CSP301 Assignment-1

(Prefuse – Data Visualisation)

Study of Social Networks via Data Visualisation

Submitted by:

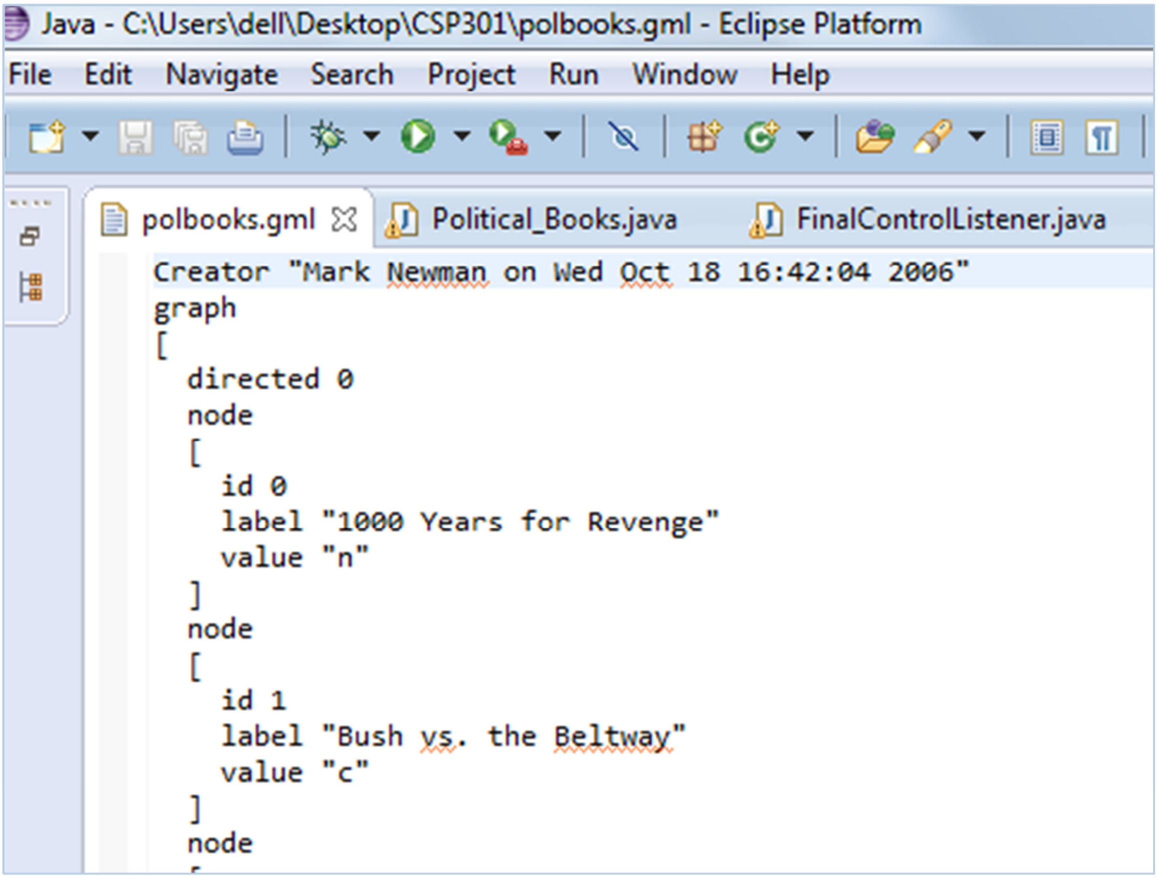
Abhishek Bansal 2011CS50271

Nipun Gupta 2011CS50289

Shashank Jain 2011CS10254

1. Visualisation of Sale of Political Books

1.1 Input: Input is in the form of .gml file(named polbooks.gml) which contains information about political books on US politics . Data is in the form of an undirected graph(Compiled by Valdis Krebs) in which Nodes represent books about US politics sold by the online bookseller Amazon.com and Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon. Nodes have been given values "l", "n", or "c" to indicate whether they are "liberal", "neutral", or "conservative".


Java - C:\Users\dell\Desktop\CSP301\polbooks.gml - Eclipse Platform
File Edit Navigate Search Project Run Window Help
polbooks.gml
Political_Books.java
FinalControlListener.java

```
Creator "Mark Newman on Wed Oct 18 16:42:04 2006"
graph
[
  directed 0
  node
  [
    id 0
    label "1000 Years for Revenge"
    value "n"
  ]
  node
  [
    id 1
    label "Bush vs. the Beltway"
    value "c"
  ]
  node
```

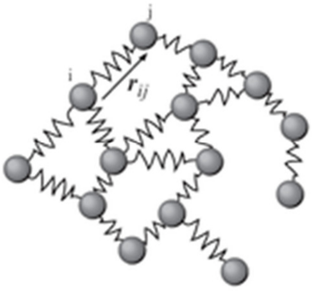
These alignments were assigned separately by Mark Newman based on a reading of the descriptions and reviews of the books posted on Amazon. These data should be cited as V. Krebs, unpublished, http://www.orgnet.com/.

Total Nodes/Books = 105

Total Edges = 441

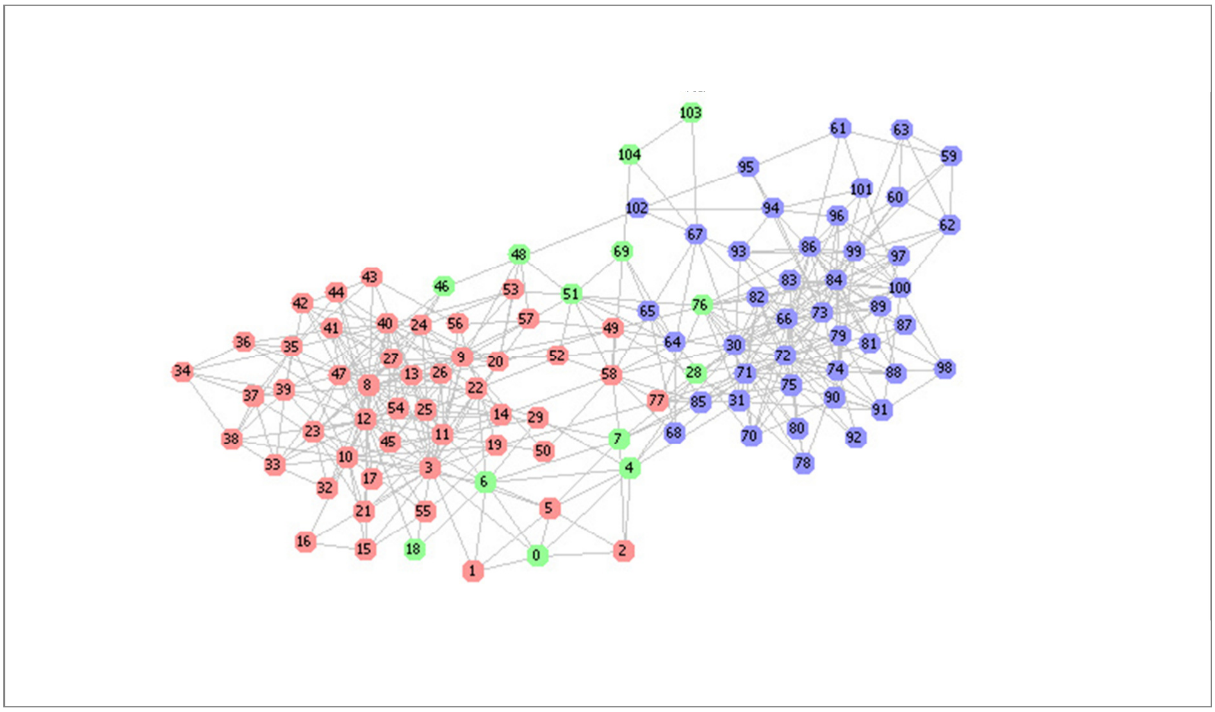
1.2. Visualisation/Objective: We want to build a visualisation to study the reading pattern among people, to study if people like to read diverse books that touch upon several different affiliations or they rather like to read stuff that possibly resonates with their own viewpoints, i.e., what is the tendency of a person to read books of different ideologies? And Extent of Polarisation of visualisation.

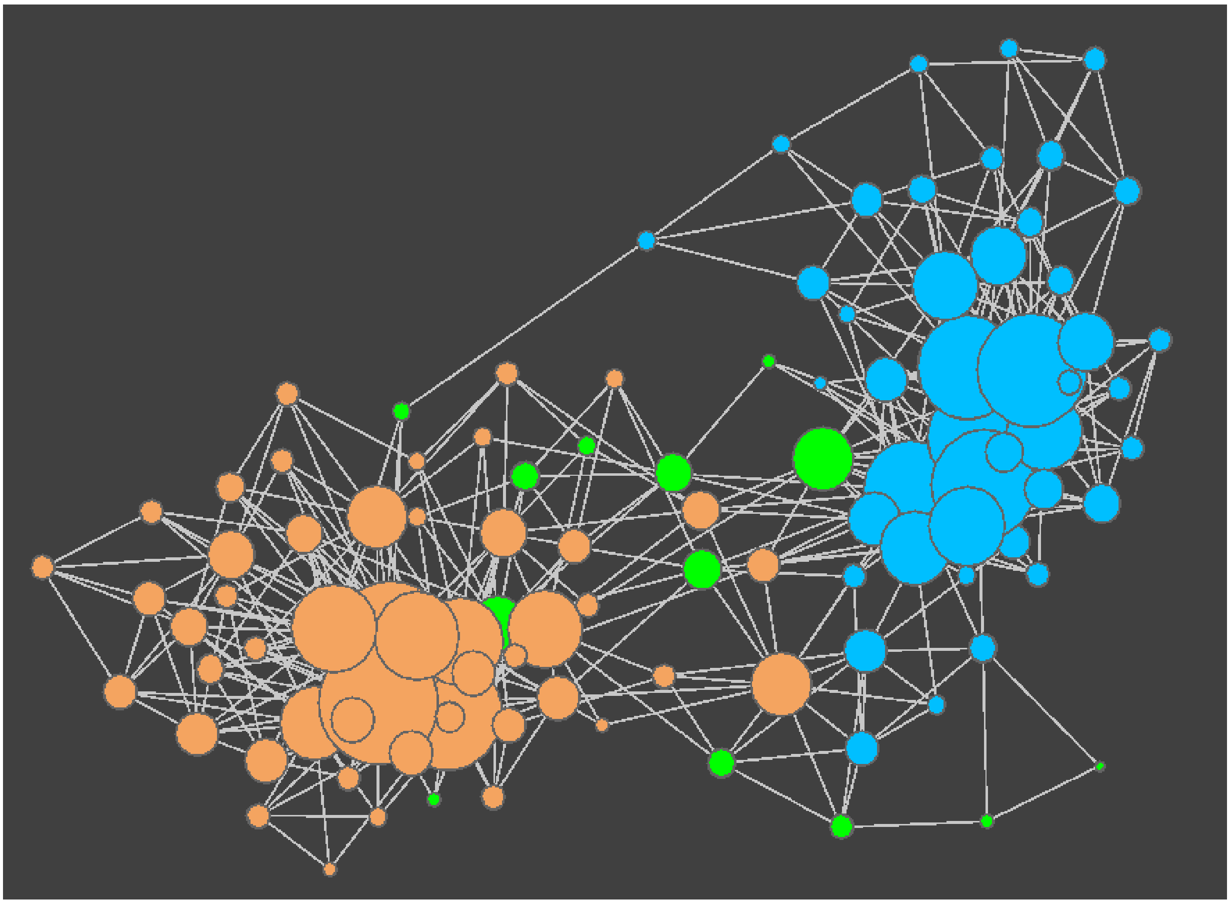
1.3. Clustering Algorithm<Force Directed algorithm/Spring algorithm>: We wanted that densely connected nodes must form clusters, so that the proximity of two nodes should be proportional to their connectivity. For that we made each node a charged point so that they repel each other (opposite charges repel!!!) and each edge is equivalent to an elongated spring between nodes which tends to bring the nodes together.



So, depending on the connectivity between two nodes, they are pulled together by the springs while their charges tend to repel them and we finally see is the most stable state or lowest energy state.

1.4. Output:


103
61
63
104
59
95
101
60
94
102
96
62
67
93
86
99
97
48
69
43
46
83
84
100
44
53
51
82
42
76
89
65
57
73
66
41
40
24
56
49
87
36
35
79
81
64
30
27
9
72
20
52
34
47
13
26
28
71
74
88
98
58
8
22
75
37
39
77
85
31
90
54
25
91
12
14
29
23
11
68
80
38
45
70
92
19
50
7
10
33
3
4
78
17
6
32
5
21
55
16
15
18
2
0
1



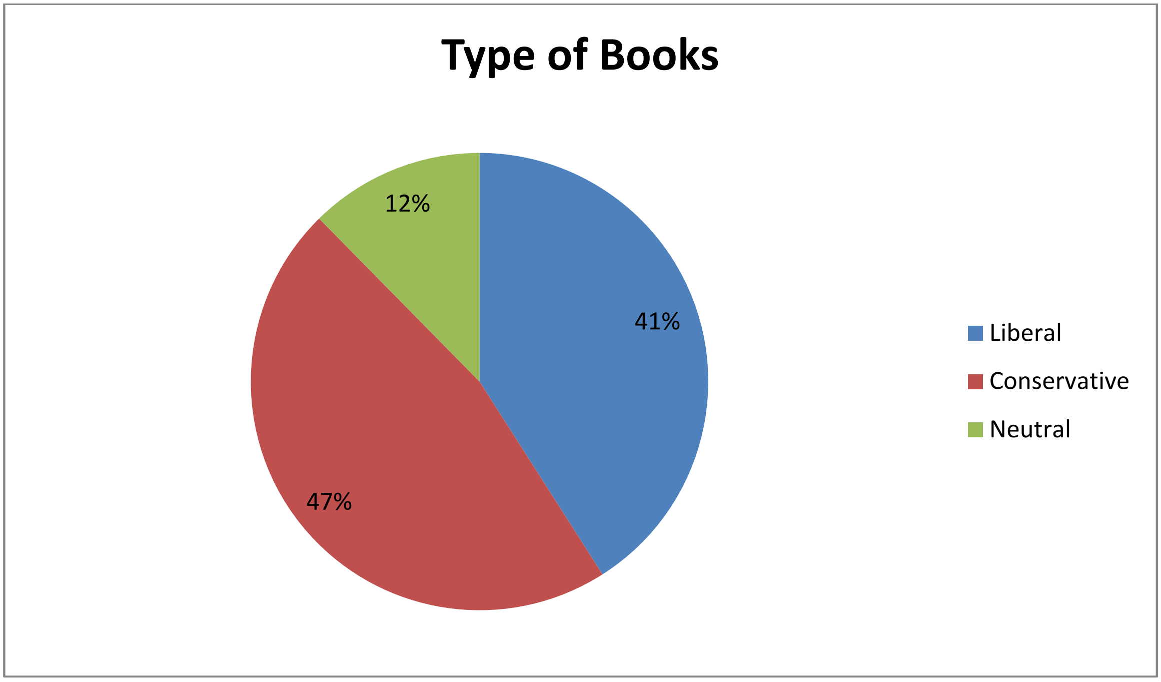
As it can be seen from first graph, nodes form two separated clusters each of liberal books(blue) and conservative books(red), while the neutral books(green), which are very few in number, lies between the two clusters.

Second graphs shows the degree of each node (size of node is proportional to its degree) and the distribution of edges between different nodes.

