Elements of Network Science(CPT_S 591)

Washington State University

Project Final Report

Community Detection Using NetworkX

Submitted By:

Sharath Kumar Karnati

ID: 011852253

Under the Guidance of

Prof. ASSEFAW GEBREMEDHIN

GitHub Repository Link:

Table of Contents

1. ABSTRACT:

This project delves into the realm of community detection within networks utilizing NetworkX, a Python library. Through the analysis of two networks sourced from the Stanford dataset, various network analysis techniques are applied, including centrality and clustering coefficients, to unveil the underlying structures. The study also integrates measures learned in class to provide deeper insights into network organization.

Additionally, it conducts a comparative analysis of community detection effectiveness between NetworkX and igraph, highlighting their respective strengths and weaknesses through tabulated data. Moreover, the project extends its exploration by implementing three distinct shortest path algorithms to determine the most efficient approach. Furthermore, it includes the calculation of hubs and authorities scores to enhance the understanding of network properties.

In essence, this project presents a comprehensive investigation into community detection and network analysis, exploring different methodologies and tools to unravel the complexities of network structures and dynamics.

2.INTRODUCTION:

The problem at hand revolves around the challenge of efficiently performing network analysis tasks, such as community detection and shortest path calculations, particularly for beginners or those finding certain tools cumbersome. This issue is pertinent as network analysis plays a crucial role in various fields, including social network analysis, biological network analysis, and transportation network optimization, among others.

In response to this problem, the basic approach undertaken is to explore alternative tools that offer a user-friendly interface, swift performance, and seamless integration with coding environments. The focus shifts towards NetworkX, a Python library, which presents itself as a promising alternative to the complexities encountered with tools like igraph. Given Python's widespread usage and ease of learning, NetworkX emerges as a viable solution that can cater to both beginners and experienced practitioners alike.

The project's alignment with related work lies in its endeavor to streamline the process of network analysis by leveraging accessible tools like NetworkX. By replicating tasks performed in class and assignments using NetworkX, the project aims to demonstrate its suitability as an alternative tool for network analysis. Furthermore, the comparison between NetworkX and igraph in terms of community detection effectiveness and the utilization of various shortest path algorithms such as Dijkstra’s, PageRank, and Random Walk algorithms adds to the discourse on tool selection and performance in network analysis.

The project aims to showcase the capabilities of NetworkX in conducting comprehensive network analysis tasks. By applying centrality measures, community detection algorithms, and statistical analyses to datasets from the Stanford dataset, the project seeks to demonstrate NetworkX's proficiency in handling diverse network analysis tasks. Additionally, the comparison with igraph and the exploration of different shortest path algorithms serve to highlight the advantages and limitations of different approaches in network analysis.

In summary, my project contributes to the broader understanding of tool selection and performance evaluation in network analysis by comparing different software options and methodologies. By providing detailed results and summarizing the advantages and disadvantages of both NetworkX and igraph, I'm facilitating better-informed decisions and advancing the field of network analysis.

NetworkX:

A Python software program called NetworkX is used to create, manipulate, and research complex networks' dynamics, structure, and operation. It is employed in the study of extensive, intricate networks shown as graphs with nodes and edges. Complex networks can be loaded and stored with networkx. We can create a wide variety of random and traditional networks, examine their structure, create models of them, create new network algorithms, and sketch them.

Igraph:

igraph is a popular network analysis and graph theory library primarily written in C with interfaces for several programming languages, including Python and R. It provides a wide range of functionalities for creating, analyzing, and visualizing graphs and networks.

igraph is widely used in academia and industry for various applications such as social network analysis, biological network analysis, transportation network optimization, and many more. It offers efficient implementations of algorithms for calculating centrality measures, community detection, shortest paths, and other graph-related tasks. While igraph is a powerful tool, some users may find its interface and learning curve challenging, particularly beginners or those seeking more user-friendly alternatives.

3.PROBLEM DEFINITION:

The problem at hand revolves around the accessibility and efficiency of network analysis tools, particularly for individuals new to the field or those encountering difficulties with existing tools. Specifically, the project aims to address the following challenges:

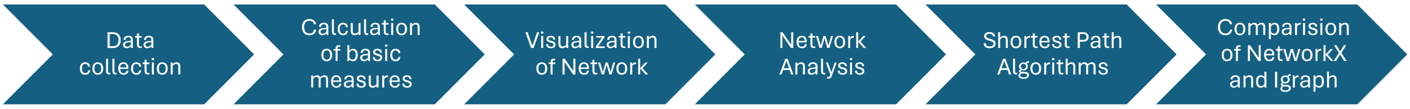
1.Accessibility: Many network analysis tools, such as igraph, may have a steep learning curve and complex interfaces, making them less accessible to beginners. This poses a barrier to entry for individuals seeking to perform network analysis tasks, such as community detection and shortest path calculations.

2.Efficiency: Even for experienced practitioners, certain tools may lack the performance and speed required for large-scale network analysis tasks. This inefficiency can hinder productivity and limit the exploration of complex network structures.

The significance of these challenges lies in the widespread application of network analysis across various domains, including social networks, biology, and transportation. Efficient and accessible tools are essential for researchers, practitioners, and students alike to unlock insights from network data and address real-world problems effectively.

By exploring alternative tools like NetworkX, which offers a user-friendly interface and seamless integration with Python, the project seeks to provide a practical solution to these challenges. Demonstrating the effectiveness of NetworkX in performing network analysis tasks and comparing it with established tools like igraph can shed light on the advantages and limitations of different approaches. Additionally, by providing detailed results and insights, the project aims to empower individuals with the tools and knowledge necessary to navigate and analyze complex networks efficiently.

4.PROJECT FLOW:



5) DATASET:

A) Email Eu core network dataset:

This network is based on email communication within a European research institution’s core member. Each node represents a member, and an edge between two nodes means they exchanged at least one email. The dataset only includes internal communication within the institution.

It comprises 1005 nodes and 25571 edges, with the largest connected component containing 986 nodes and 25552 edges. There are 803 nodes and 24729 edges in the largest strongly connected component. The average clustering coefficient is 0.3994, indicating the tendency of nodes to form clusters. There are 105461 triangles in the network, with 10.85% of them being closed. The longest shortest path between nodes is 7, and the 90th percentile of effective diameter is 2.9.

In simple terms, this network shows how members of a research institution communicate via email. It's highly connected, with most members reachable within a few steps. Nodes tend to form clusters, suggesting cohesive groups within departments.

Link : https://snap.stanford.edu/data/email-Eu-core.html

Basic Parameters of the Dataset:

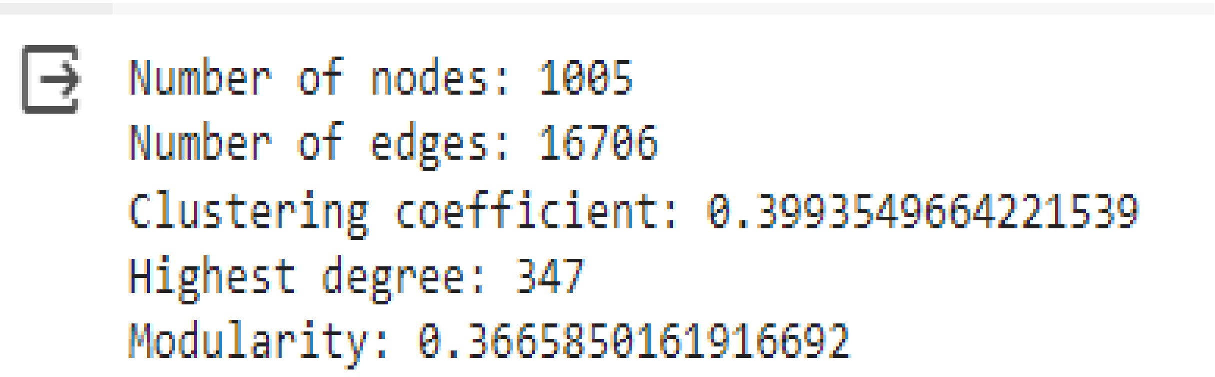
1)Number of Nodes: Indicates how many individuals (members of the institution) are in the network. Each person is represented as a node.

2)Number of Edges: Reveals the total connections (email exchanges) between individuals. An edge signifies communication between two members.

3)Clustering Coefficient: Reflects how closely connected members are within the institution. It shows how much individuals tend to communicate with others within their department or group.

4)Highest Degree: Highlights the member with the most connections (emails exchanged). This person may play a significant role in communication within the institution.

5)Modularity: Quantifies how the network divides into smaller communities or departments. It measures how tightly knit these groups are internally compared to connections outside their group.



```
Number of nodes: 1005
Number of edges: 16706
Clustering coefficient: 0.3993549664221539
Highest degree: 347
Modularity: 0.3665850161916692
```

Figure 1 : Basic Parameters of this Dataset.

