms are used to decide which one is giving better performance. Likewise all clustering algorithms can be considered for clustering the nodes in text file and by comparing the query dependent summary using certain criteria, the best clustering algorithm can be suggested. Furthermore same technique can be applied on different file formats and best clustering algorithm can be suggested for different file formats.

VII. CONCLUSION

Nearest neighbor clustering algorithm is showing better performance than agglomerative hierarchical clustering algorithm. If we cluster the text document using these two clustering algorithms, then it is observed that the execution time, space complexity of nearest neighbouring clustering algorithm is less than agglomerative hierarchical clustering technique. It is also observed that the length of summary and quality of summary for big size cluster, performance of nearest neighboring clustering algorithm is better. So I have a conclusion that in any sense out of these two, nearest neighboring clustering algorithm is better.

VIII. REFERENCES

[1] Ramakrishna Varadarajan, Vangelis Hristidis ,"A System for Query-Specific Document Summarization"
[2] C. Ravindranath Chowdary P Sreenivasa Kumar "An Incremental Summary Generation System"
[3]Regina Barzilay and Michael Elhadad ,"Using Lexical Chains for Text Summarization"
[4]Mohamed Abdel Fattah, and Fuji Ren ,"Automatic Text Summarization"
[5]Parul Agarwal, M. Afshar Alam, Ranjit Biswas ,"Analyzing the agglomerative hierarchical Clustering Algorithm for Categorical Attributes "
[6]Jackie CK Cheung ,"Comparing Abstractive and Extractive Summarization of Evaluative Text: Controversiality and Content Selection"
[7]Jie Tang, Limin Yao, and Dewei Chen ,"Multi-topic based Query-oriented Summarization"
[8]R.M.Aliguliyev ,"Automatic Document Summarization by Sentence Extraction"
[9]Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Proceedings of the ACL-04 Workshop: Text Summarization Branches Out, pages 74–81, Barcelona, Spain.
[10]Software Engineering: A Practitioner's Approach (Sixth Edition) - by Roger S. Pressman.
[11]The complete Reference of .NET - by Matthew,Tata MacGraw Hill Publication Edition 2003
[12]Object Oriented Analysis and Design with Applications By Grady Booch, Pearson Education Asia
[13] Luhn H. P. 1958, The automatic creation of literature abstracts, IBM Journal, pages 159-165
[14] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Proceedings of the ACL-04 Workshop: Text Summarization Branches Out, pages 74–81, Barcelona, Spain.
[15] Deshmukh Yogesh,Shirole Bajirao,"Analysis of query dependent text document summarization using clustering techniques".
[16] Nikita Munot, Sharvari S. Govilkar, " Comparative Study of Text Summarization Methods" International Journal of Computer Applications (0975 – 8887) Volume 102– No.12, September 2014.