2. Studying the Visualisation

2.1. Number of books:

Total number of books : 105
Conservative (purple nodes): 49
Neutral (black nodes): 13
Liberal (green nodes): 43

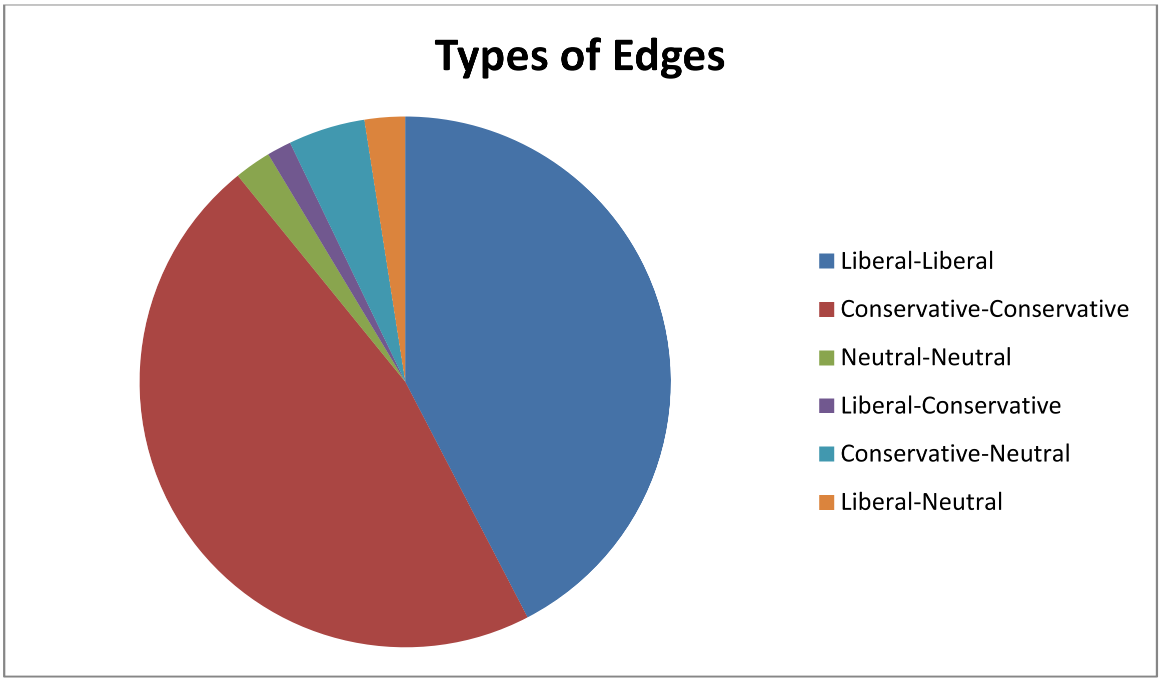


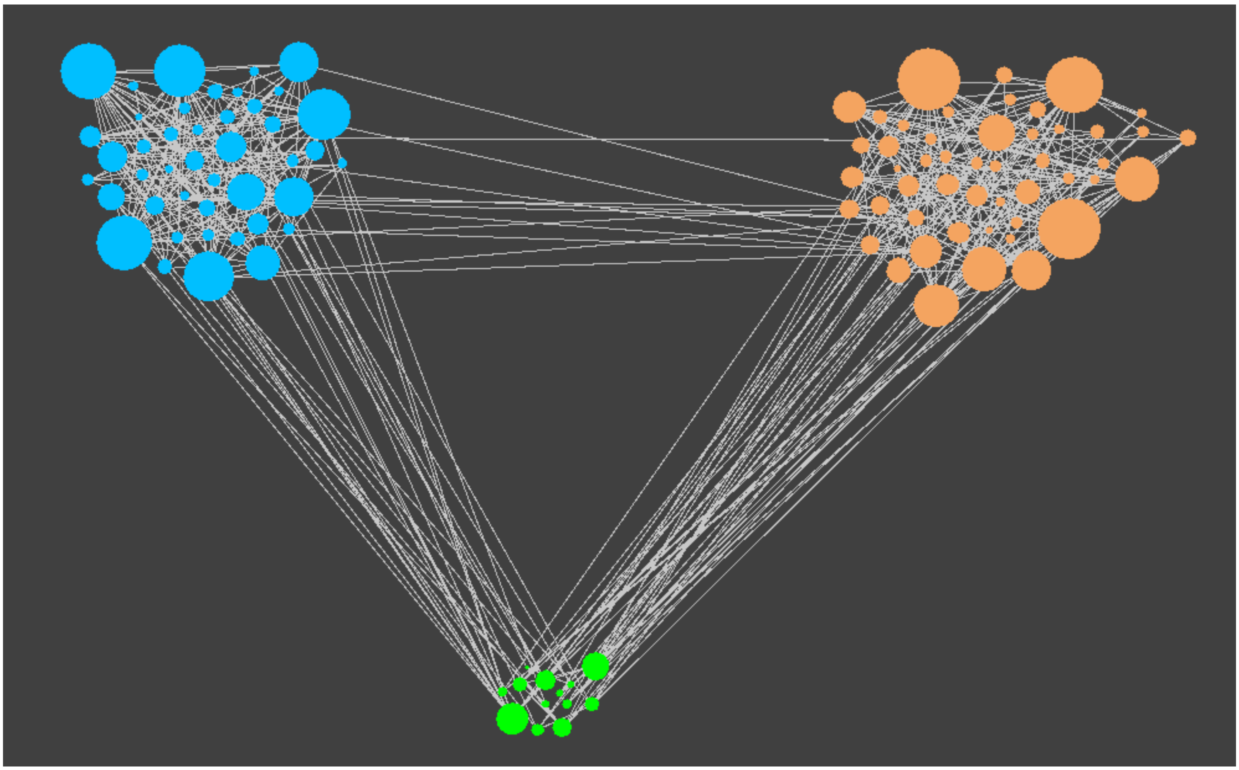
#Point to note: Number of neutral ideology books is much less than the number of radical books (liberal/conservative)

2.2. Number of edges:

Total number of edges: 441

Conservative-Conservative: 190	Conservative-Neutral: 19
Neutral-Neutral: 9	Neutral-Liberal: 10
Liberal-Liberal: 172	Liberal-Conservative: 6



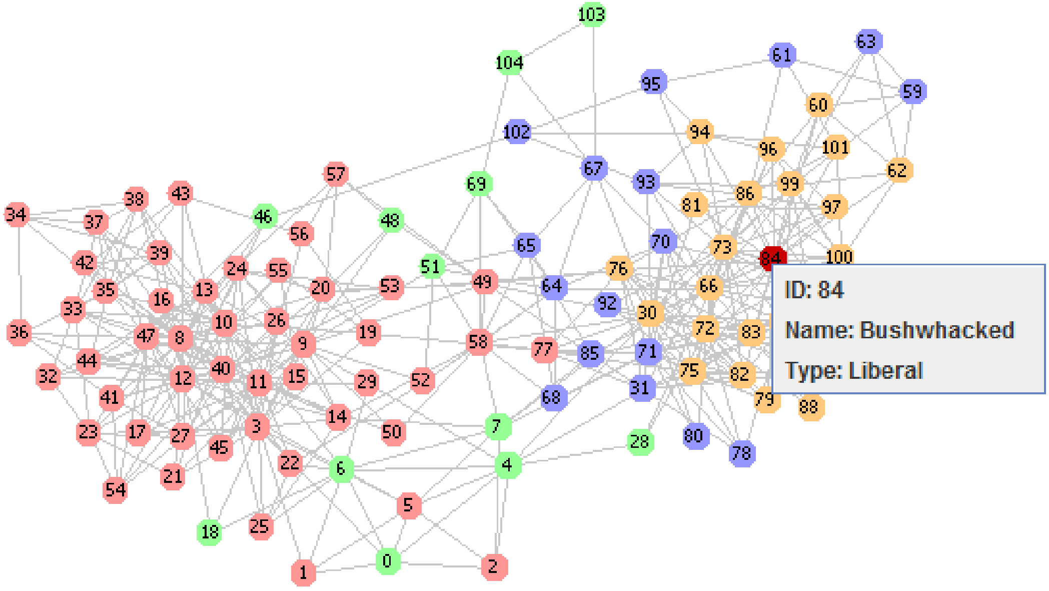


#Point to Note: For both Liberal and conservative books, edges within themselves is much higher than that with other type books, but for neutral books, neutral-neutral edges are only 9 while the edges between neutral and other type books is 19, showing that people having read neutral books have a high tendency to read books of radical ideologies rather than reading neutral books. Another point to note is that the number of inter-linkages between Neutral-Liberal and Neutral-Conservative are higher than those between Liberal-Conservative.

2.3. Most Popular Books:

2.3. Most Popular Books: More the edges of a node in the graph, more is the popularity of that node because edges in some sense also shows the tendency of the buyer to buy that book.

2.3.A. Liberal Book :



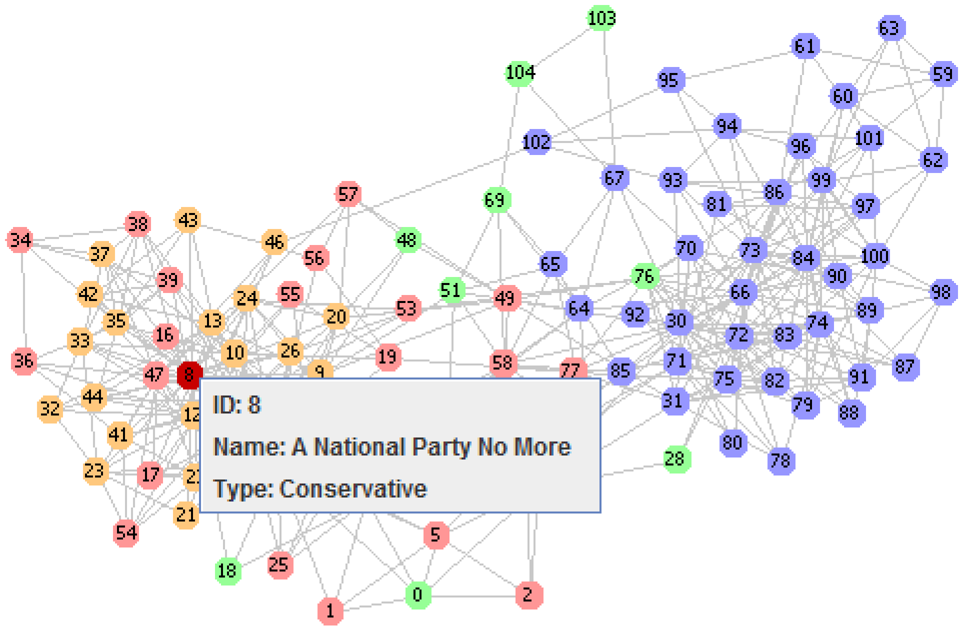
ID: 84

Name: Bushwhacked

Type: Liberal

Book named "Bushwhacked" has maximum 23 edges.

2.3.B. Conservative Book:

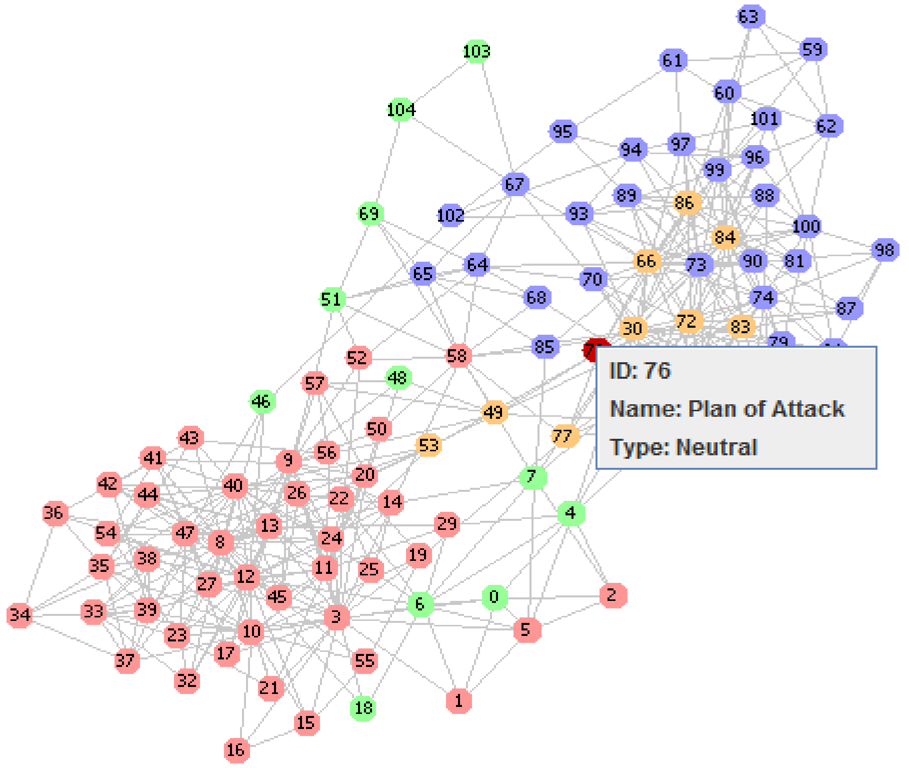

ID: 8

Name: A National Party No More

Type: Conservative

Books named “A National Party No More” and “Off with their Heads” have maximum 25 edges.

2.3.C. Neutral Book :

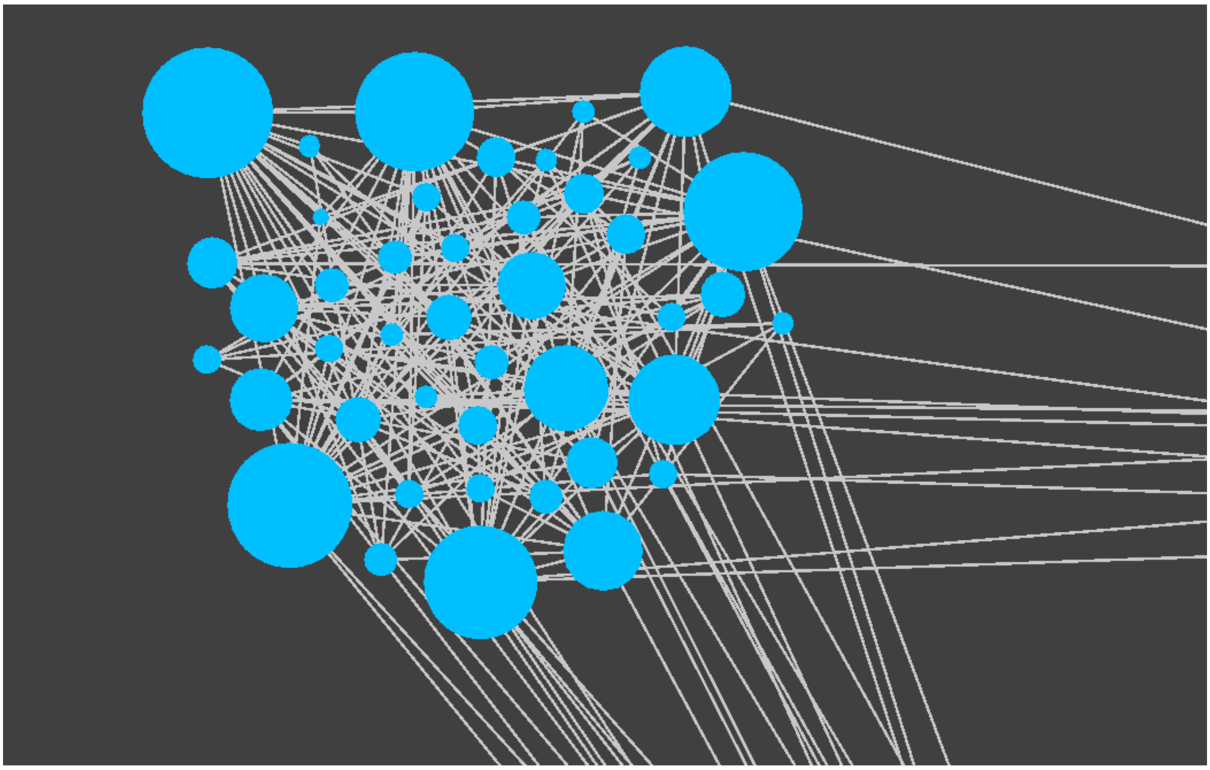

ID: 76

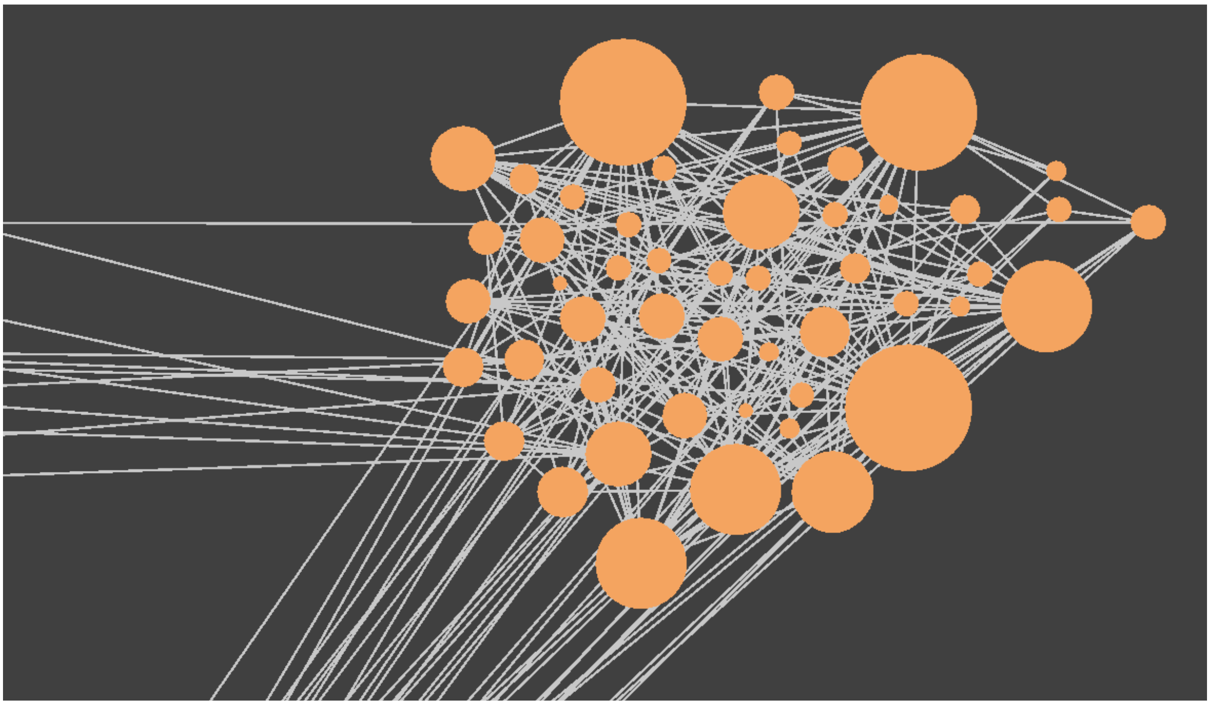
Name: Plan of Attack

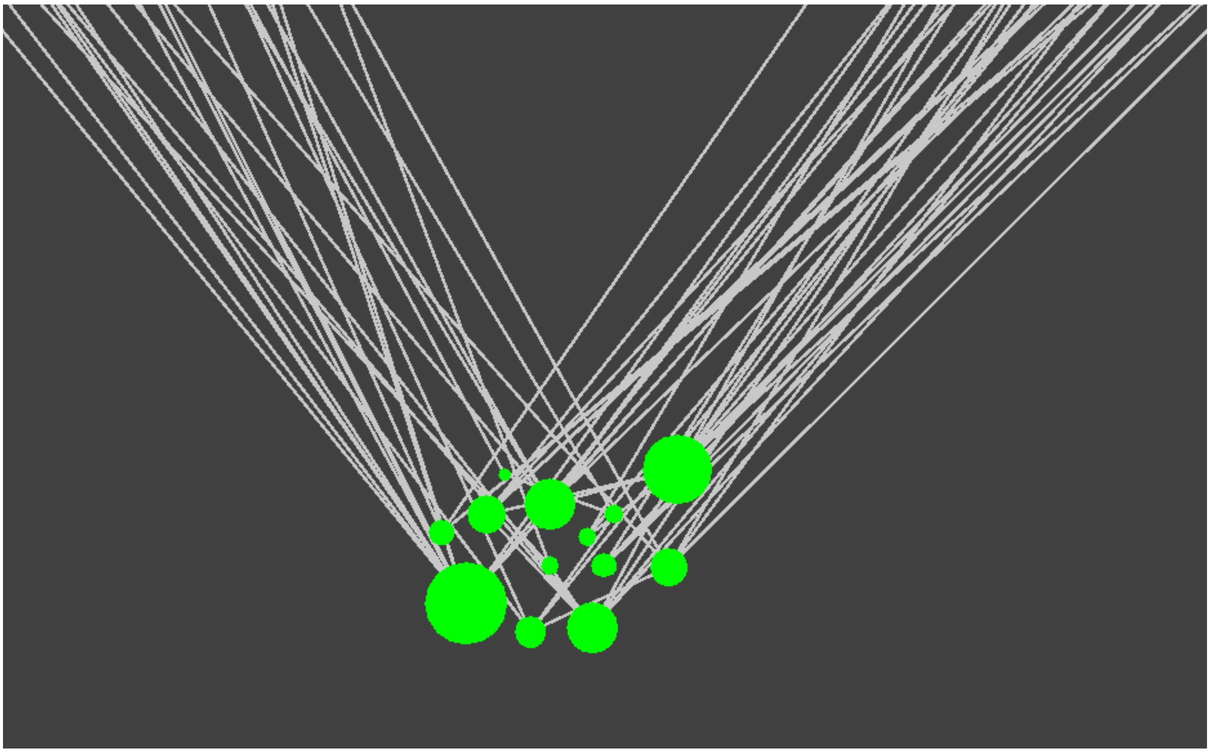
Type: Neutral

Book named “Plan of Attack” has maximum 13 edges.