Dataset Visualization:

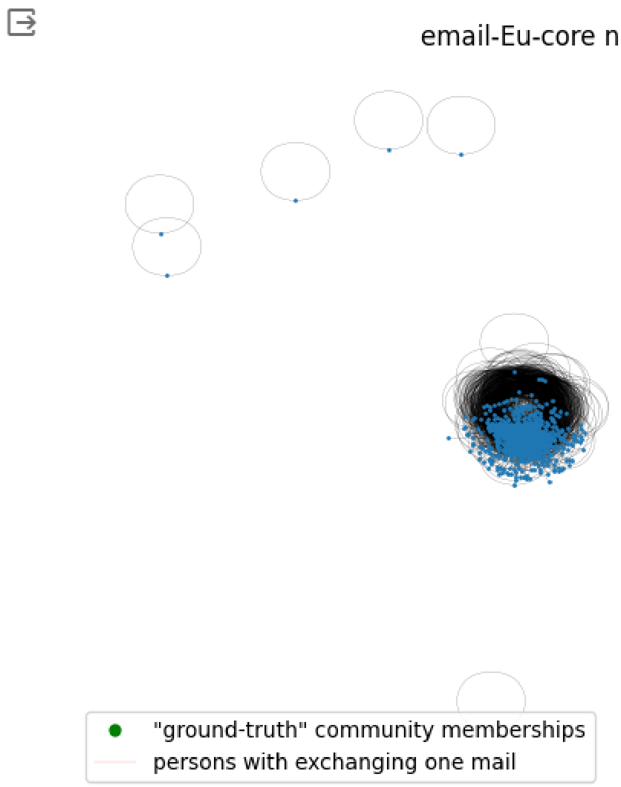

email-Eu-core network
● "ground-truth" community memberships
persons with exchanging one mail

Figure 2: Visualization of email-Eu-core Network

B) Twitter interaction network of US congress dataset:

This network depicts how members of the 117th United States Congress interact on Twitter, covering both the House of Representatives and the Senate. Data was gathered through Twitter's API, and interactions like retweets, quote tweets, replies, and mentions were analyzed to determine the likelihood of communication between members. Directed edges show the direction of interaction. Node and edge features weren't included in the dataset.

It consists of 475 nodes representing Congress members and 13,289 edges representing interactions between them. The data shows who is interacting with whom and how often. Each edge indicates a specific type of interaction between two members. The network provides insights into how politicians engage with each other on social media, shaping public discourse and political dynamics. This dataset allows researchers to delve into the intricate web of connections and communication patterns within the Congress, shedding light on its digital

link: https://snap.stanford.edu/data/congress-twitter.html

Basic Parameters of this dataset:

1.Number of Nodes: Indicates how many members of Congress are in the network. Each member is represented as a node.

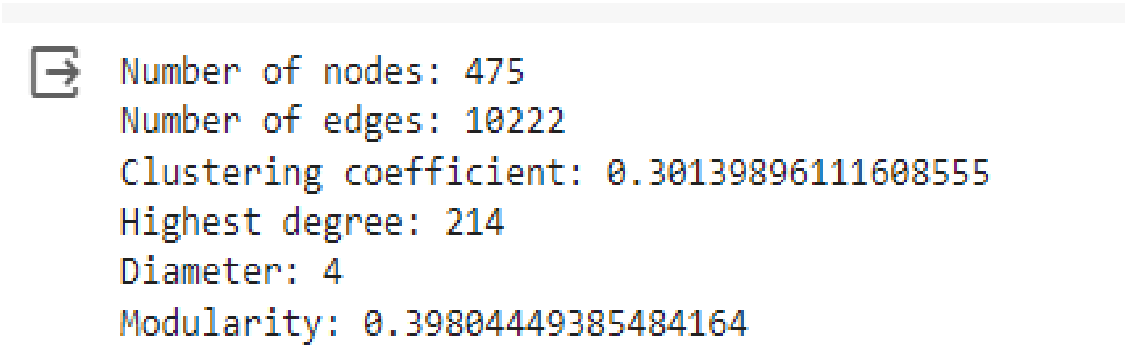
2.Number of Edges: Reveals the total interactions (such as retweets, mentions, replies) between members. An edge signifies a communication or engagement between two members.

3.Clustering Coefficient: Reflects how interconnected members are within Congress. It measures the tendency for members to engage with others who are also connected to their connections.

4.Highest Degree: Highlights the member with the most interactions within the network. This could represent a highly active or influential member of Congress.

5.Diameter: Represents the longest shortest path between any two members in the network. It shows the maximum number of steps required for a message to reach from one member to another.

6.Modularity: Quantifies how the network is divided into smaller groups of members who interact more among themselves than with those outside their group. It measures the structure and organization of Congress into distinct subgroups based on interaction patterns.



```
Number of nodes: 475
Number of edges: 10222
Clustering coefficient: 0.30139896111608555
Highest degree: 214
Diameter: 4
Modularity: 0.39804449385484164
```

Figure 3: parameters of twitter congress dataset

Visualization of the Twitter Congress network:

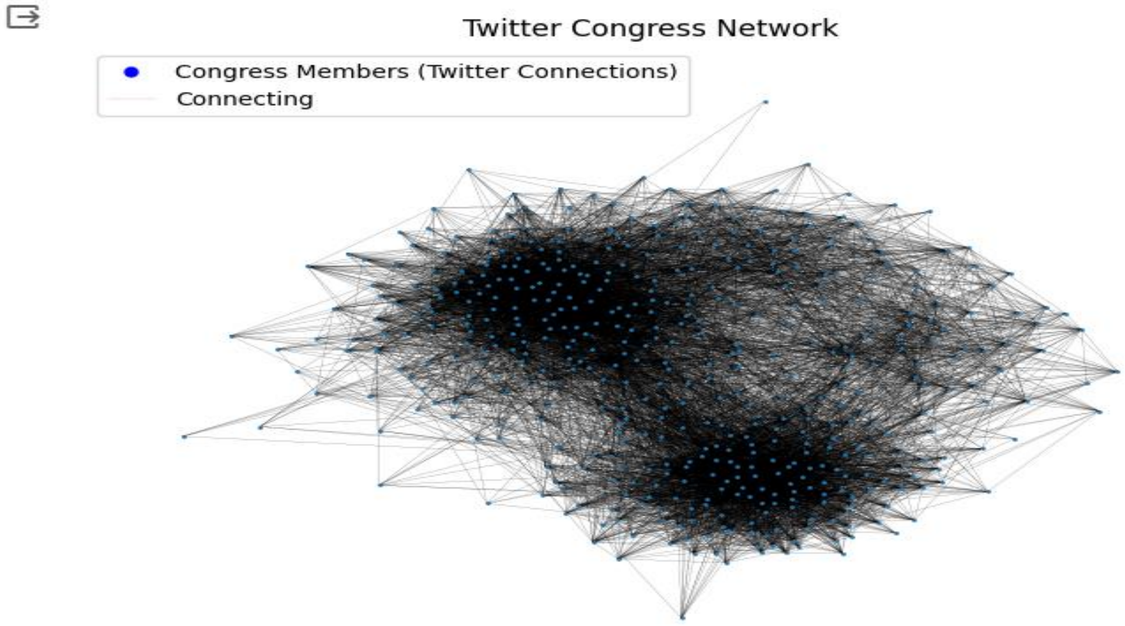


Figure 4: Twitter congress network visualization

6) NETWORK ANALYSIS:

I am planning to analyze the network by looking for groups of closely connected nodes (community detection), identifying the most influential nodes and their importance (centrality measures), and understanding how the connections between nodes are distributed (degree distribution). This analysis will help uncover the network's structure, key players, and how information or influence may propagate within it.

A) Centrality Measures:

- Degree centrality can reveal how connected each member of the institution is by counting the number of email exchanges they have. A member with high degree centrality communicates frequently with many others, potentially influencing the flow of information within the institution.
- Betweenness centrality helps identify members who act as crucial bridges between different departments or groups. These individuals play a key role in facilitating communication across the institution, making it easier for information to flow between different parts of the network.
- Eigenvector centrality identifies influential members who are connected to other influential members. These individuals have the potential to shape the communication dynamics within the institution, potentially influencing decision-making processes and information dissemination.

Overall, these centrality measures can provide insights into how information flows within the institution's email network. They help identify individuals who are popular, function as bridges between groups, and wield influence, thus influencing how members navigate and communicate within the network.