Let us also look the degree distribution of each node:

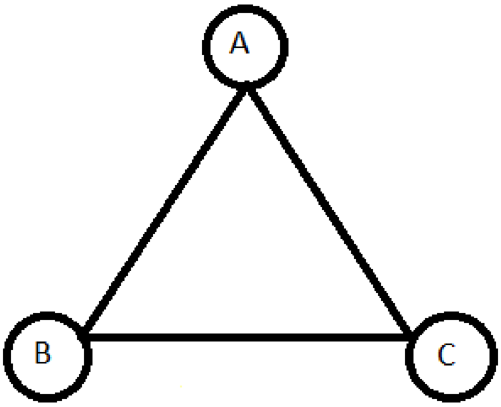






#Interesting Fact: Most popular Liberal and conservative books have edges to books only of their own type, while the most popular neutral book does not have any edge to any neutral book.

2.4. Triads: 3 nodes connected to each other, that is A-B, B-C and C-A, form a triad.



Total number of triads: 560

C-C-C triads: 241

N-N-N triads: 1

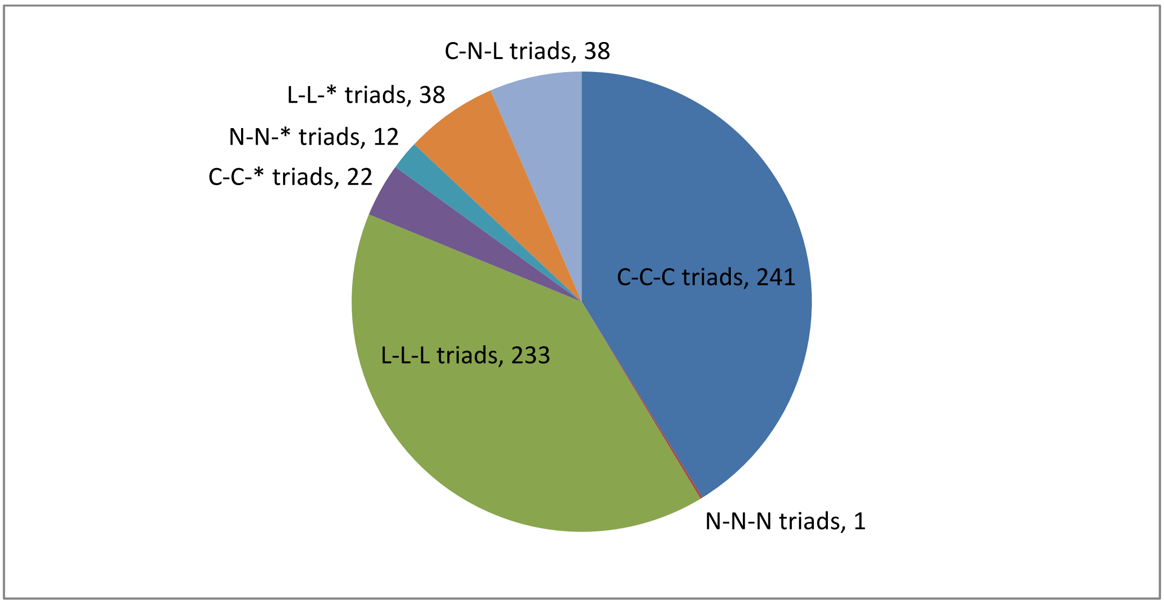
L-L-L triads: 233

C-C-* triads: 22

N-N-* triads: 12

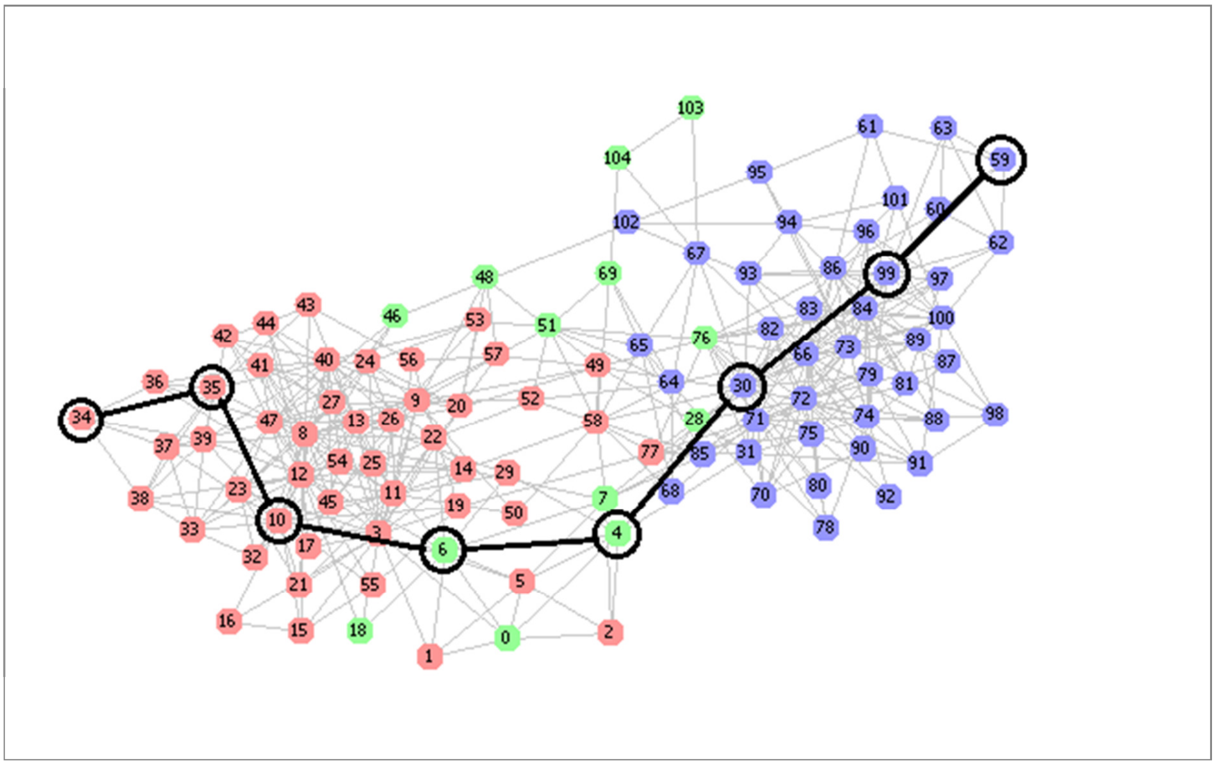
L-L-* triads: 38

C-N-L triads: 13



#Point to Note: Triads can be seen as a measure of density of connectivity/Edges. Above data shows that nodes are much densely connected among themselves than with nodes of other types.

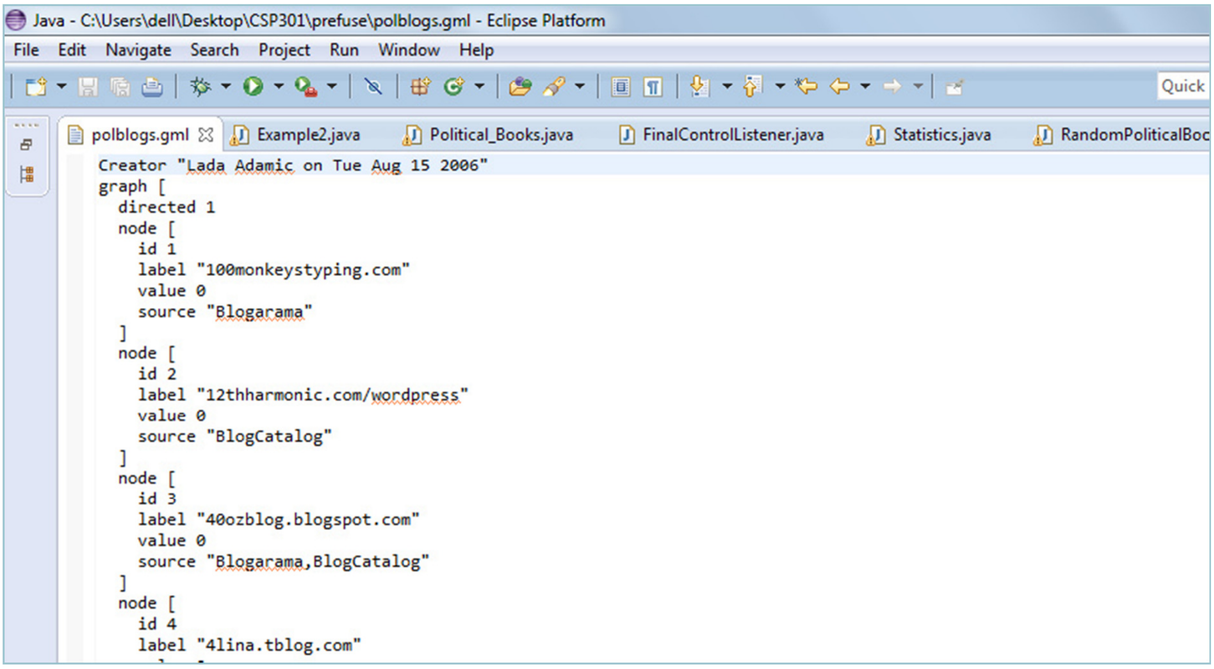
2.5. Diameter: The diameter of a graph is the maximum eccentricity of any vertex in the graph. That is, it is the greatest distance between any pair of vertices. To find the diameter of a graph, first find the



shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph.
For our given graph, diameter = **7 unit**.

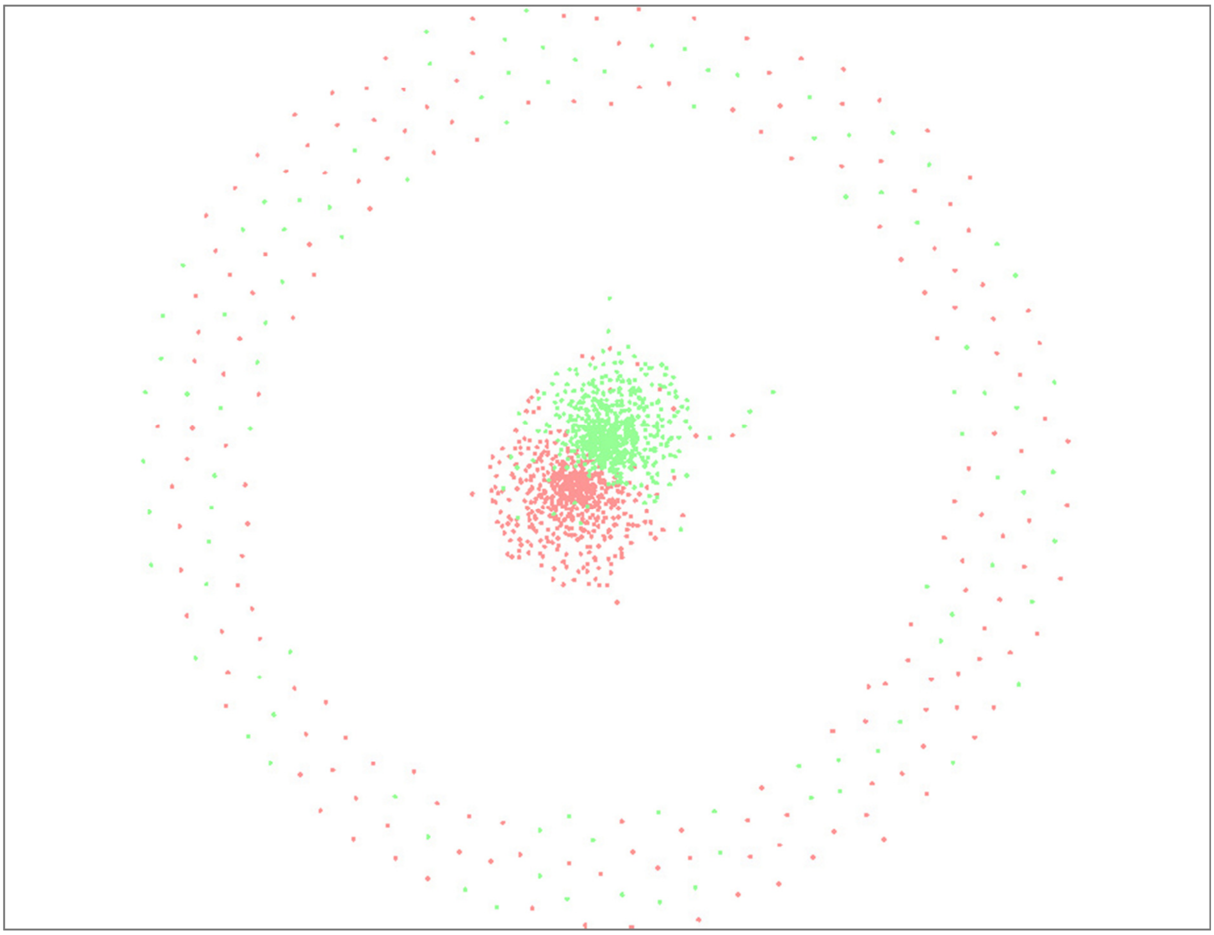
3. Visualisation of a similar data – "References to Political Blogs":

3.1. Input: Input is the form of a .gml extension file (named "polblogs.gml) which contains information about political blogs in US. Each node represent a political blog which has a value "0" or "1", depending on whether the blog is politically leaning left(liberal) or right(conservative). One major difference between PolBooks graph and PolBlogs graph is that the latter is a directed graph. Each edge has a source and a target and an edge between two nodes/blogs represent the reference of the target blog on the source blog.


Java - C:\Users\dell\Desktop\CSP301\prefuse\polblogs.gml - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Quick
polblogs.gml
Example2.java
Political_Books.java
FinalControlListener.java
Statistics.java
RandomPoliticalBoc

```
Creator "Lada Adamic on Tue Aug 15 2006"
graph [
  directed 1
  node [
    id 1
    label "100monkeystyping.com"
    value 0
    source "Blogarama"
  ]
  node [
    id 2
    label "12thharmonic.com/wordpress"
    value 0
    source "BlogCatalog"
  ]
  node [
    id 3
    label "40ozblog.blogspot.com"
    value 0
    source "Blogarama,BlogCatalog"
  ]
  node [
    id 4
    label "4lina.tblog.com"
```

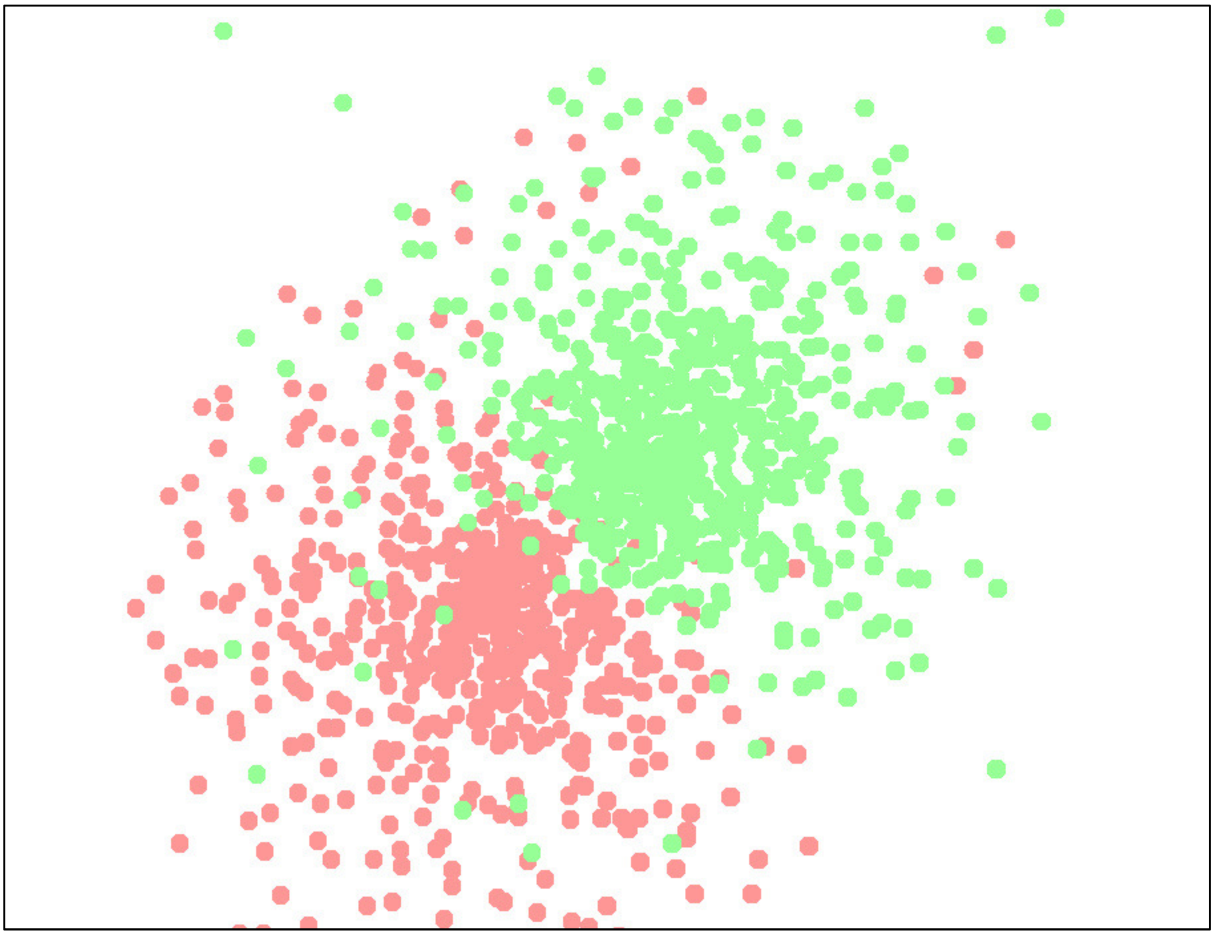
3.2. Output:



As it can be seen, the visualisation is much similar to that of PolBooks, with two large clusters of Liberal and Conservative nodes. Clustering algorithm followed is the same, i.e., Force Directed Layout, comprising of similarly charged nodes with springs between them and what we finally see is the most stable state of the given situation.