For Email-Eu-core network:

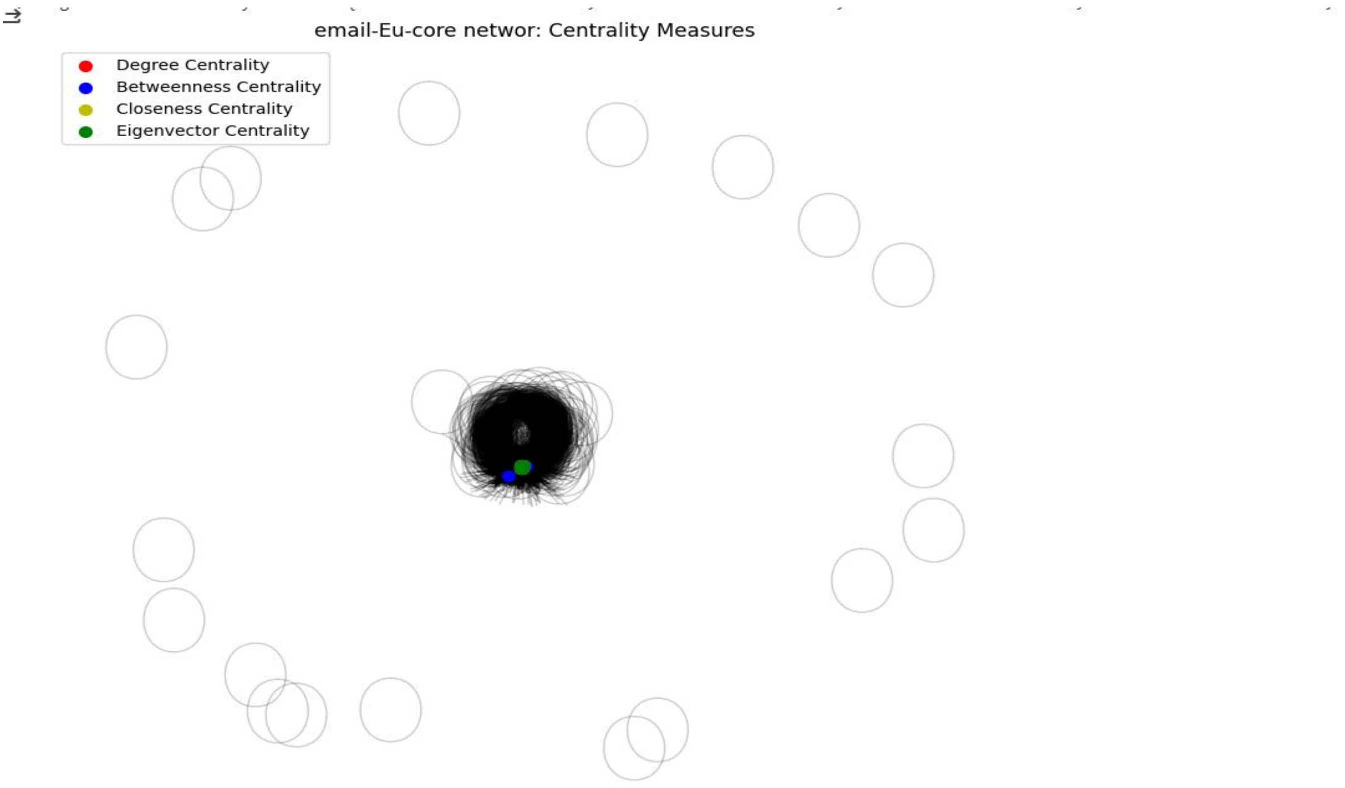


Figure5: Centrality Measures

In the analysis of the Email EU core network, centrality measures were used to assess network accesability. The top 10 nodes with the highest degree, betweenness, closeness, and eigenvector centrality values were identified and highlighted on a graph. Each centrality measure provides unique insights into node importance: degree centrality reflects connectivity, betweenness centrality indicates frequent use in shortest paths, closeness centrality signifies proximity to other nodes, and eigenvector centrality suggests connections to other important nodes. By examining these measures, we gain a better understanding of how easily information flows and how nodes are interconnected within the Email EU core network.

For Twitter congress network:

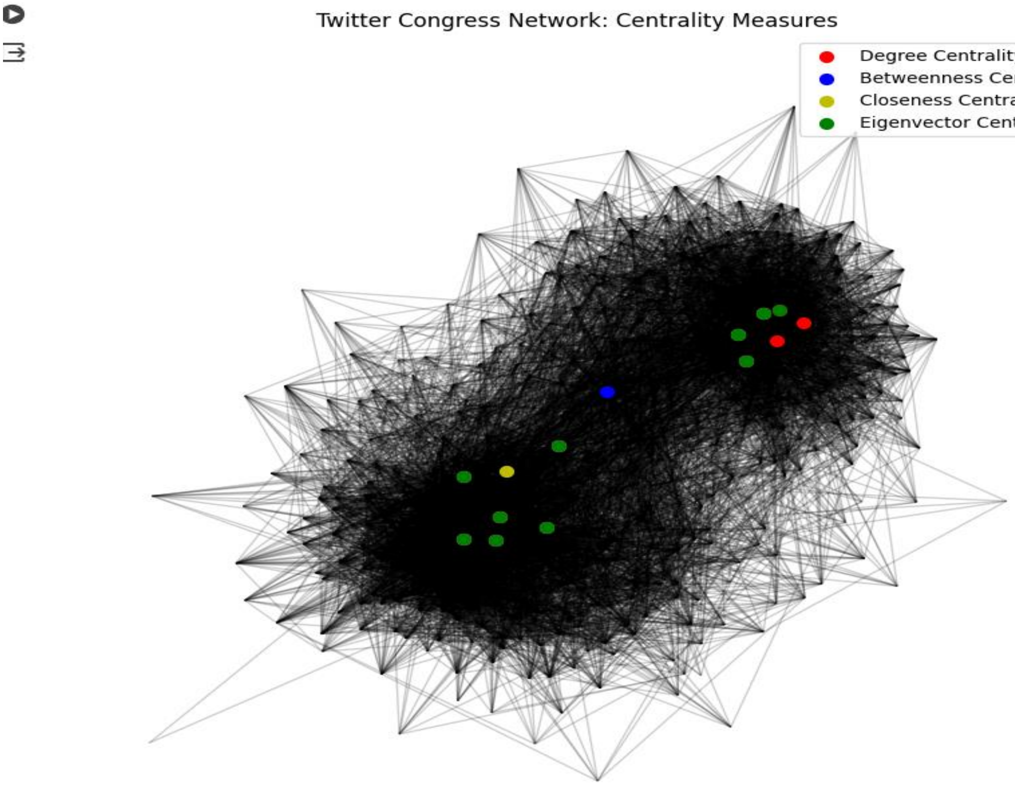


Figure6: Centrality Measures

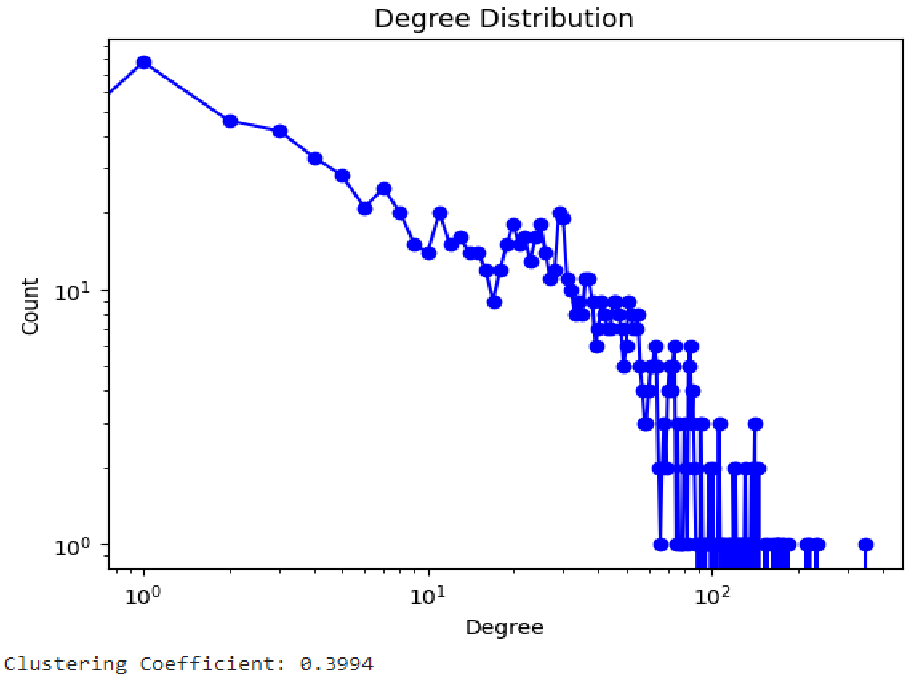
In the analysis of the Twitter Congress network, centrality measures were employed to assess network accessability. The top 10 nodes with the highest degree, betweenness, closeness, and eigenvector centrality values were identified and visually represented on a graph. Each centrality measure offers distinct insights into node importance: degree centrality reflects connectivity, betweenness centrality indicates frequent involvement in shortest paths, closeness centrality suggests proximity to other nodes, and eigenvector centrality signifies connections to other influential nodes. Examining these measures aids in understanding how information spreads and how nodes are interconnected within the Twitter Congress network.

B)Degree Distribution and Clustering Coefficient :

For both the email EU core network and the Twitter Congress network, degree distribution and clustering coefficient serve as crucial metrics for understanding their accesability.

Degree distribution reveals the frequency of nodes with a specific number of connections, offering insights into how nodes are interconnected. In the email EU core network, it helps gauge the prominence of individual members and their connections. Similarly, in the Twitter Congress network, degree distribution aids in understanding the popularity of congress members and their interactions.