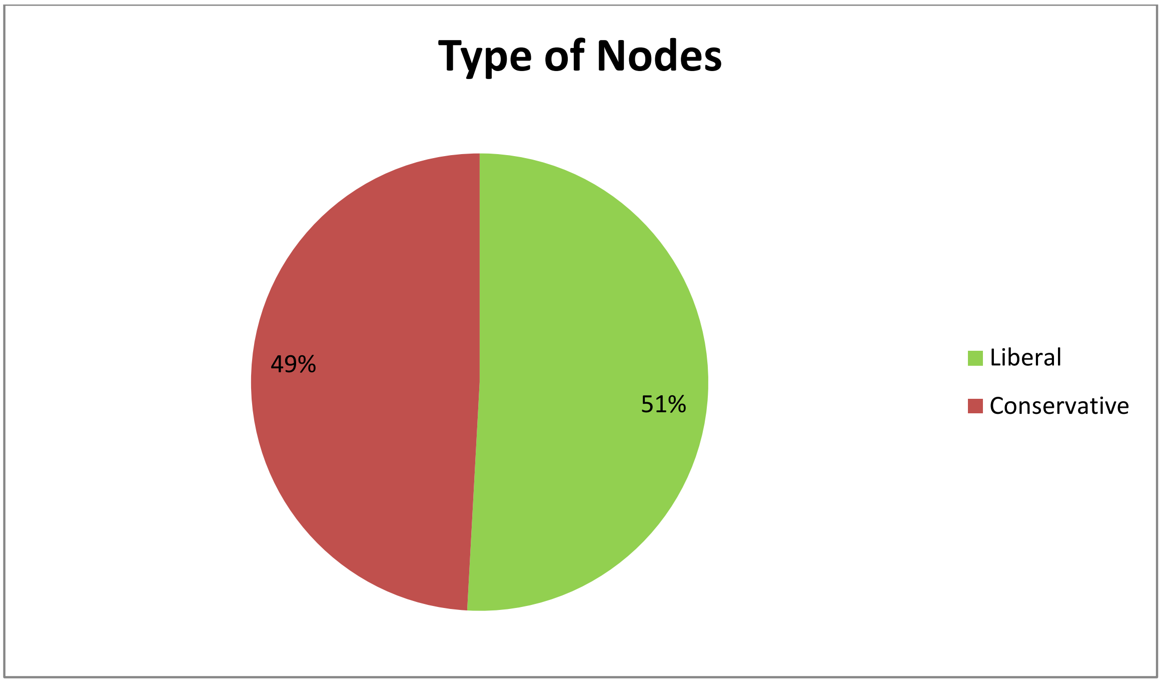
#Point to Note: There are some nodes which are much far away than our main clusters. It is because these nodes are very weakly connected to the other nodes (i.e. , number of edges joining these nodes to other nodes are very less), and hence due to the strong repulsive forces from the centred clusters, these nodes have the tendency to move as far away as possible, in order to attain a stable state.

Zoomed in:



3.3. Types and Number of Nodes:

Total Nodes = 1490
Liberal Nodes (green) : 758
Conservative Nodes (red) : 732



3.4. Types of edges:

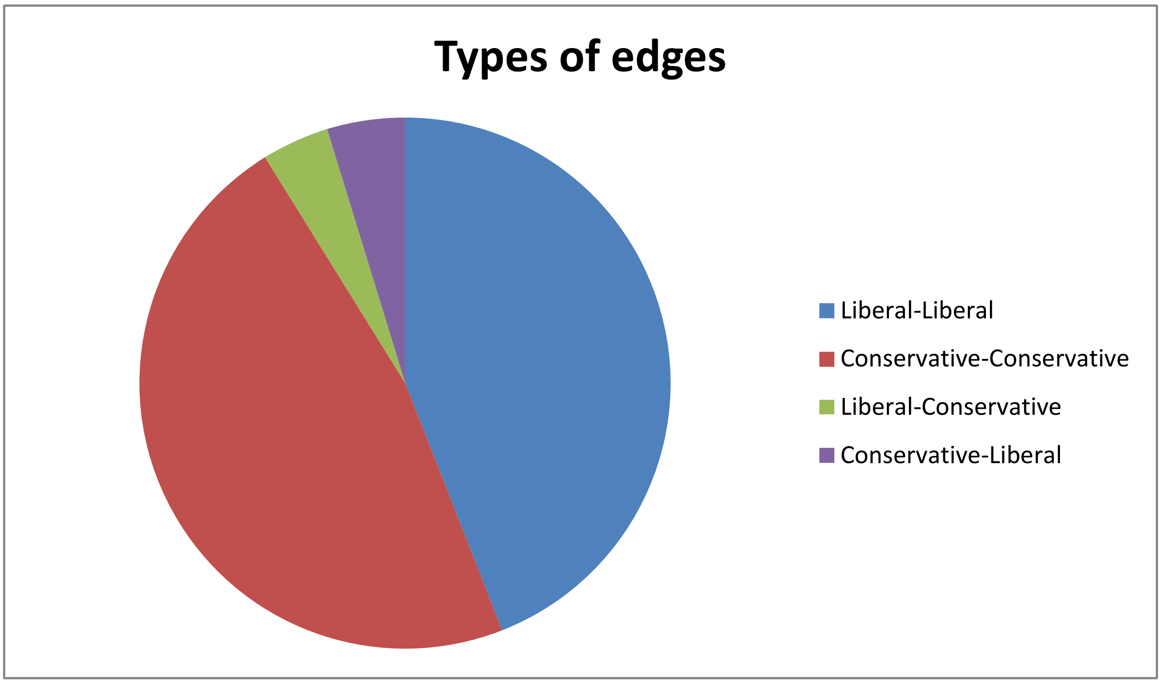
Total edges = 19090

Liberal-Liberal: 8408.0

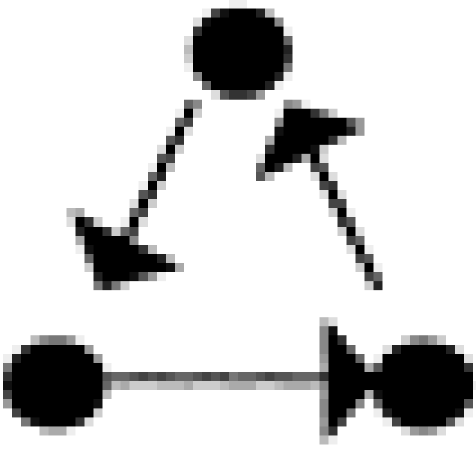
Conservative-Conservative: 8994.0

Liberal-Conservative: 783.0

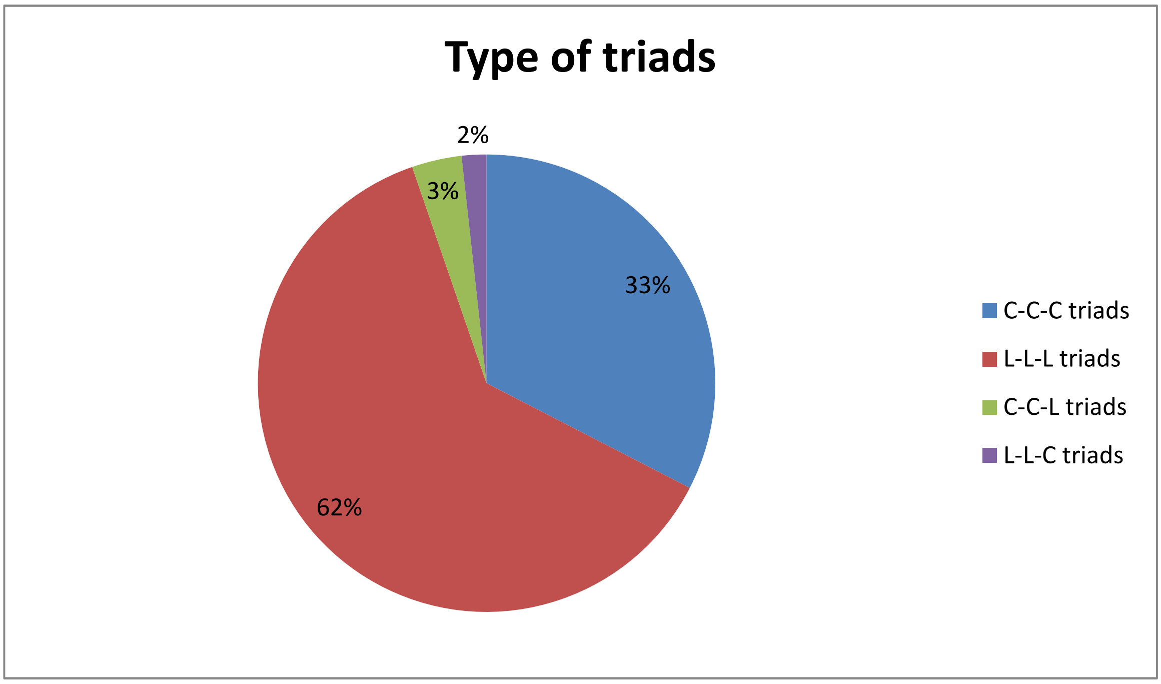
Conservative-Liberal: 905.0



3.5. Triads: In directed graphs, the triads must form a cycle.



Total triads = 21654
C-C-C triads: 7052
L-L-L triads: 13459
C-C-L triads: 761
L-L-C triads: 381



#Point to note: As mentioned before, number of triads can be linked to the density of connections/edges. The above data shows that L-type nodes are much densely connected than C-type nodes. Also the connectivity between nodes of different types is much less than among themselves.

3.6. Diameter: Diameter of the given graph = 12.

3.7. Strongly-connected components (Kosaraju's algorithm):

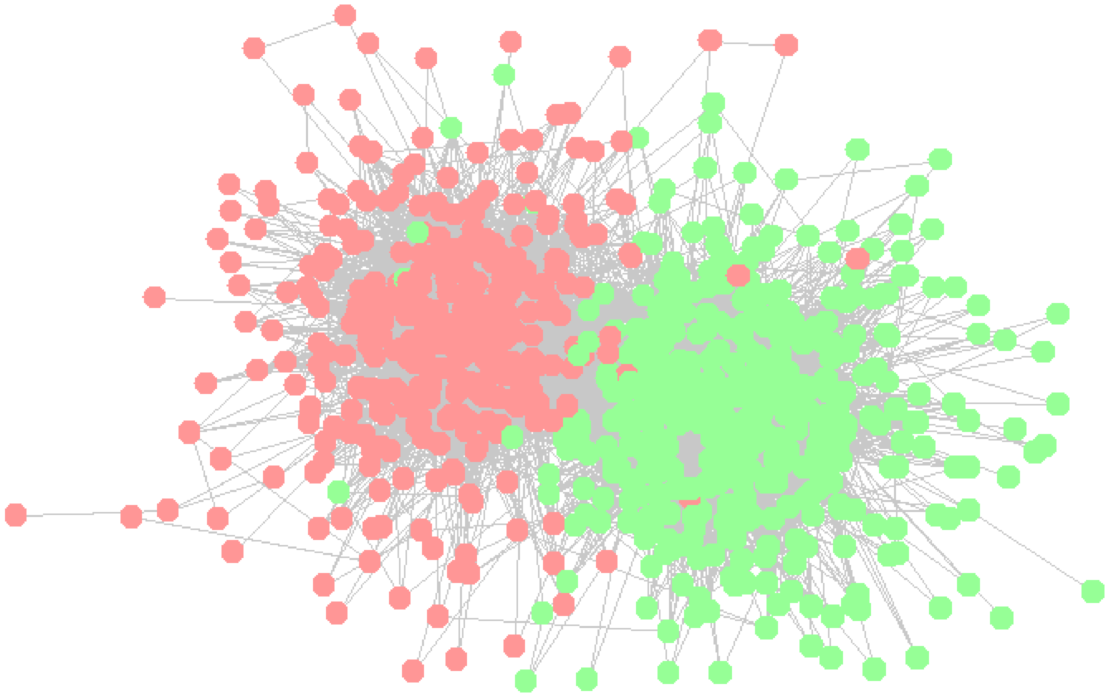
A graph component is said to be strongly-connected, if there exists a path from each node of that component to every other node of that component.

Kosaraju's algorithm (also known as the Kosaraju-Sharir algorithm) is an algorithm to find the strongly connected components of a directed graph. It makes use of the fact that the transpose graph (the same graph with the direction of every edge reversed) has exactly the same strongly connected components as the original graph.

Kosaraju's algorithm is simple and works as follows:

1. Let G be a directed graph and S be an empty stack.
2. While S does not contain all vertices, choose an arbitrary vertex v not in S. Perform a depth-first search starting at v. Each time when depth-first search finishes expanding a vertex u, push u onto S.
3. Reverse the directions of all edges to obtain the transpose graph.
4. While S is nonempty, pop the top vertex v from S. Perform a depth-first search starting at v. The set of visited vertices will give the strongly connected component containing v; record this and remove all these vertices from the graph G and the stack S. Equivalently, breadth-first search (BFS) can also be used instead of depth-first search.

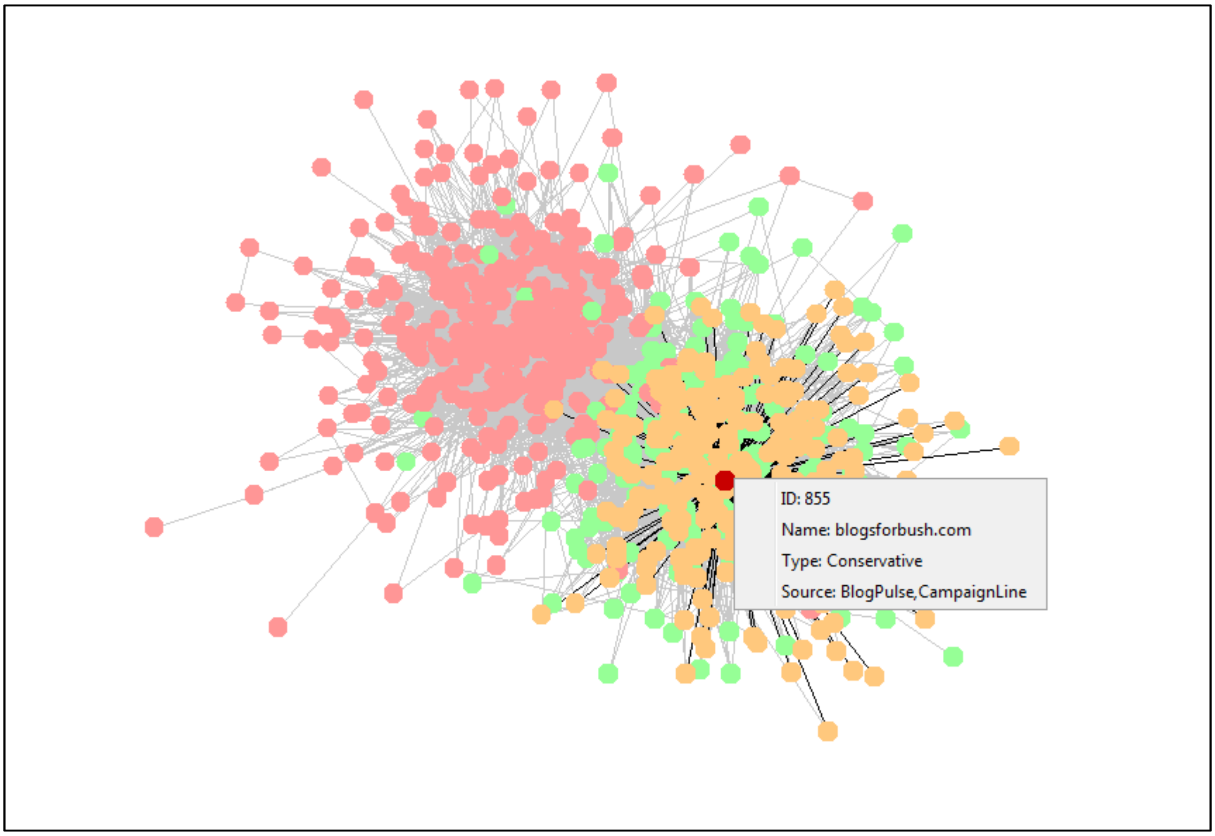
Observation: We observed a very large strongly connected component of 793 nodes out of 1490 total nodes, visualisation of which is given below. Other than that, there was only a few more strongly connected components of 2 or 3 nodes.



This shows that almost 50% of the blogs have references to each other, thus forming a strong social network among them, while the rest are not so established.

3.8. Max degree Node:

Max degree of the nodes in the graph is ‘468’. This is a conservative blog site, named “blogsforbush.com”. Max degree of Liberal nodes is ‘384’, of the blog site “dailykos.com”. Such high degrees directly implies how strongly connected these nodes are.

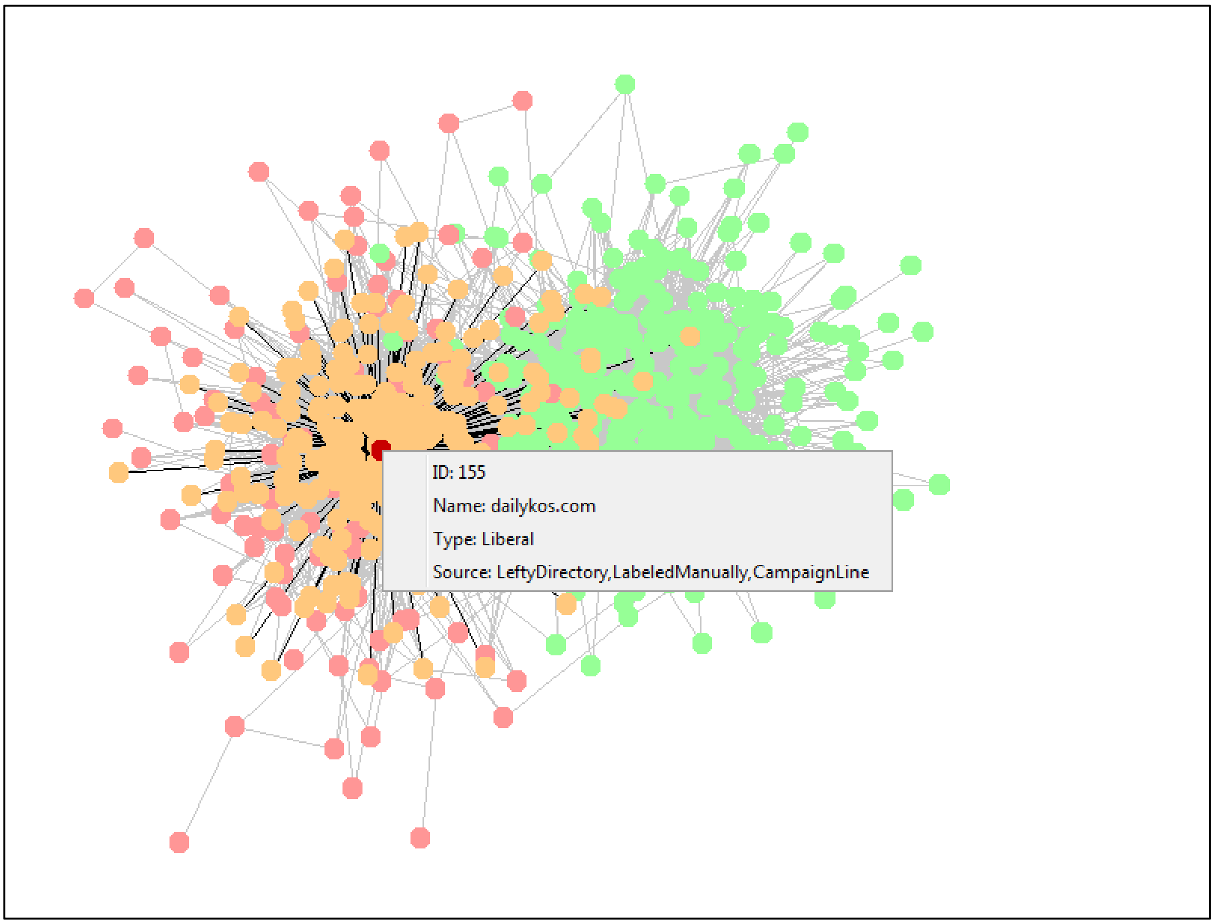


ID: 855

Name: blogsforbush.com

Type: Conservative

Source: BlogPulse,CampaignLine



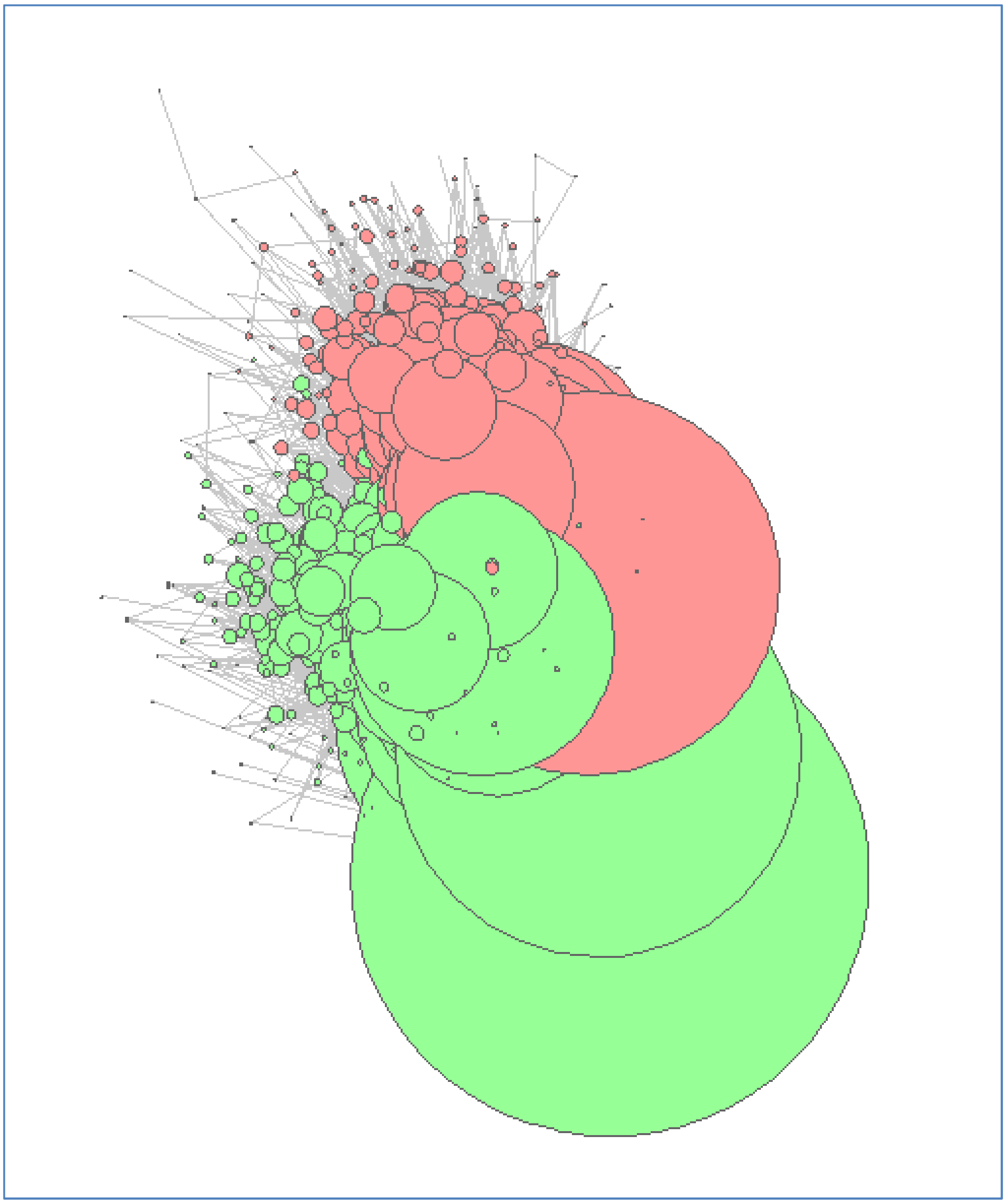
ID: 155

Name: dailykos.com

Type: Liberal

Source: LeftyDirectory,LabeledManually,CampaignLine

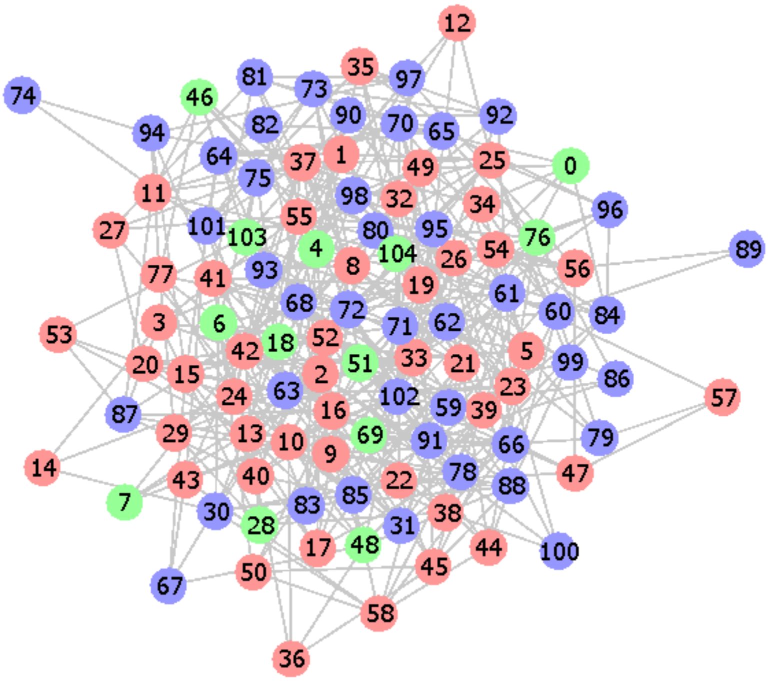
Another interesting thing we observed was when we tried to make the size of the node proportional to their degree, and the result which we obtained is quite interesting:



This visualisation is of the central Strongly-connected component (as in 3.7). As it can be seen, some of the nodes are extremely large in number, showing how strongly connected those nodes are. You may also observe some nodes which appear as dots in between these large nodes. The overall view shows how degree varies among various nodes of the same strongly connected component.

4. Comparison with Random graphs:

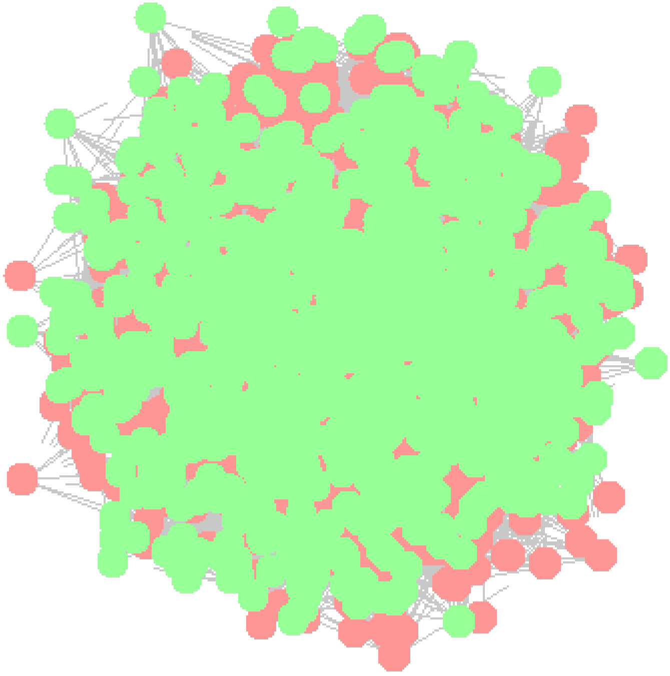
4.1. Clustering/Output:



We generated a visualisation with same books and the number of edges, but this time the edges between nodes were created randomly. Thee output we obtained is shown above. As expected, unlike our original visualisation, the nodes were distributed randomly all over the space and there was no clustering.

#Point to note: Above randomly generated graph is much more compact than our original PolBooks visualisation. Why? It is because in our original visualisation, there are many nodes which are connected to only 2-3 nodes and hence are much away from the main clusters leading to an expanded graph, while in our randomly generated visualisation, each node, on an average in connected to 8.4(= 441*2/105) nodes are hence held intact leading to a compact visualisation.

Similarly, we obtained a random a random graph for PolBlogs and found similar observations.

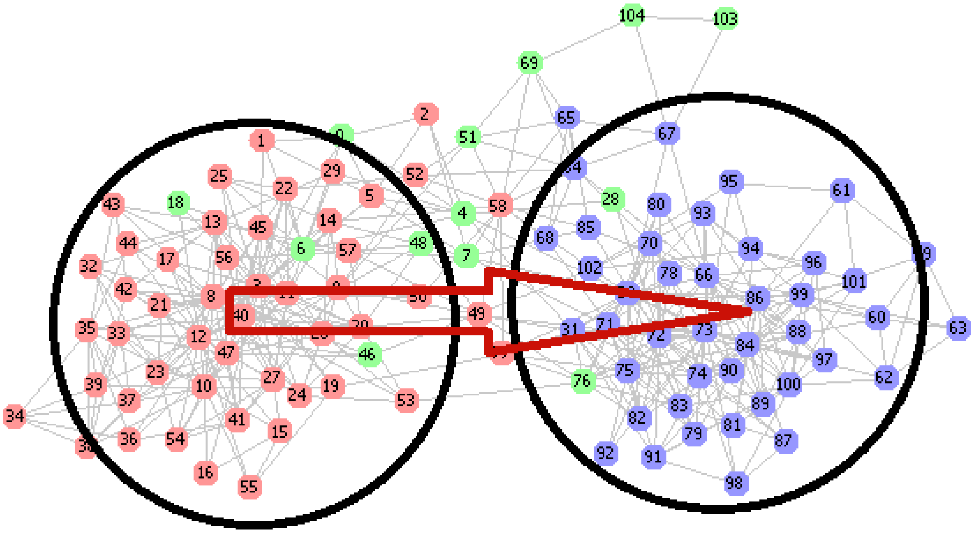


4.2. Polarisation coefficient: In order to find the polarisation of our given visualisation, we have introduced a Polarisation coefficient which is defined as:

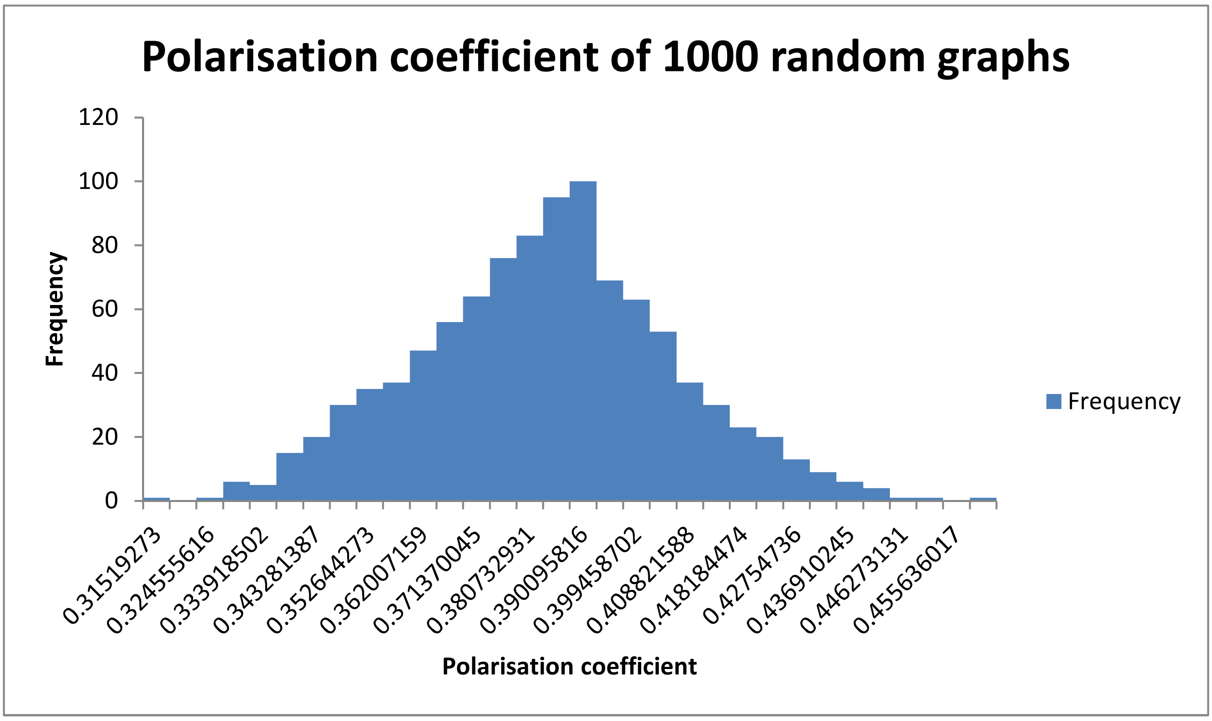
Polarisation coefficient = [the number of edges between nodes of the same type] : [the total number of edges]

4.2.A. Comparison with PolBooks :