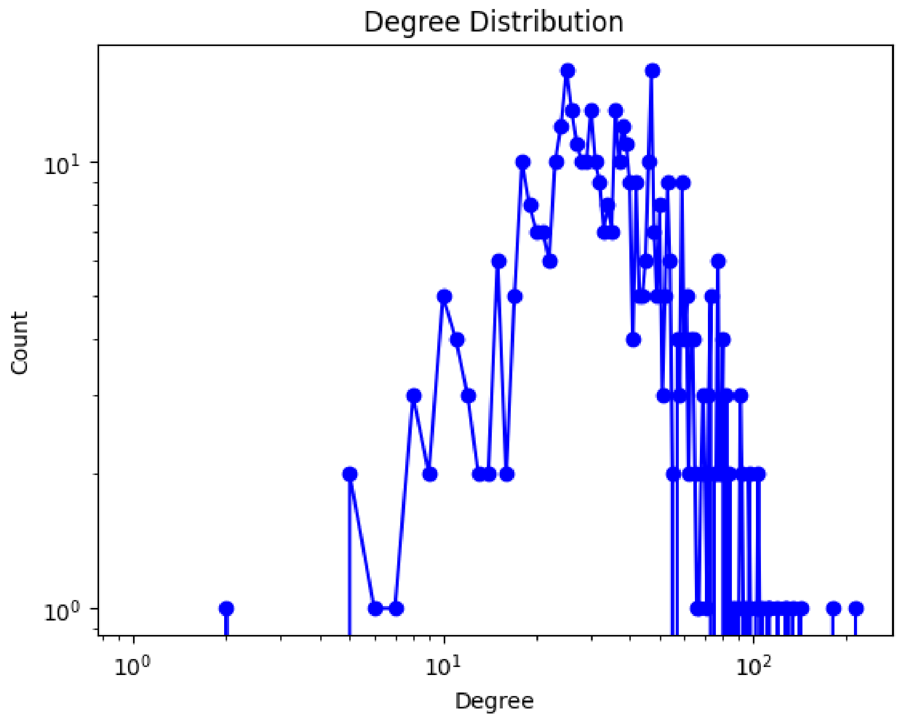
The clustering coefficient measures how tightly connected nodes are within local clusters. In the email EU core network, it indicates the cohesion within departments or research groups. In contrast, in the Twitter Congress network, the clustering coefficient reflects the formation of cliques or ideological clusters among legislators. Both metrics contribute to comprehending the network's structure and the likelihood of alternative pathways or communication channels.



```
Clustering Coefficient: 0.3994
```

Figure 7: Degree distribution and clustering coefficient for Email-eu-core network

In the analysis of the email network, the degree distribution plot demonstrates a power-law distribution, indicating the presence of a few highly connected nodes (hubs) and many nodes with fewer connections. This pattern is typical in complex networks and suggests the existence of central nodes that play significant roles. With a clustering coefficient of 0.3994, the network exhibits a moderate level of interconnectedness, implying the formation of cohesive groups or "communities" where nodes are tightly interconnected.



```
Clustering Coefficient: 0.3014
```

Figure 8: Degree distribution and clustering coefficient for Twitter Congress network

Here we can observe that the clustering coefficient is 0.3014 which is moderate, which tells us that the network exhibits a moderate level of interconnectedness, implying the formation of cohesive groups or "communities" where nodes are tightly interconnected.

C)Community Detection:

Community detection is like finding groups of friends in a big social gathering where people naturally form cliques. It helps us understand how things are organized within a network, like who hangs out with whom.

There are diverse ways to find these groups, like playing a game where you try to break friendships. One popular method, the Girvan-Newman algorithm, looks at how many "middlemen" are between people. It removes the busiest connections until the network breaks into smaller groups.

Another method, the Louvain algorithm, looks at how well groups stick together. It keeps reshuffling people between groups until it finds the best arrangement where everyone feels connected.

Then there is the Infomap algorithm, which tries to find the simplest way to describe the network. It looks for the most efficient way to represent the connections between people.

The label propagation algorithm is like giving everyone stickers with their group name and seeing which stickers spread the most. People who end up with the same stickers are part of the same group.

Finally, the spectral clustering algorithm looks at how people are connected through shared friends. It uses a fancy math trick to group people based on their connections.

Each method has its own pros and cons, and the best one to use depends on what you're trying to find out about the network and its structure.

I have used Louvain algorithm for the Community Detection:

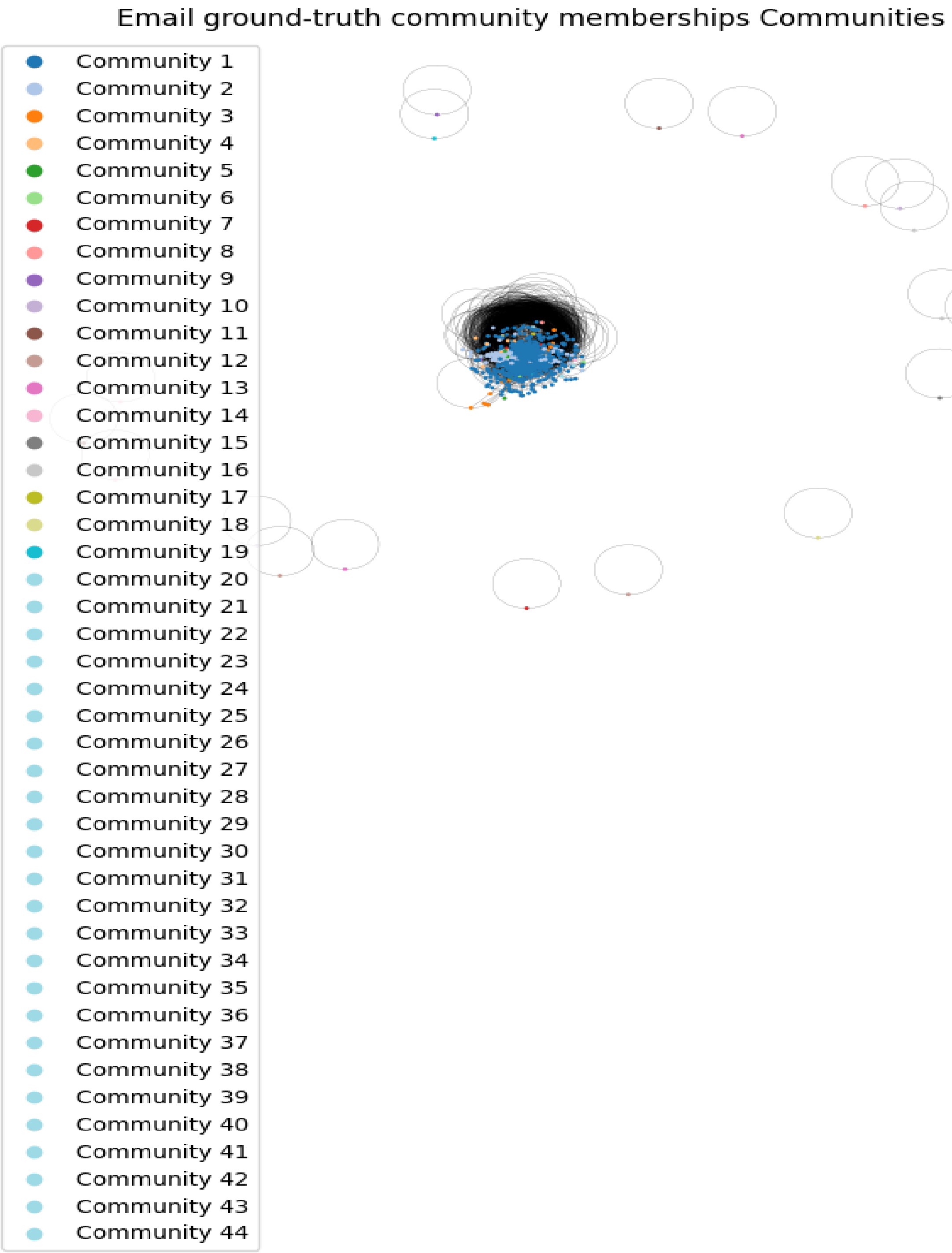


Figure9: Community detection of Email-eu-core network

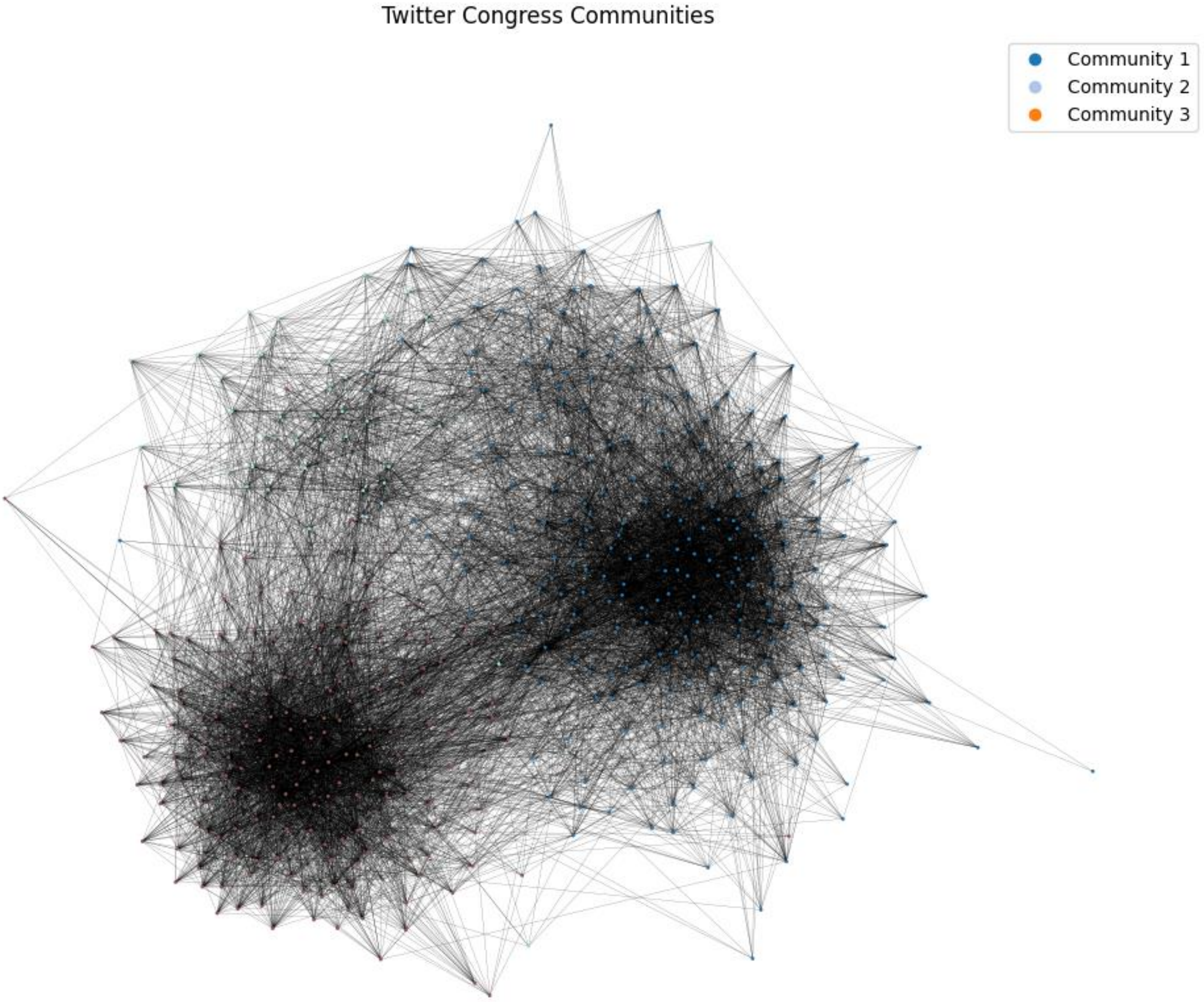


Figure10: Community detection of Twitter Congress network

6.2.)SMALL-WORLD PHENOMENA:

In many social networks, any two people can be connected by a short chain of social connections, frequently consisting of only a few mediators. This phenomenon is known as the "small world phenomenon." The concept of "six degrees of separation," which postulates that any two individuals can be connected by a chain of six or fewer social relationships, helped bring this theory to a broad audience.

This idea is not exclusive to social networks; it also holds true for a number of other kinds of networks, like those used for communication or transportation. The highly linked people in the network, sometimes referred to as "hubs," are the reason behind the small world phenomena. These hubs allow the creation of brief pathways between nodes that would otherwise be far apart by acting as bridges between various clusters or groups.

In simpler terms, the small world phenomenon means that even in large networks like the email communication network within the research institution, people are surprisingly intricately connected. There are usually just a few degrees of separation between any two individuals. This happens because some individuals have many connections and function as bridges, bringing distinct parts of the network closer together.

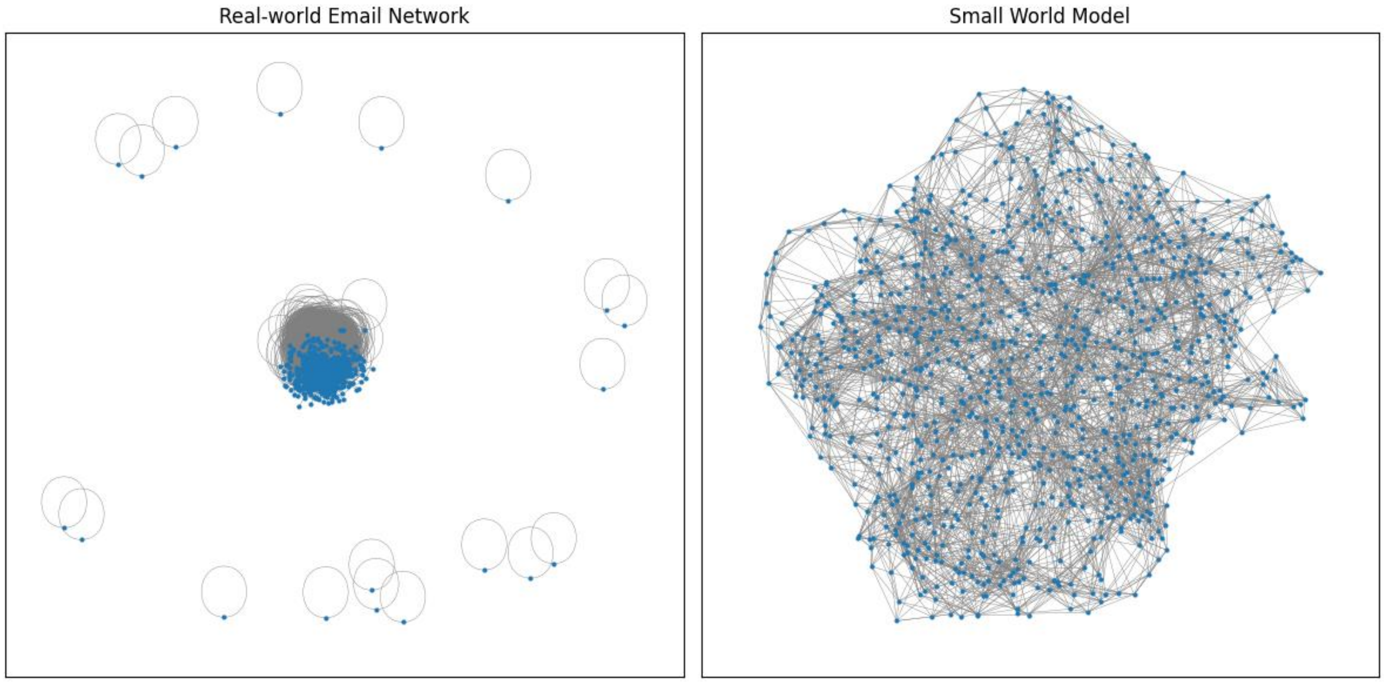
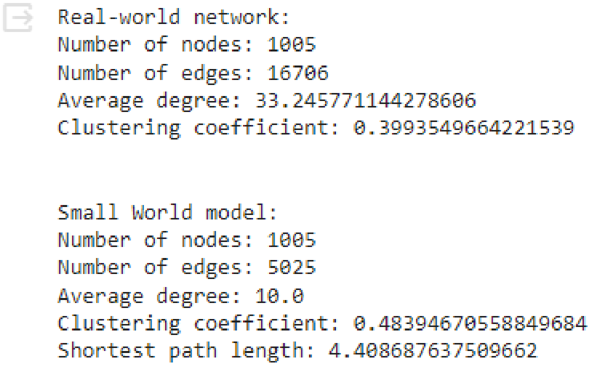


Figure 11: Real-world network vs small world networks visualizations



```
Real-world network:
Number of nodes: 1005
Number of edges: 16706
Average degree: 33.245771144278606
Clustering coefficient: 0.3993549664221539


Small World model:
Number of nodes: 1005
Number of edges: 5025
Average degree: 10.0
Clustering coefficient: 0.48394670558849684
Shortest path length: 4.408687637509662
```

Figure 12 : Parameters in small world network bs real world networks

In the real-world network of email eu core , which likely represents a communication network, there are 16706 connections among 1005 nodes, with an average of 33 connections per node. The clustering coefficient of 0.399 indicates a moderate level of local interconnectedness, suggesting the presence of community-like structures.

In contrast, the Small World model has fewer connections, with 5025 edges among the same 1005 nodes, resulting in an average of 10 connections per node. However, despite the lower connectivity, the clustering coefficient is higher at 0.484, indicating a higher level of local clustering compared to the real-world network.

This suggests that while the Small World model may have fewer connections overall, nodes are more likely to form tightly knit clusters or communities, potentially facilitating faster communication or information spread within these clusters.

For twitter congress network:

In simpler terms, the small world phenomenon means that even within the Twitter interaction network of the 117th United States Congress, members are likely to be connected through a few intermediaries. This happens because some members have many connections and serve as bridges between different groups, making communication and interaction more efficient across the network.

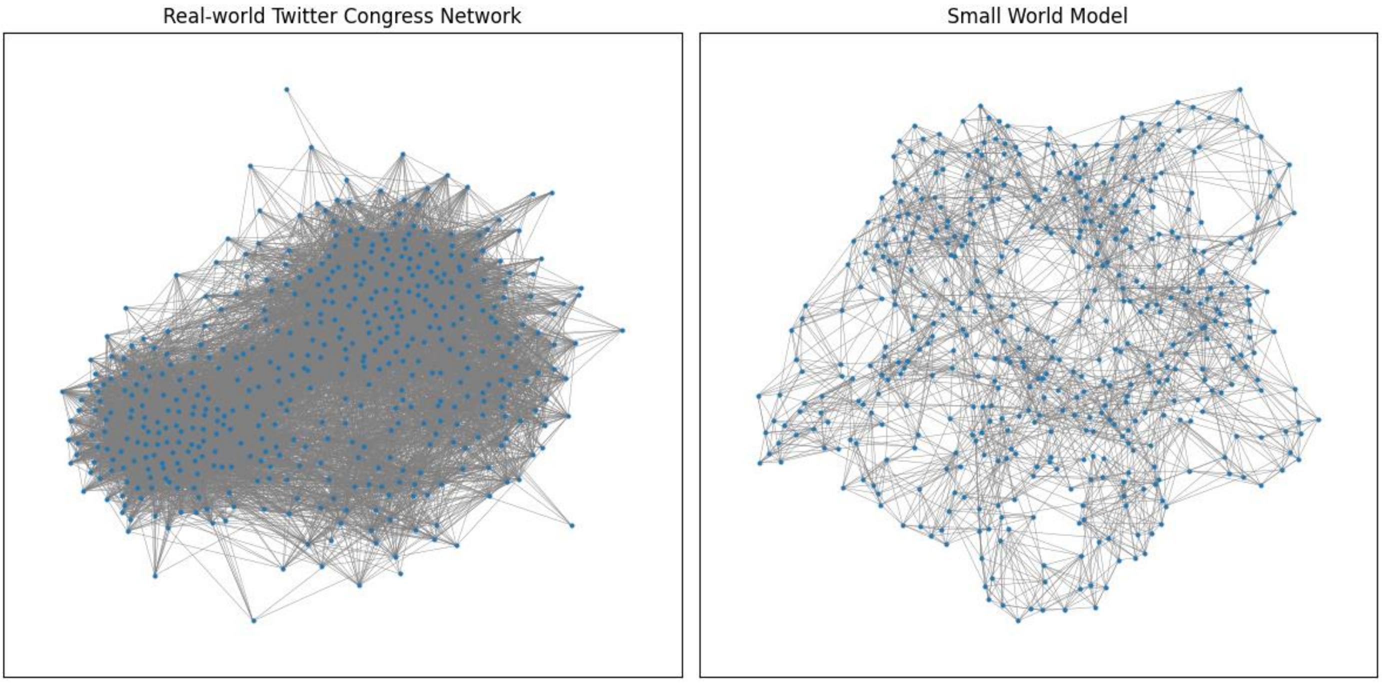
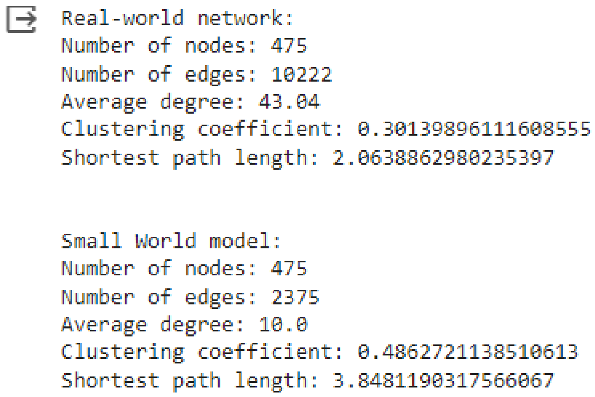


Figure 13 : Real-world network vs small world networks visualizations



```
Real-world network:
Number of nodes: 475
Number of edges: 10222
Average degree: 43.04
Clustering coefficient: 0.30139896111608555
Shortest path length: 2.0638862980235397


Small World model:
Number of nodes: 475
Number of edges: 2375
Average degree: 10.0
Clustering coefficient: 0.4862721138510613
Shortest path length: 3.8481190317566067
```

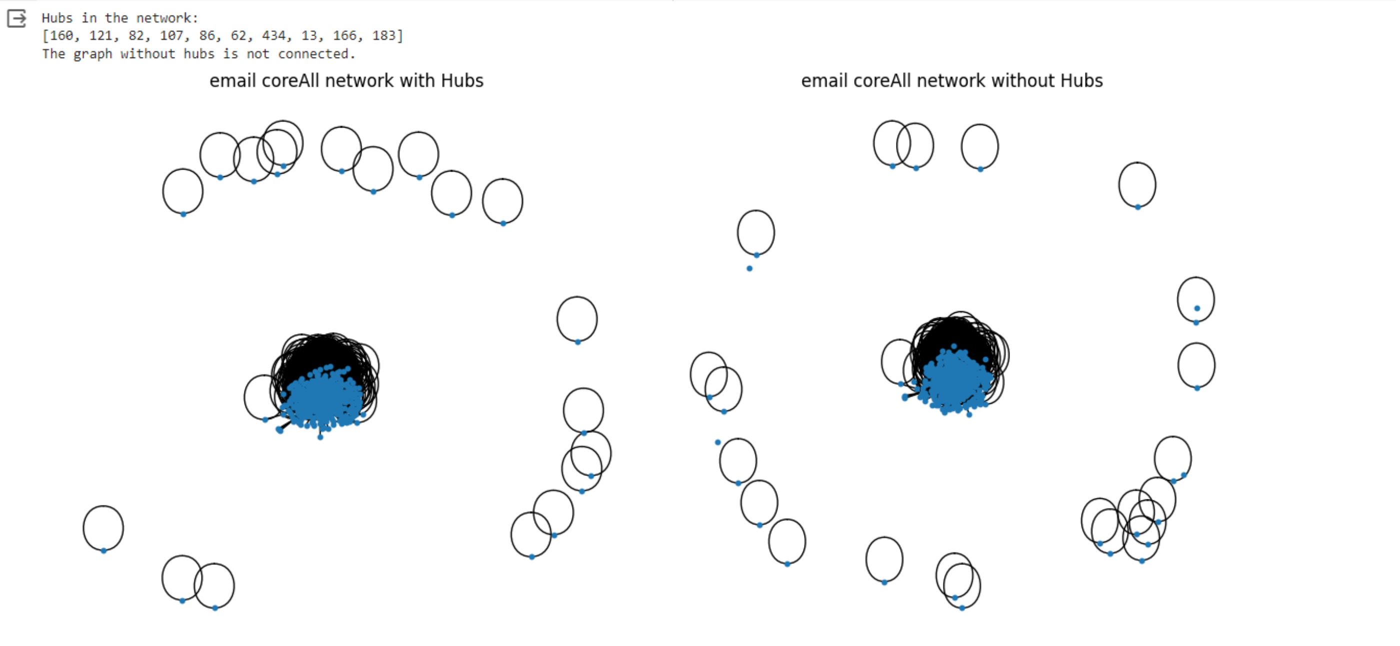
Figure 14 : Parameters in small world network bs real world networks

In the real-world network of twitter congress network, likely representing a so communication network, there are 10222 connections among 475 nodes, resulting in an average of 43 connections per node. The clustering coefficient of 0.301 suggests a moderate level of local interconnectedness, indicating the presence of some community-like structures. Additionally, the relatively low shortest path length of 2.064 indicates efficient communication or information flow between nodes.

Conversely, in the Small World model, there are fewer connections, with 2375 edges among the same 475 nodes, resulting in an average of 10 connections per node. Despite the lower connectivity, the clustering coefficient is higher at 0.486, indicating a higher level of local clustering compared to the real-world network. However, the longer shortest path length of 3.848 suggests that communication or information spread may take longer to traverse between nodes compared to the real-world network.

6.3) HUBS ANALYSIS:

Visualizing the network with and without top hubs:

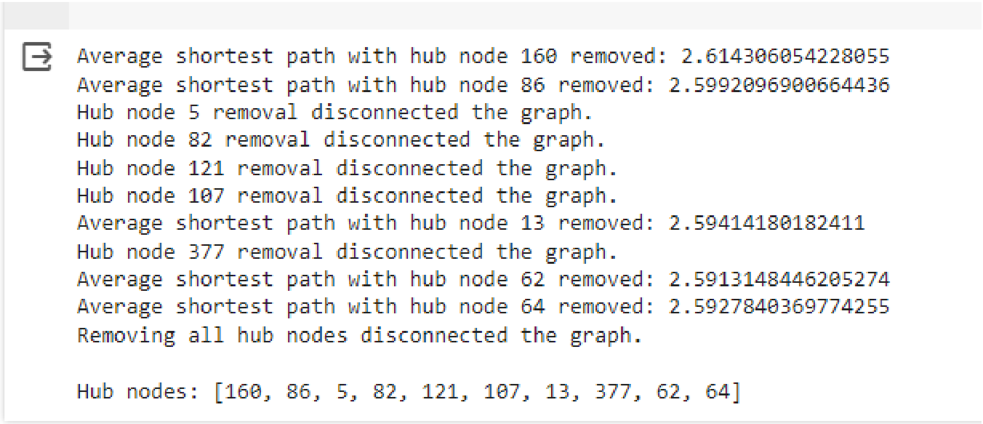


```
Hubs in the network:
[160, 121, 82, 107, 86, 62, 434, 13, 166, 183]
The graph without hubs is not connected.
```