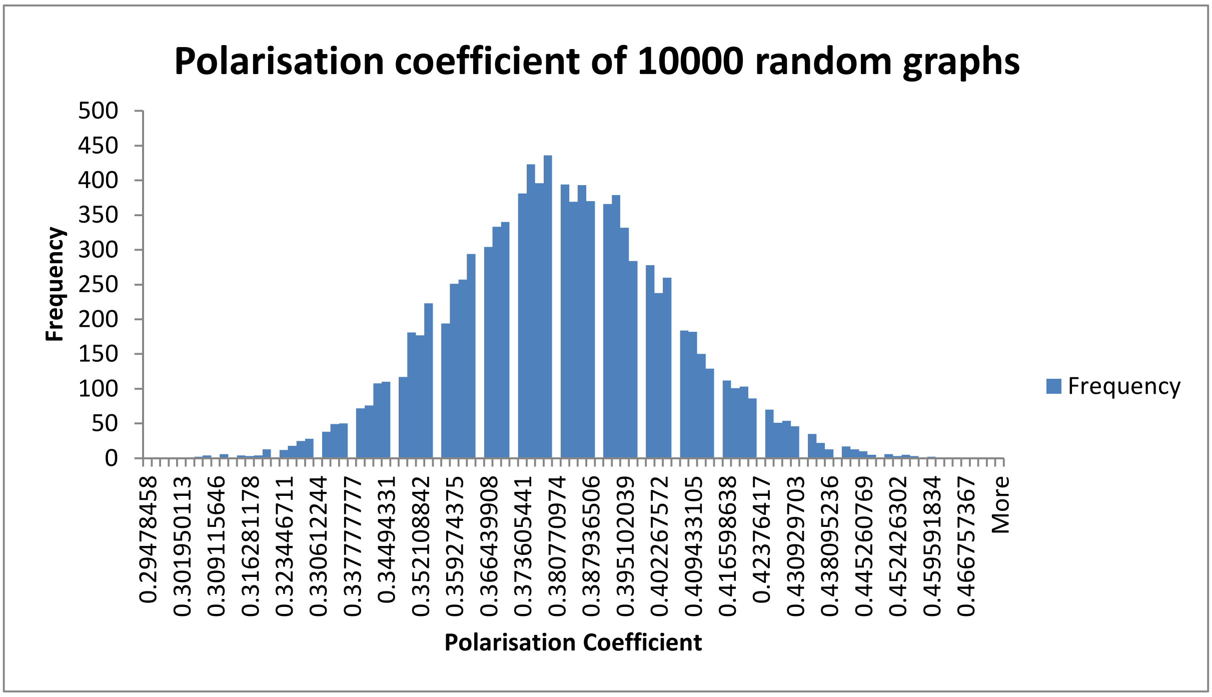
So, for our PolBooks visualisation:
Total number of edges (T) = 441
Number of Liberal-Liberal edges(x) = 172
Number of Conservative-Conservative edges(y) = 190
Polarisation coefficient = (x + y)/T = (190 + 172)/441 = **0.820862**



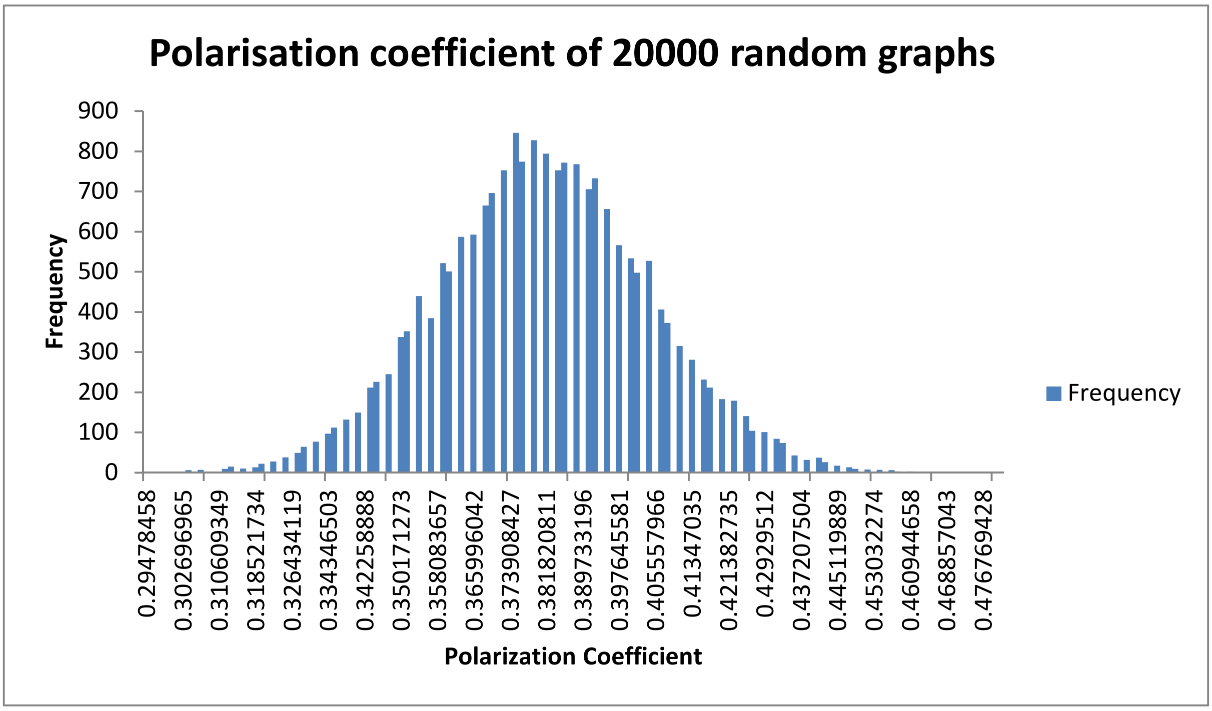
1000 random graphs:



10,000 random graphs:



20,000 random graphs:



4.2.B. Comparison with PolBlogs

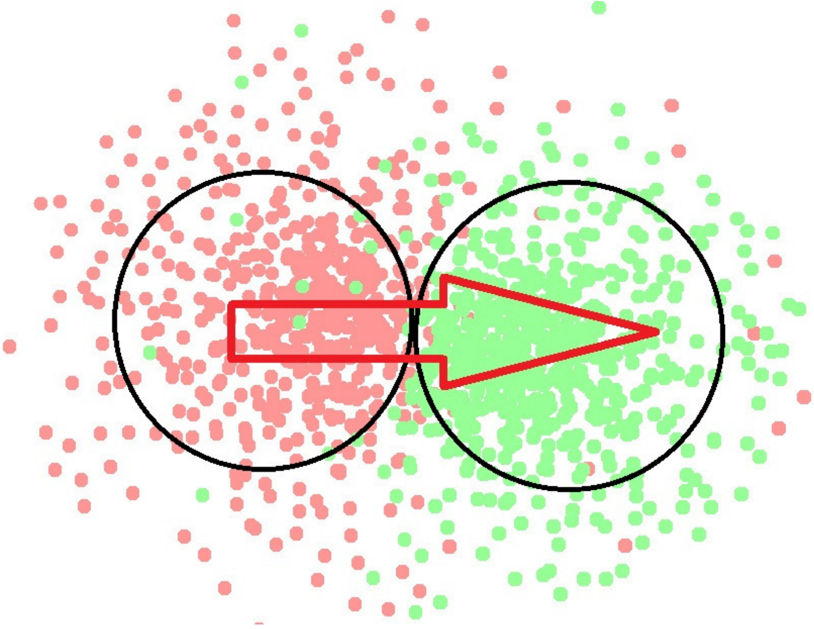
Similarly, for our PolBlogs visualisation:

Total number of edges (T) = 19090

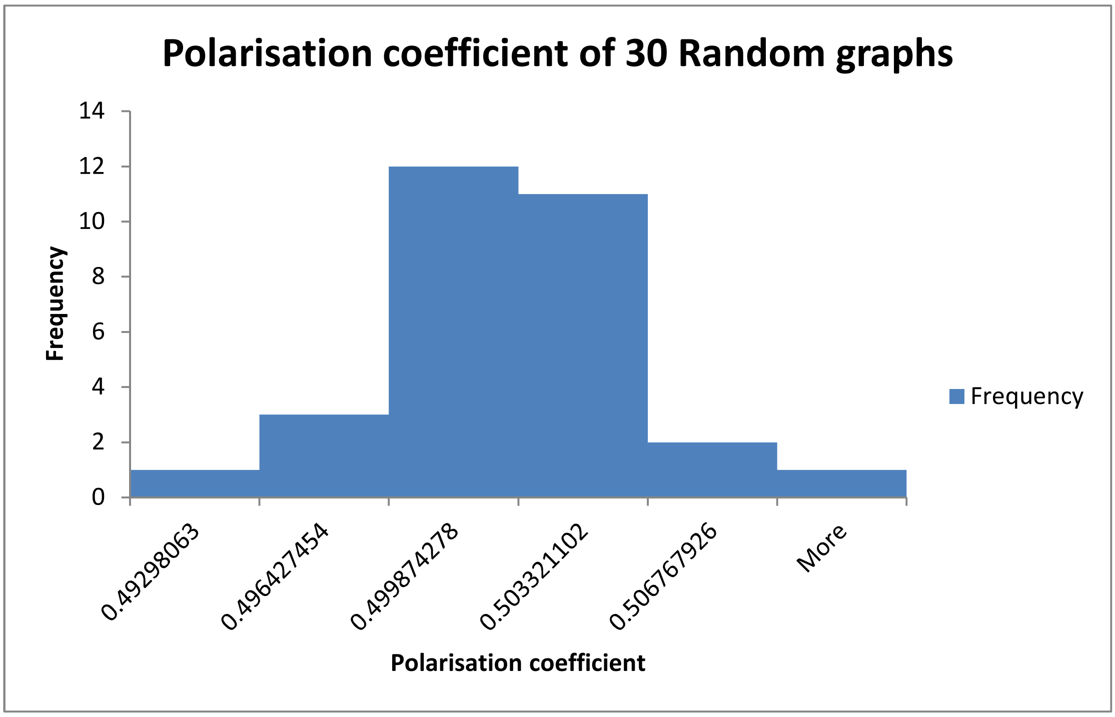
Number of Liberal-Liberal edges(x) = 8408

Number of Conservative-Conservative edges(y) = 8994

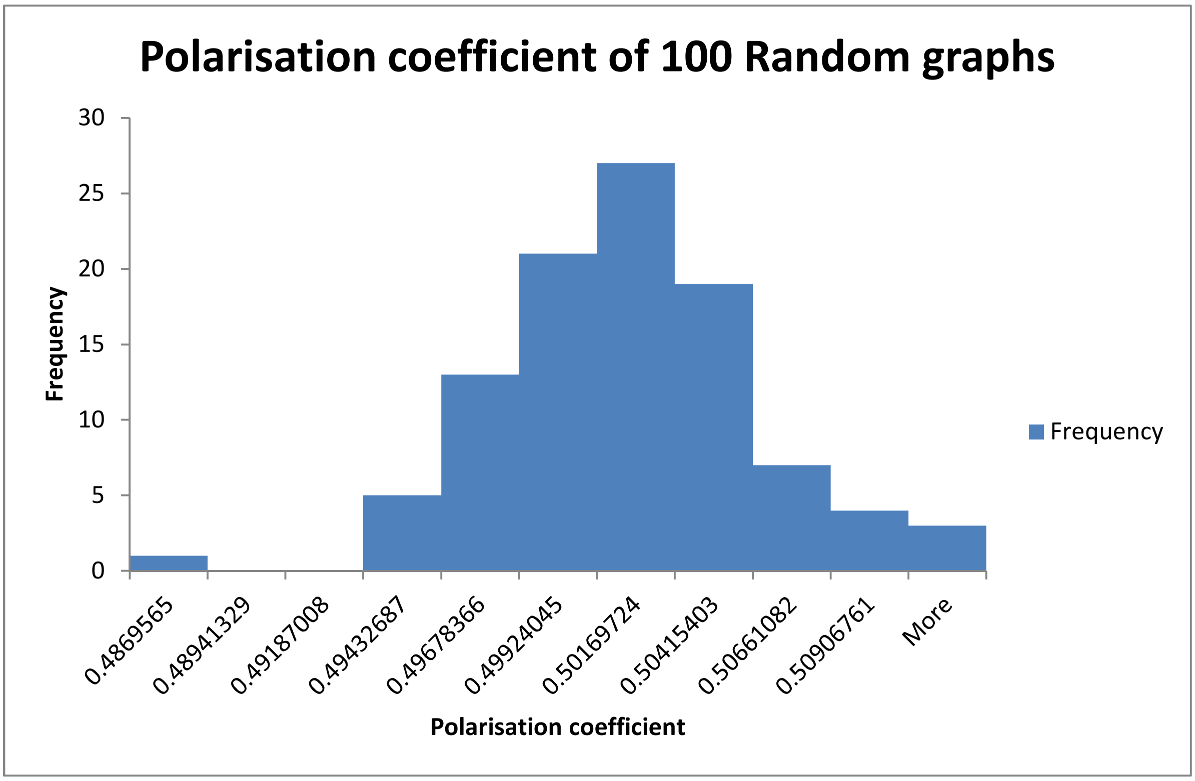
Polarisation coefficient = (x + y)/T = (190 + 172)/441 = **0.911577**



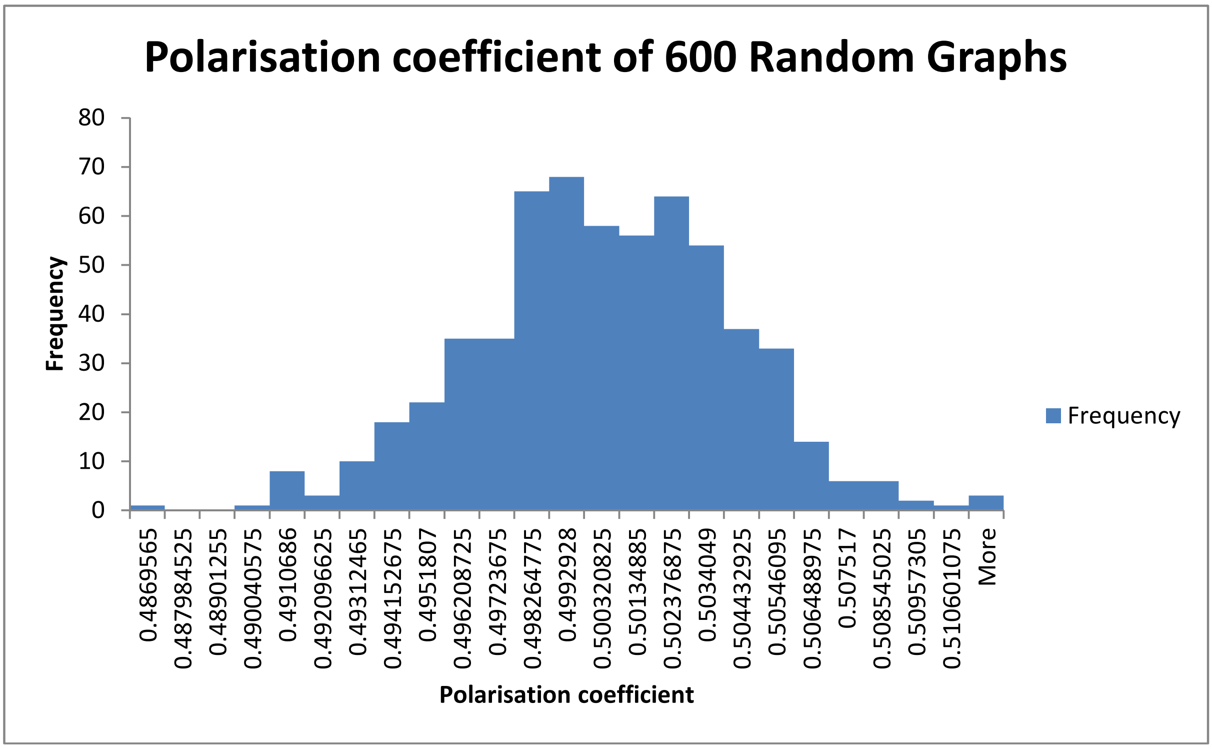
30 Random graphs of blogs:



100 Random Graphs of blogs:



600 Random Graphs of blogs :



It can be seen that both our graphs are highly polarised.
In randomly generated graphs, edges between nodes are randomly generated and hence we did expected much less polarised graph.

#Point to note:For random graphs, We have obtained a uniformly distributed bell curve, with most probable Polarisation coefficient lying around 0.380000 for Political Books and around 0.500000 for Political Blogs, which are much less than that of PolBooks graph and PolBlogs graphs respectively.

4.3. Global Clustering Coefficient: Global Clustering coefficient helps us to understand the degree of clustering in a graph, for example, if a lot of triads exist where there are edges between nodes A-B, B-C, and C-A then an existence of these triads would indicate that people tend to buy books or reference each other's blogs in clusters.

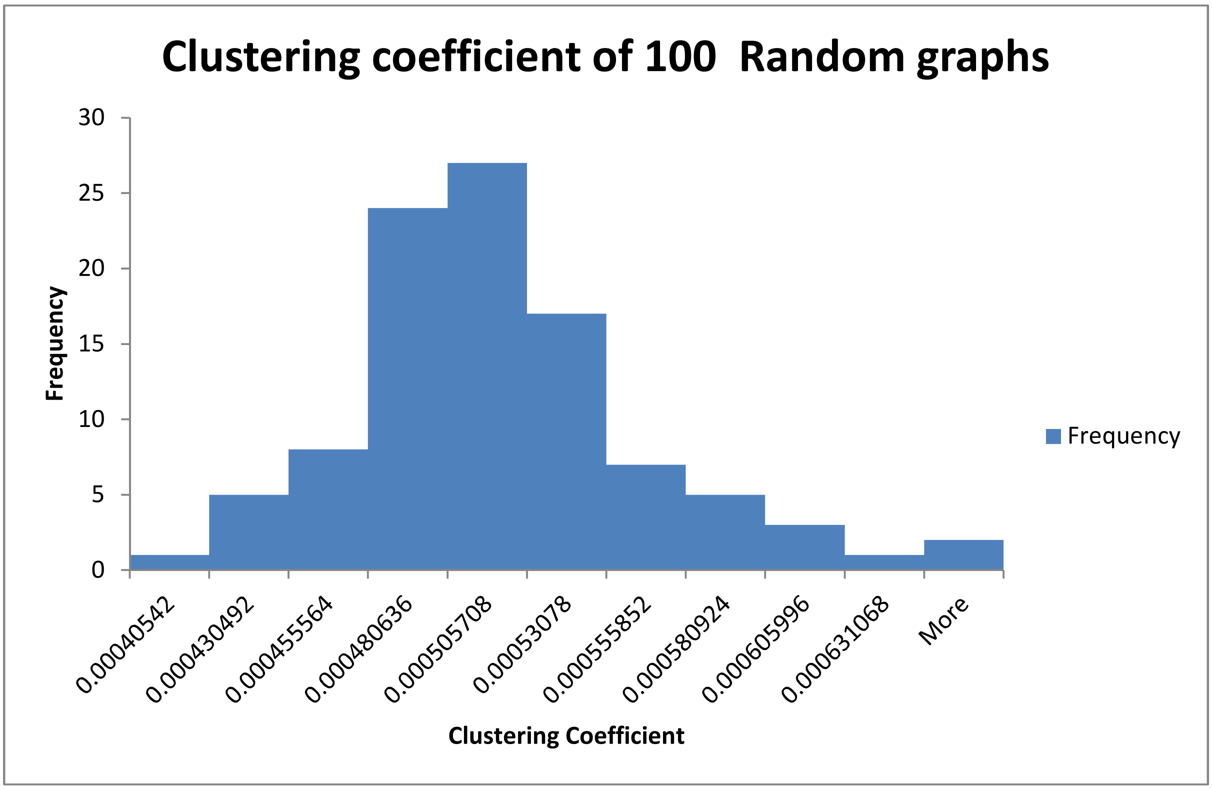
Global Clustering coefficient is defined as:
Global Clustering Coefficient = [number of triads in the graph] : [n C 3, i.e. number of ways in which you can choose 3 nodes out of n]

4.3.A. Comparison with PolBooks :

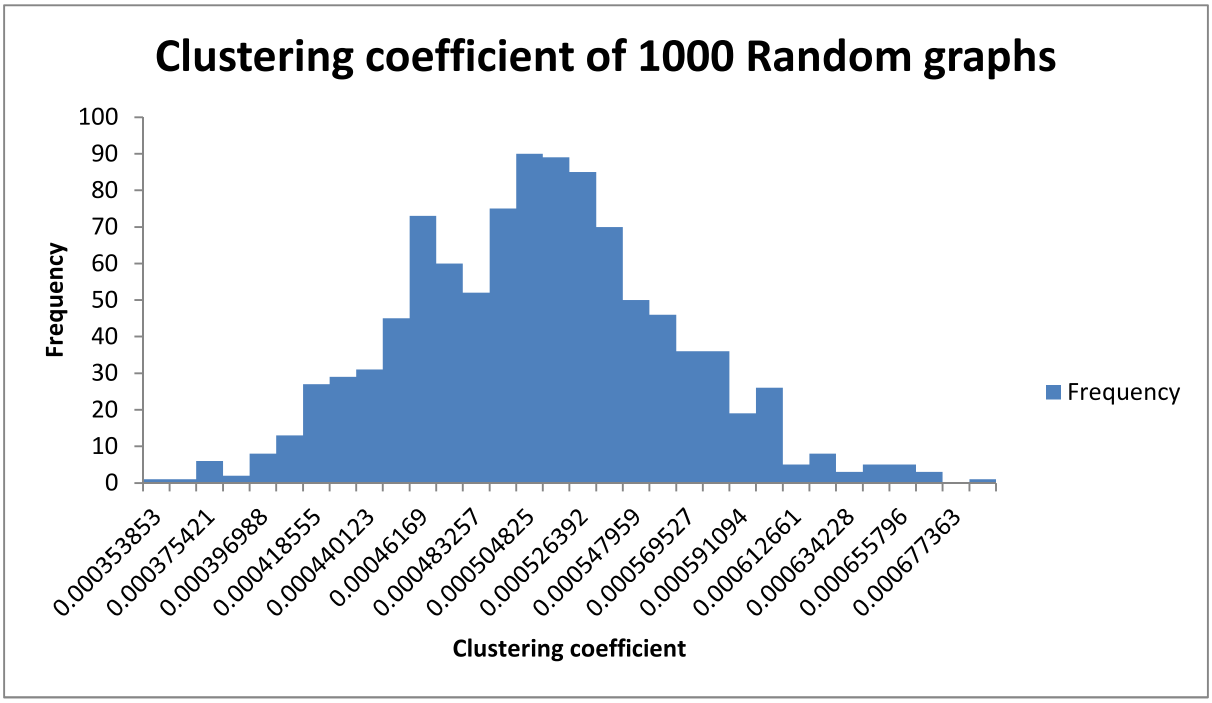
So, for our PolBooks visualisation,
Total number of nodes (n) = 105
N C 3 = 187460
Total number of triads = 560
Global clustering coefficient = (560 / 5460) = **0.002987**

Now, let us look at the clustering in some randomly generated graphs:

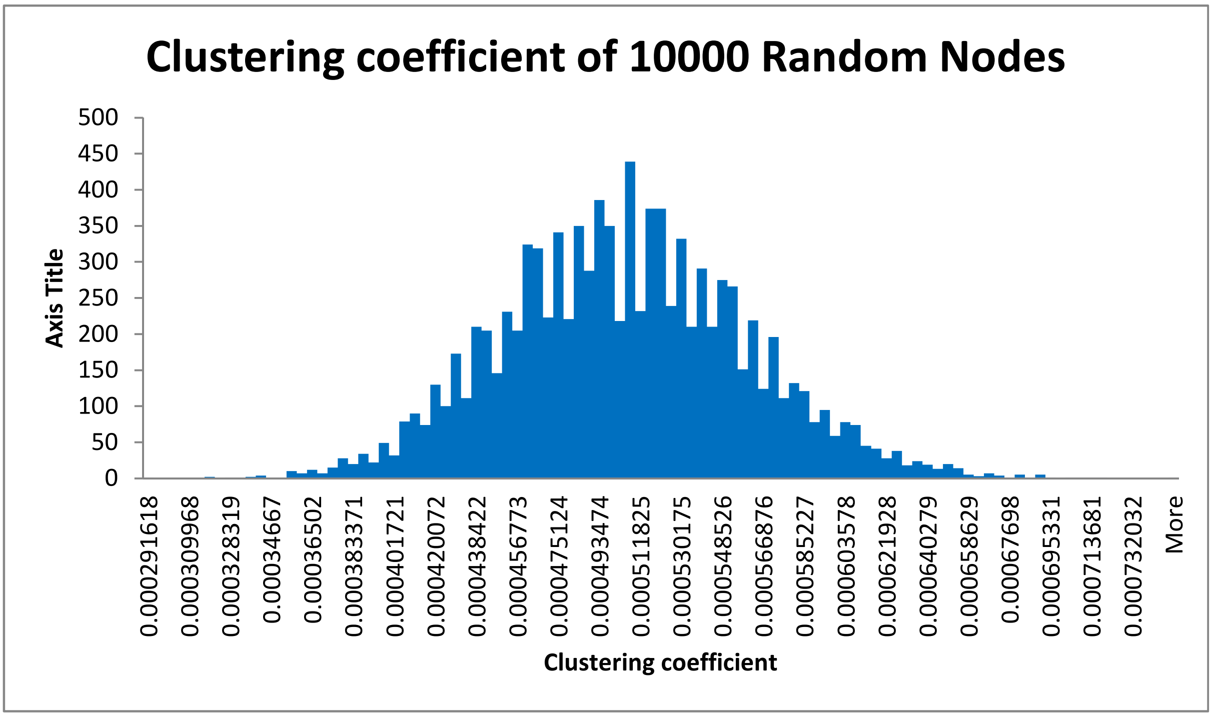
100 Random Graphs:



1,000 Random Graphs:



10,000 Random Graphs:



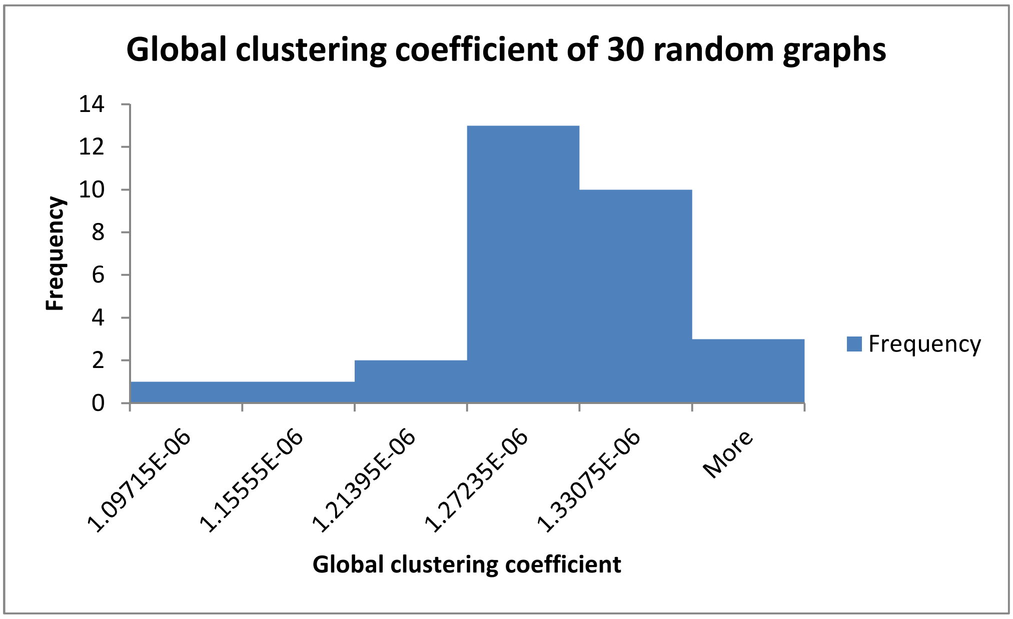
#Point to Note: We have obtained a uniformly distributed bell curve, with most probable Clustering coefficient lying around 0.000510 which is much less than that of PolBooks graph, which shows the uniqueness of the given data.

4.3.B. Comparison with PolBlogs :

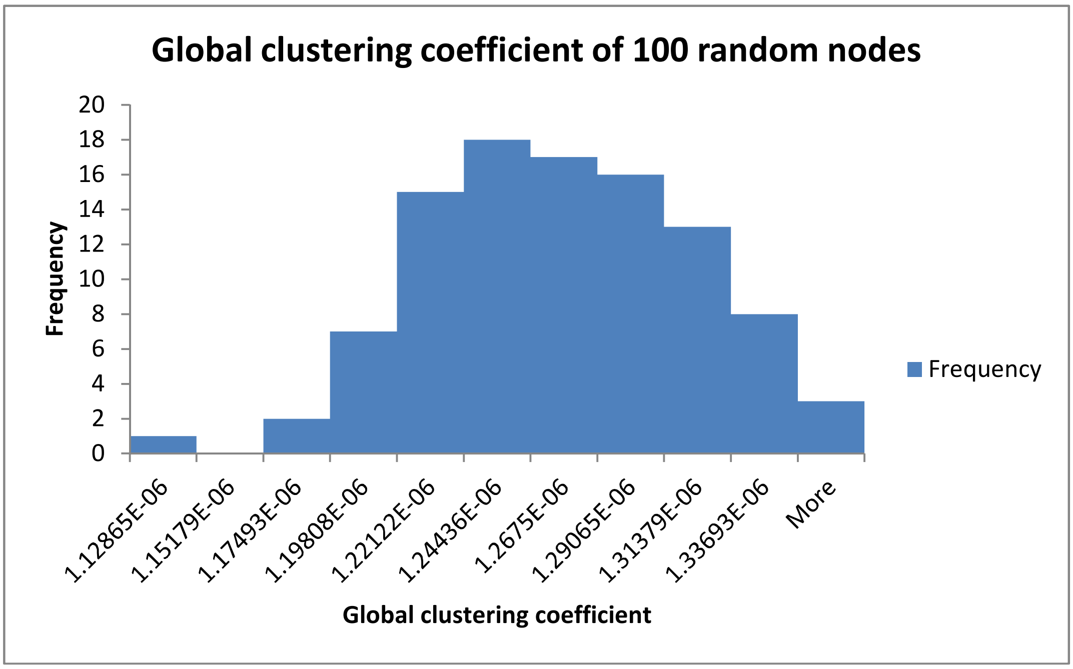
Similarly, for our PolBlogs visualisation,
Total number of nodes (n) = 1490
Total number of triads = 21654
Global clustering coefficient = 0.000280

Now, let us look at some interesting random graphs:

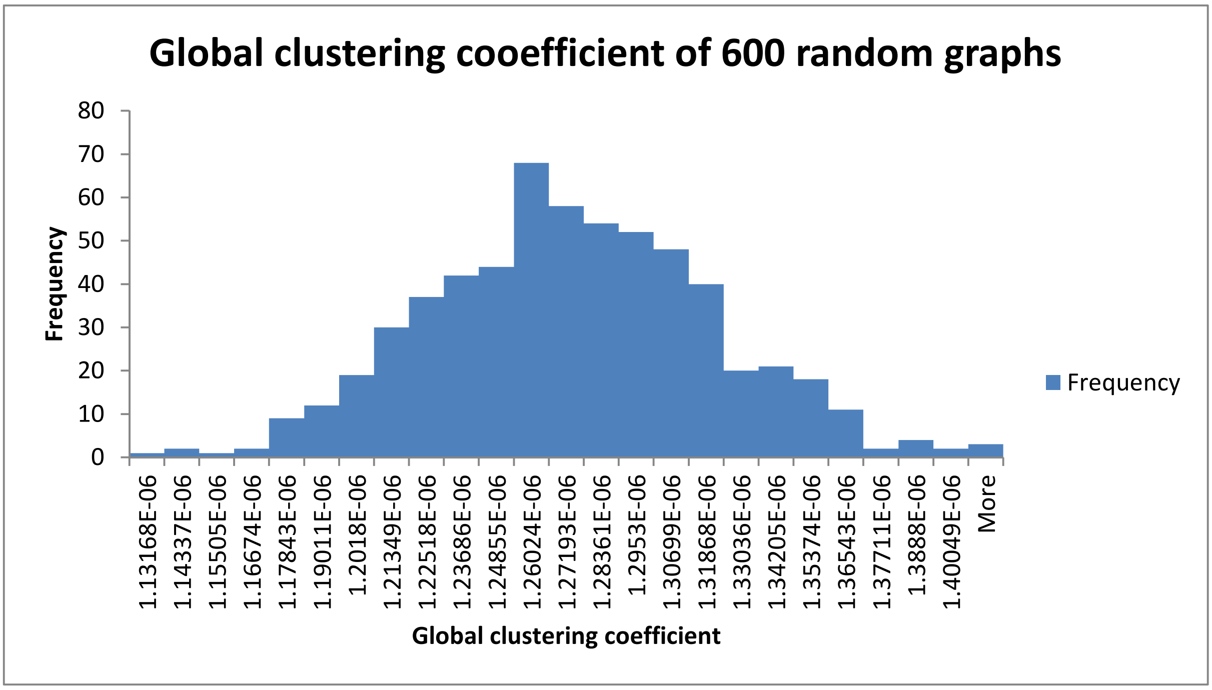
30 Random graphs:



100 Random graphs:



600 Random graphs:



#Point to Note: We have obtained a uniformly distributed bell curve, with most probable Clustering coefficient lying around 0.00000126 which is much less than that of PolBlogs graph, which shows the uniqueness of the given data.

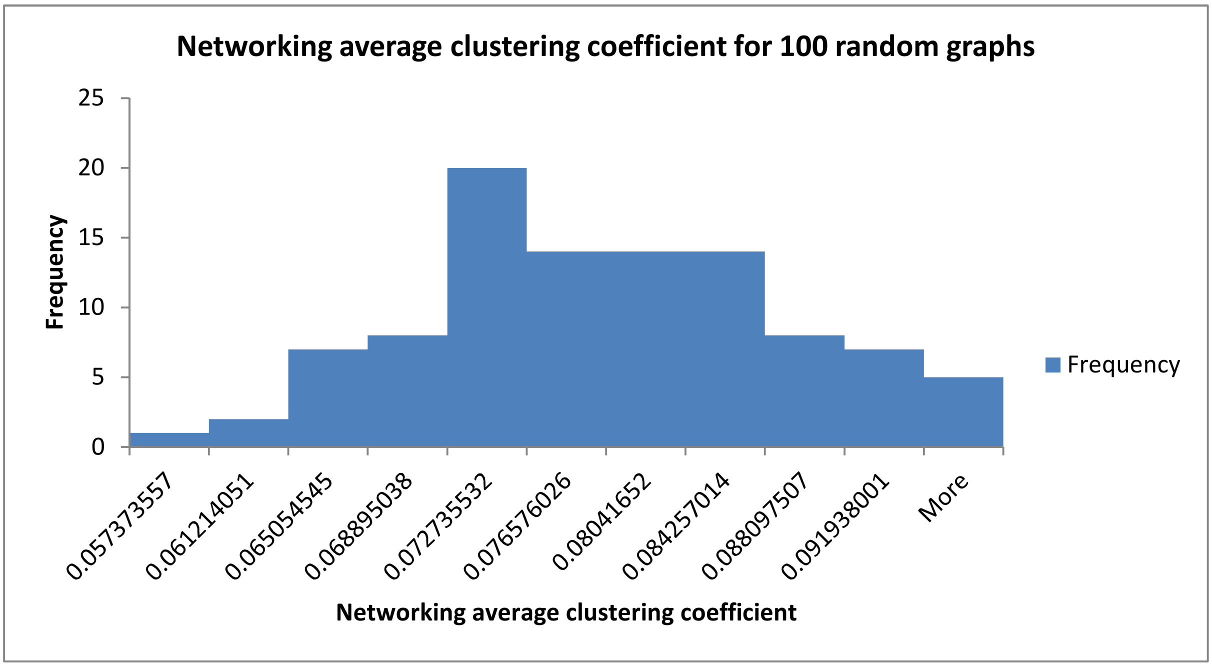
4.4. Networking average clustering coefficient: the local clustering coefficient of a node is defined as the ratio of [the number of triads involving that node]:[n C 2, where n is the number of neighbours of the node]. We first find the local clustering coefficient of each node and then take their average to obtain the networking average clustering coefficient.