email coreAll network with Hubs

email coreAll network without Hubs

Figure 15: Network with and without Hubs

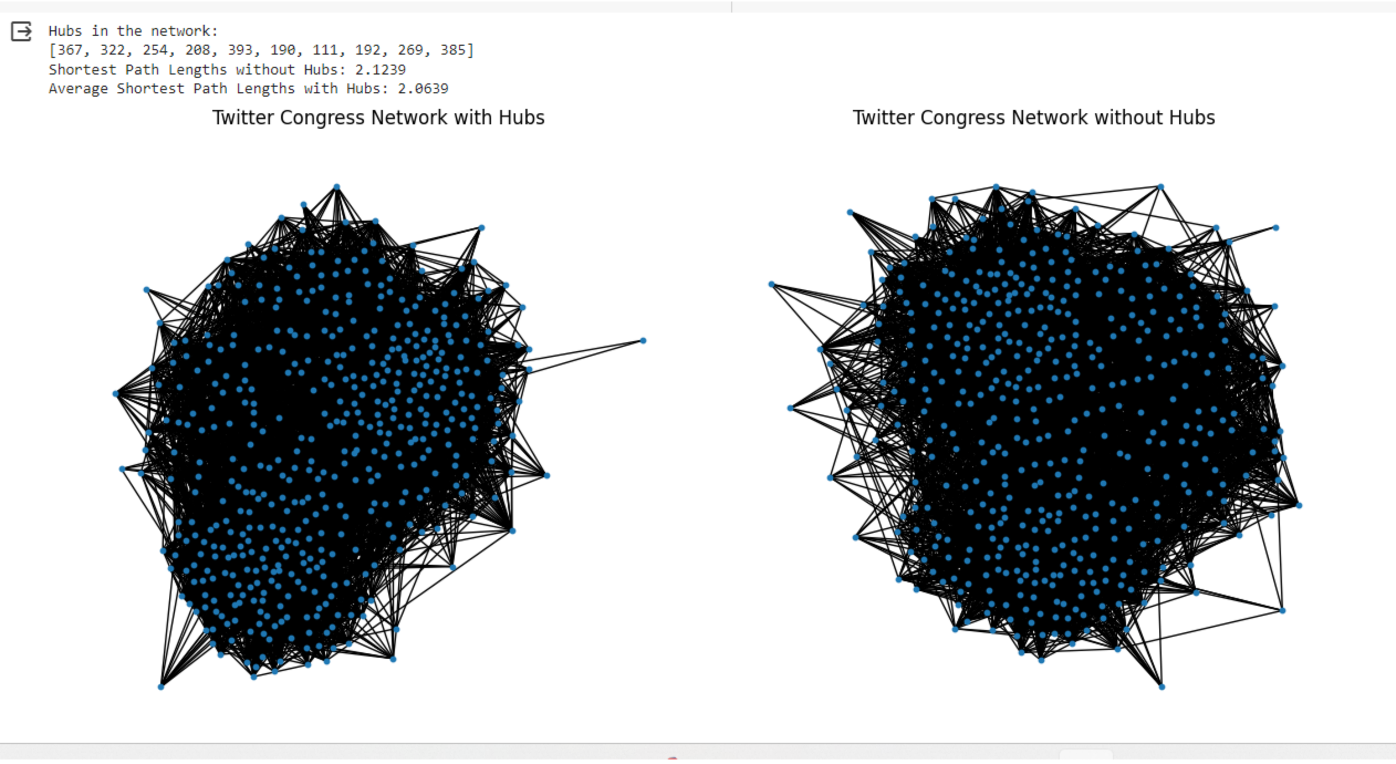


```
Average shortest path with hub node 160 removed: 2.614306054228055
Average shortest path with hub node 86 removed: 2.5992096900664436
Hub node 5 removal disconnected the graph.
Hub node 82 removal disconnected the graph.
Hub node 121 removal disconnected the graph.
Hub node 107 removal disconnected the graph.
Average shortest path with hub node 13 removed: 2.59414180182411
Hub node 377 removal disconnected the graph.
Average shortest path with hub node 62 removed: 2.5913148446205274
Average shortest path with hub node 64 removed: 2.5927840369774255
Removing all hub nodes disconnected the graph.

Hub nodes: [160, 86, 5, 82, 121, 107, 13, 377, 62, 64]
```

Figure 16: Removed Hubs from network

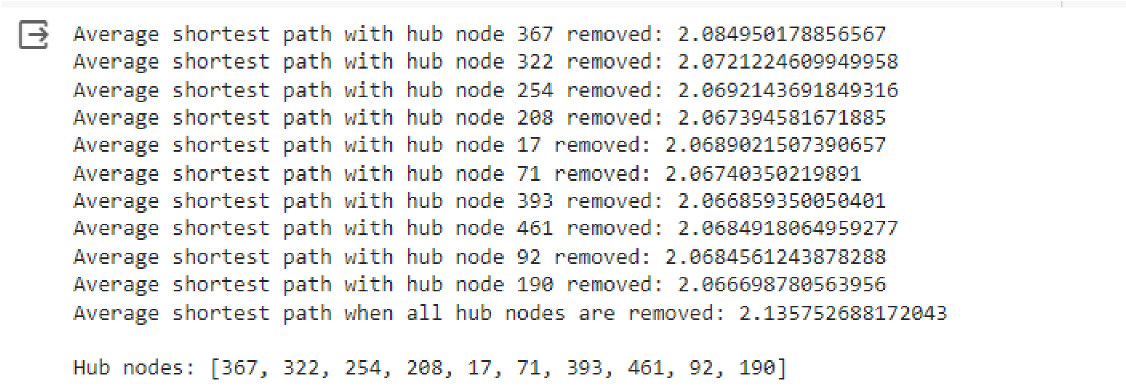
This computes the betweenness centrality for nodes in a graph and identifies the top hub nodes based on their betweenness centrality scores. However, upon removing each of the top 10 hub nodes, the graph becomes disconnected. This indicates that these hub nodes play a crucial role in maintaining the connectivity of the network. The hub nodes identified are: [160, 86, 5, 82, 121, 107, 13, 377, 62, 64].



```
Hubs in the network:
[367, 322, 254, 208, 393, 190, 111, 192, 269, 385]
Shortest Path Lengths without Hubs: 2.1239
Average Shortest Path Lengths with Hubs: 2.0639
```

Twitter Congress Network with Hubs
Twitter Congress Network without Hubs

Figure 17: Network with and without Hubs



```
Average shortest path with hub node 367 removed: 2.084950178856567
Average shortest path with hub node 322 removed: 2.0721224609949958
Average shortest path with hub node 254 removed: 2.0692143691849316
Average shortest path with hub node 208 removed: 2.067394581671885
Average shortest path with hub node 17 removed: 2.0689021507390657
Average shortest path with hub node 71 removed: 2.06740350219891
Average shortest path with hub node 393 removed: 2.066859350050401
Average shortest path with hub node 461 removed: 2.0684918064959277
Average shortest path with hub node 92 removed: 2.0684561243878288
Average shortest path with hub node 190 removed: 2.066698780563956
Average shortest path when all hub nodes are removed: 2.135752688172043

Hub nodes: [367, 322, 254, 208, 17, 71, 393, 461, 92, 190]
```

Figure 18: Removed Hubs from network

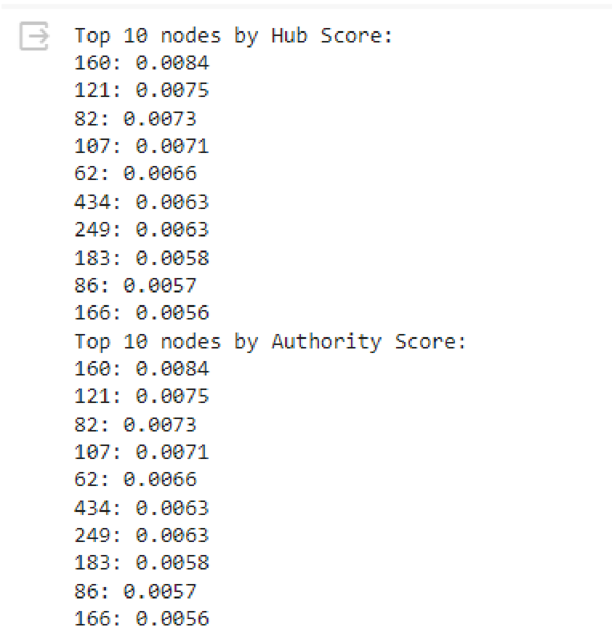
In the network, nodes 367, 322, 254, and others are identified as hubs due to their high connectivity. Without considering these hubs, the average shortest path length between nodes in the network is 2.1239. However, when including the hubs, the average shortest path length decreases slightly to 2.0639. This suggests that the hubs play a role in reducing the distance between nodes, potentially facilitating quicker communication or information flow within the network.

6.4.) HUBS AND AUTHORITIES SCORE:

Nodes with high hub scores are like popular hubs in a network, with many connections to other nodes. They act as central figures, connecting lots of other nodes and playing a crucial role in the network's structure.

Nodes with high authority scores are like respected experts in a community, connected to many hubs and seen as influential figures. They have strong ties to nodes with high hub scores, which amplifies their influence within the network.