4.4.A. Comparison with PolBooks :

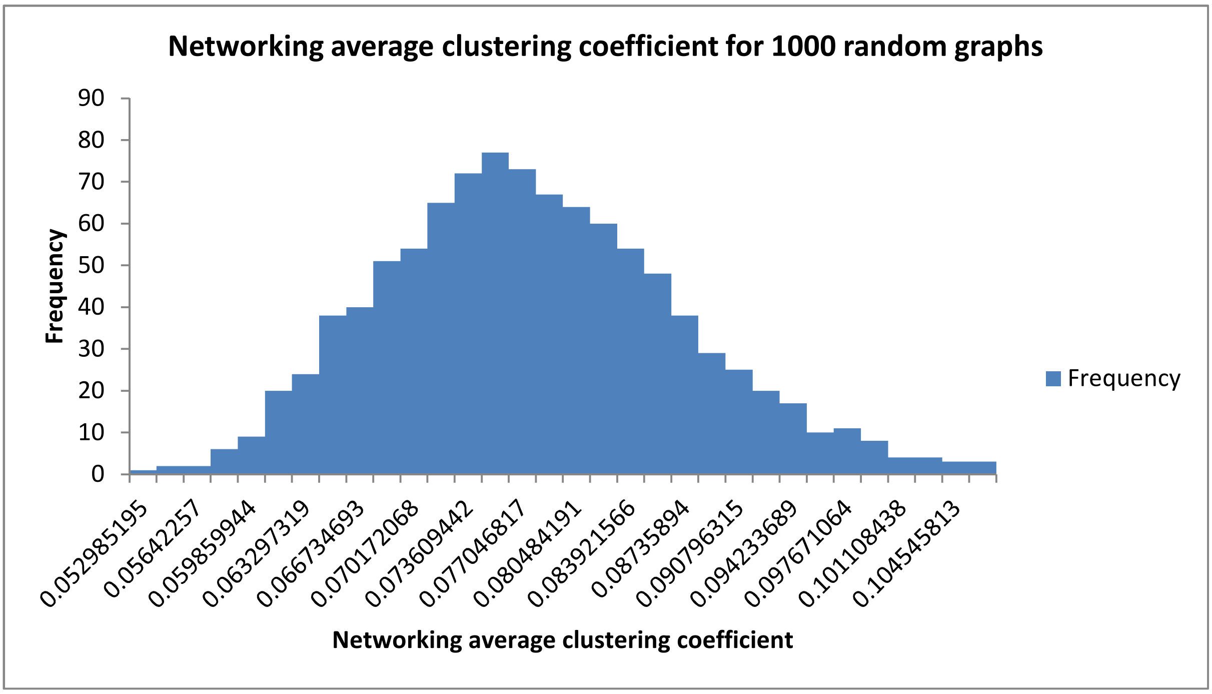
Networking average clustering coefficient= 0.4875266

Now, let us look at some of the random graphs:

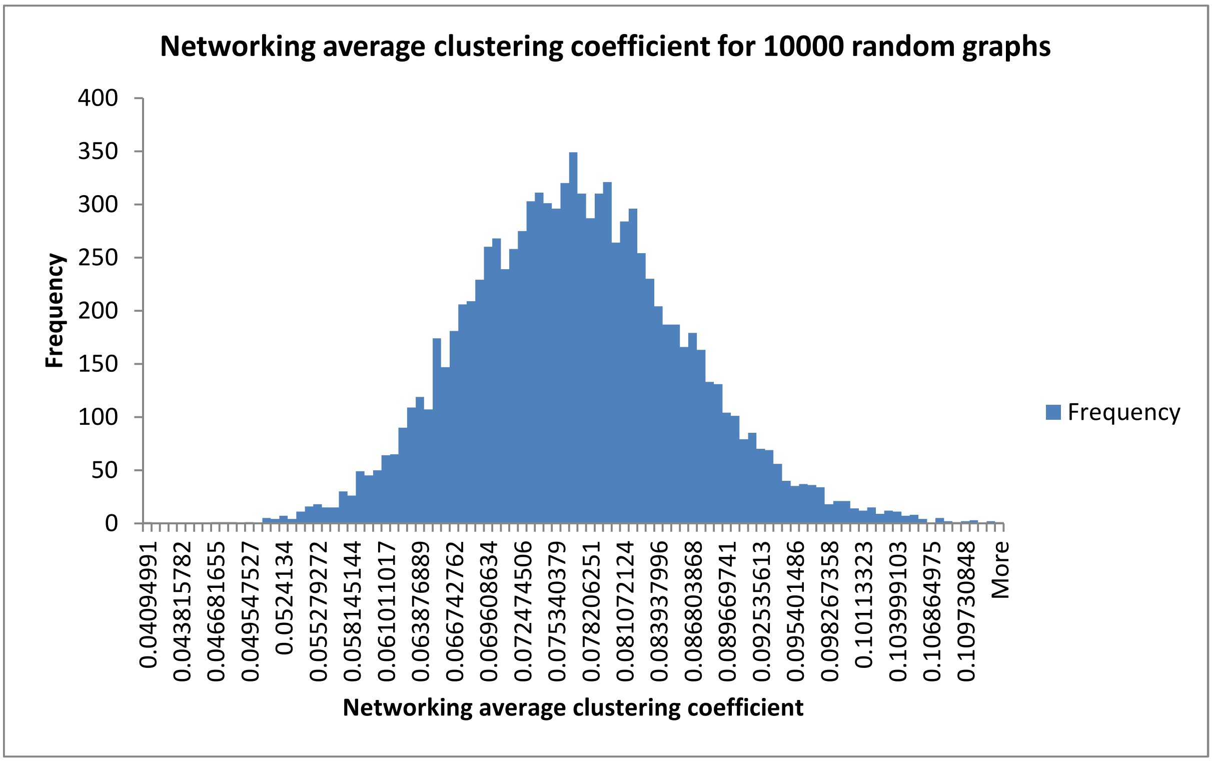
100 Random Graphs



1000 Random Graphs



10000 Random Graphs



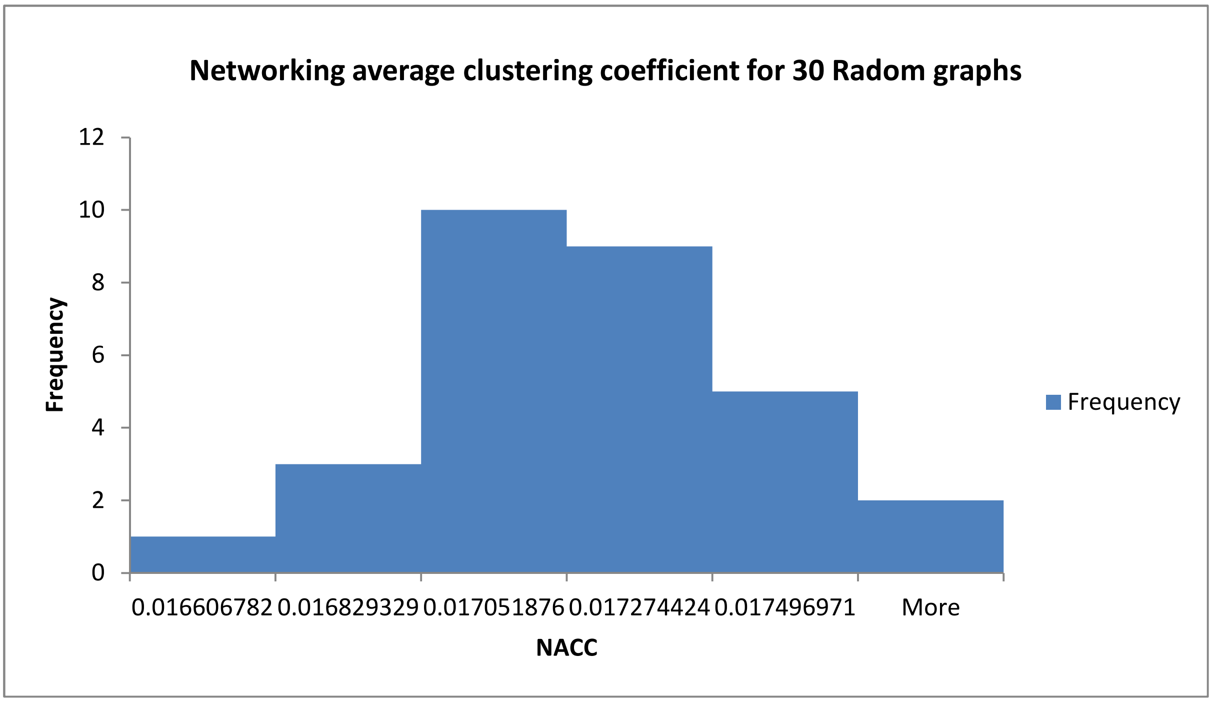
#point to note: Most probabale value of networking average clustering coefficient of Random graphs = 0.075, much less than that of PolBooks

4.4.B. Comparison with Polblogs:

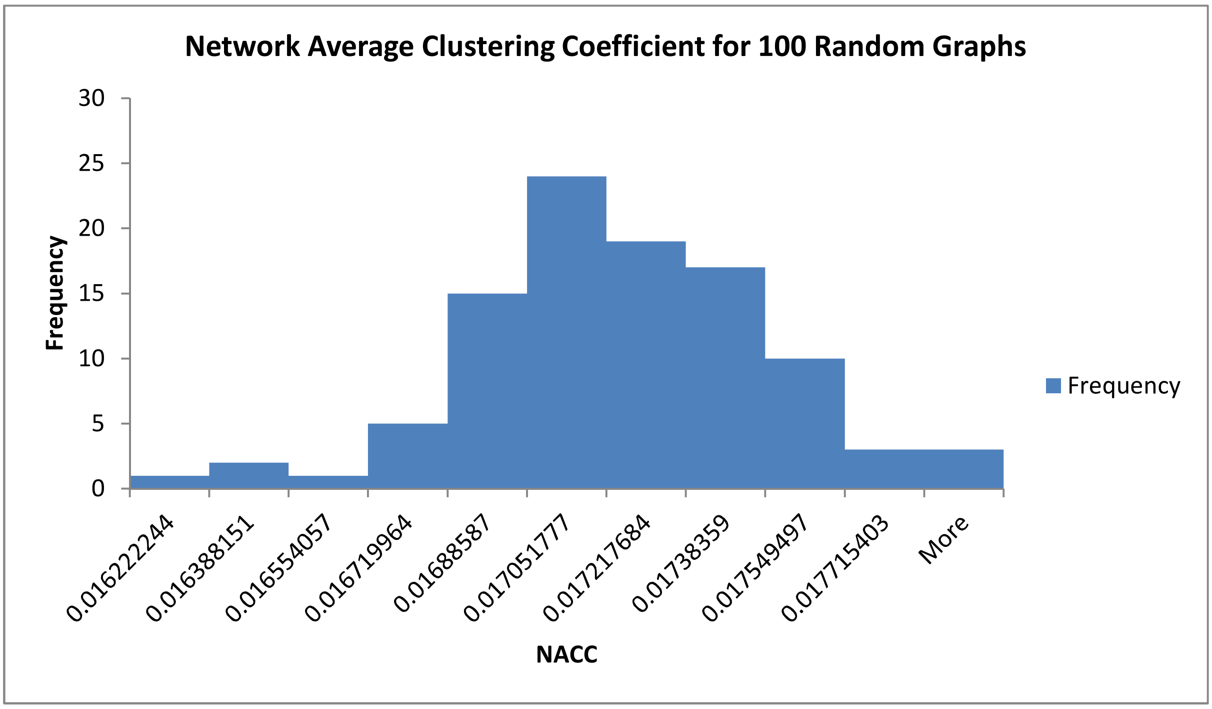
Networking average clustering coefficient = 0.13269182

Now, let us look at some random graphs:

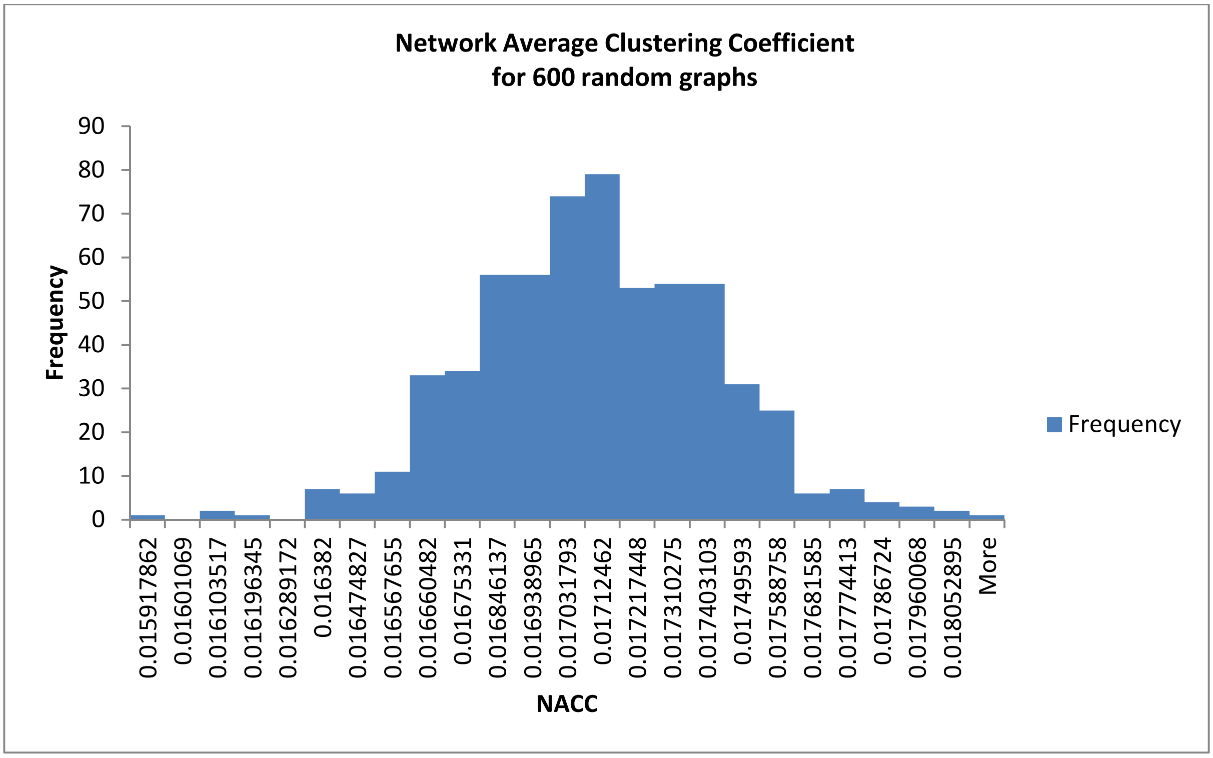
30 Random Graphs



100 Random Graphs



600 Random Graphs

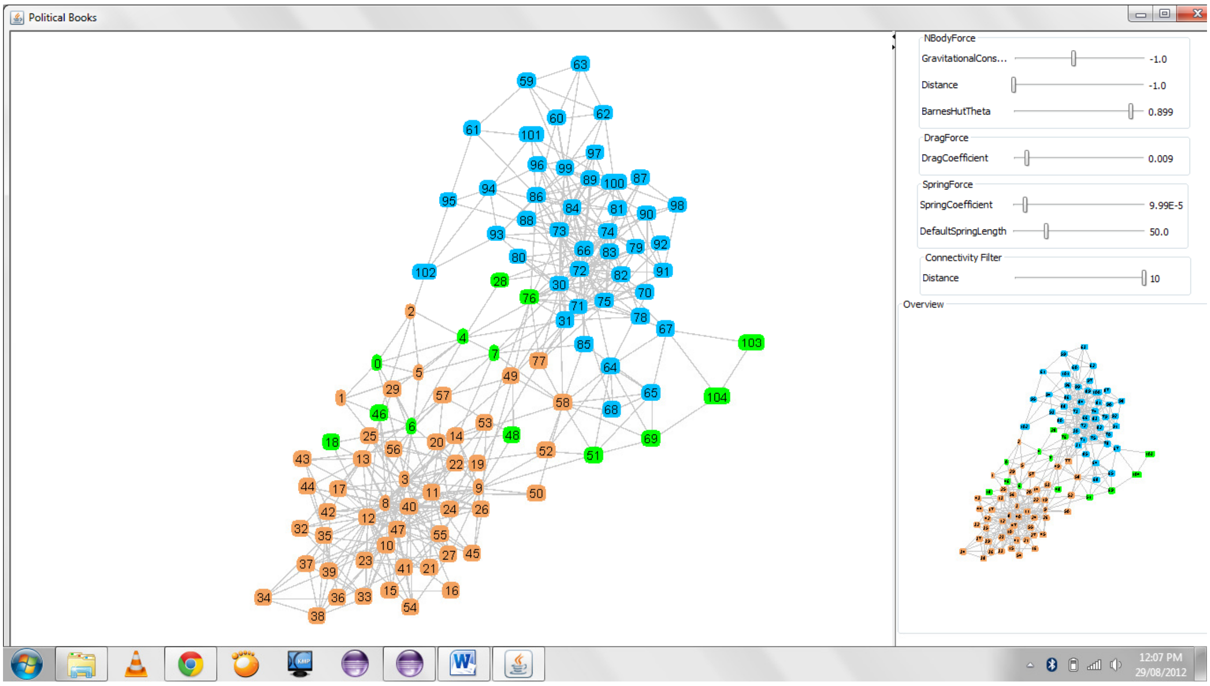


#point to note: Most probabale value of networking average clustering coefficient of Random graphs = 0.017, much less than that of PolBlogs

References:

- *API docs of Prefuse : http://prefuse.org/doc/api/*
- *Course web page :* http://act4d.iitd.ernet.in/index.php?option=com_content&view=article&id=30&Itemid=39
- *Doubt clearing :* http://piazza.com
- *https://github.com/jainshashank99/Assignment1.git*
- *Wikipedia :* http://wikipedia.org
- *Google :* http://google.co.in
- *http://office.microsoft.com*

[End of Report]



NBodyForce

GravitationalCons... -1.0

Distance -1.0

BarnesHutTheta 0.899

DragForce

DragCoefficient 0.009

SpringForce

SpringCoefficient 9.99E-5

DefaultSpringLength 50.0

Connectivity Filter

Distance 10