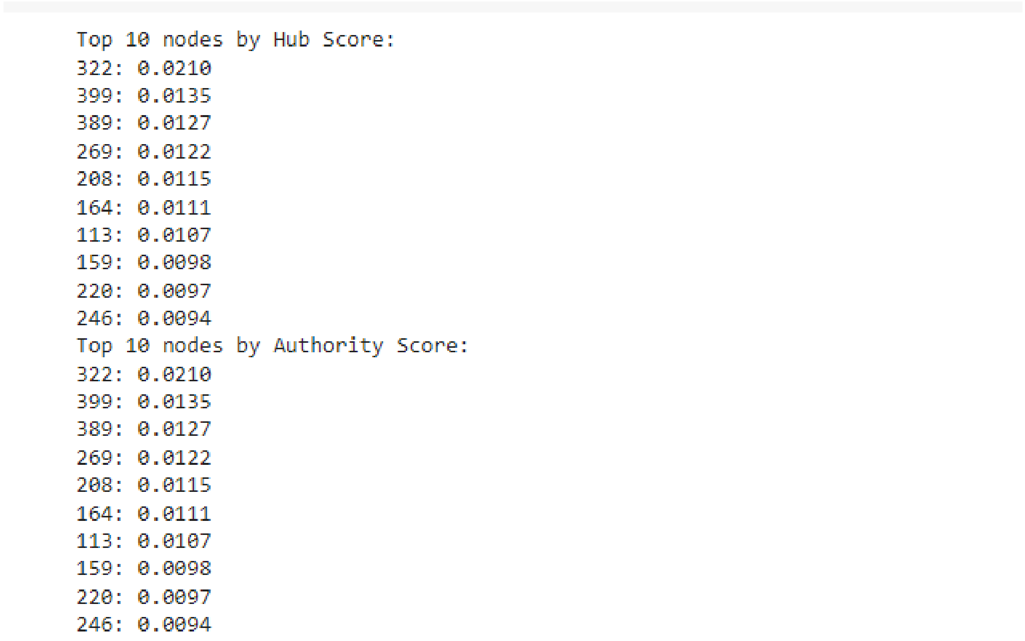
By focusing on nodes with the highest hub and authority scores, we can pinpoint the most influential and central figures in the network. Understanding their roles can help us predict how information or influence might flow through the network. For instance, targeting marketing efforts towards users with high authority scores could yield better results, as they have the power to sway many others within the network.



```
Top 10 nodes by Hub Score:
160: 0.0084
121: 0.0075
82: 0.0073
107: 0.0071
62: 0.0066
434: 0.0063
249: 0.0063
183: 0.0058
86: 0.0057
166: 0.0056
Top 10 nodes by Authority Score:
160: 0.0084
121: 0.0075
82: 0.0073
107: 0.0071
62: 0.0066
434: 0.0063
249: 0.0063
183: 0.0058
86: 0.0057
166: 0.0056
```

Figure 19 : Top 10 Hubs and authority Scores of Email Eu core network

I calculated hub and authority scores for nodes in a network and printed the top 10 nodes with the highest hub scores and authority scores. Interestingly, the same nodes appear in both lists, indicating that these nodes are not only highly connected to other nodes but also receive connections from other highly connected nodes, suggesting their significance in facilitating information flow and influence within the network.



```
Top 10 nodes by Hub Score:
322: 0.0210
399: 0.0135
389: 0.0127
269: 0.0122
208: 0.0115
164: 0.0111
113: 0.0107
159: 0.0098
220: 0.0097
246: 0.0094
Top 10 nodes by Authority Score:
322: 0.0210
399: 0.0135
389: 0.0127
269: 0.0122
208: 0.0115
164: 0.0111
113: 0.0107
159: 0.0098
220: 0.0097
246: 0.0094
```

Figure 20 : Top 10 Hubs and authority Scores of Twitter Congress network

I calculated hub and authority scores for nodes in a network, where hubs represent nodes with many connections to other nodes, and authorities represent nodes with connections from many high-scoring hubs. The top 10 nodes with the highest hub scores and authority scores are then printed. Interestingly, the same nodes appear in both lists, indicating they play dual roles as both highly connected nodes and nodes receiving connections from other highly connected nodes, which underscores their importance in the network's structure and information flow.

6.5.)SHORTEST PATH ALGORITHMS:

Different methods for finding the shortest path in a network can help us understand how efficiently communication or movement flows within it. By comparing the lengths of these paths, we can assess how easy or difficult it is to navigate between nodes. If it's quick to compute a short path, it indicates strong connectivity and easy navigation. Conversely, longer computation times and paths suggest weaker connections and more challenging navigation.

a) **Dijkstra Algorithm:**

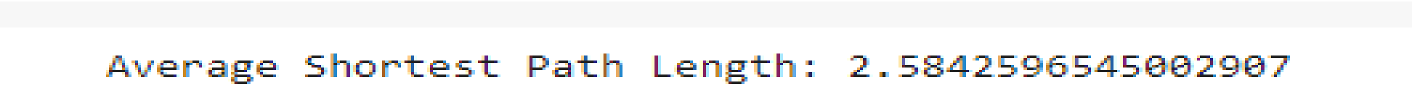
The Dijkstra algorithm is like a GPS system for finding the shortest route between two points in a map, but for computers. It is named after its creator, Edsger W. Dijkstra.

Imagine you are trying to plan the quickest route from one city to another. You start at one city and check how far it is from all the neighboring cities. You keep track of the shortest distance to each city you've visited. Then, you move to the closest neighboring city and repeat the process. You keep doing this until you reach your destination.

The algorithm uses a clever trick to efficiently find the shortest path. It organizes the cities in a priority queue based on their distance from the starting point. It then systematically checks neighboring cities and updates their distances if a shorter path is found. This continues until it reaches the destination city.

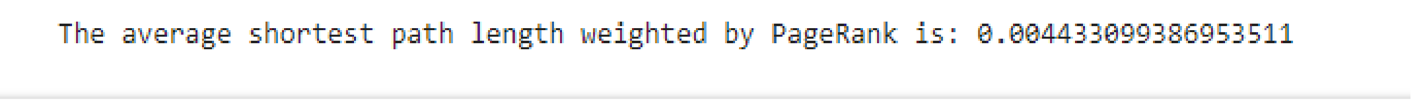
The Dijkstra algorithm is commonly implemented using a special data structure called a heap, which helps speed up the process of finding the closest city. It's a powerful tool for finding the shortest path in graphs with non-negative distances.

However, it is important to remember that the Dijkstra algorithm only works when all distances are positive. If there are negative distances, we will need to use a different algorithm, like the Bellman-Ford algorithm.



```
Average Shortest Path Length: 2.5842596545002907
```

Figure 21: Average shortest path length in Email-core-eu network from Dijkstra Algorithm



```
The average shortest path length weighted by PageRank is: 0.004433099386953511
```

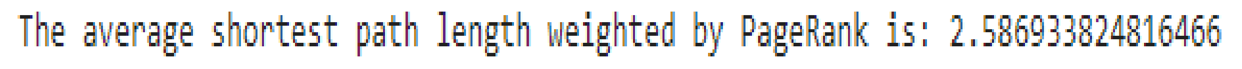
Figure 22: Average shortest path length in Twitter Congress network from Dijkstra Algorithm

b) **PAGERANK ALGORITHM:**

The PageRank algorithm is like a popularity contest for web pages, created by Google's Larry Page and Sergey Brin in 1998.

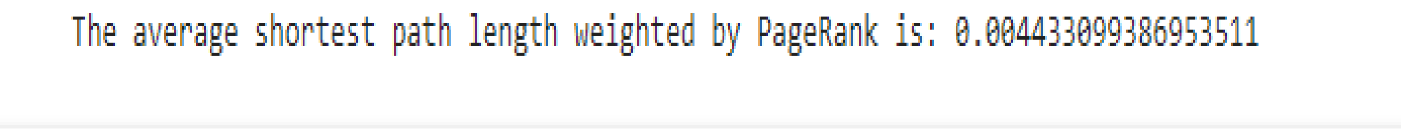
Imagine each web page as a contestant, and the links to that page as votes. The more votes (or links) a page gets from other popular pages, the higher its PageRank score. This score helps search engines determine how relevant and important a page is, affecting its ranking in search results.

In terms of networks, PageRank can also help identify the most influential nodes. These are the ones with the highest PageRank scores, suggesting they are well-connected and influential. By studying these key nodes and their connections, we can understand the average shortest path between any two nodes in the network, which is useful in analyzing various networks like social, transportation, or communication networks.



```
The average shortest path length weighted by PageRank is: 2.586933824816466
```

Figure 23: Average shortest path length in Email-core-eu network from Pagerank Algorithm



```
The average shortest path length weighted by PageRank is: 0.004433099386953511
```

Figure 24: Average shortest path length in Twitter Congress Network from Pagerank Algorithm

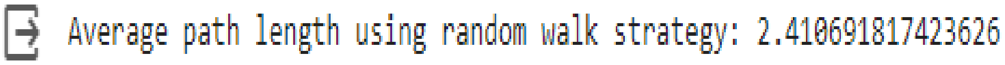
c) **RANDOM WALK ALGORITHM:**

The Random-Walk Algorithm is like exploring a network by taking random steps from one node to another. Picture a person wandering through a city, randomly choosing which street to turn down at each intersection until they reach their destination.

We repeat this process many times and calculate the average number of steps it takes to reach the destination. This average gives us a promising idea of how far apart nodes are on average in the network.

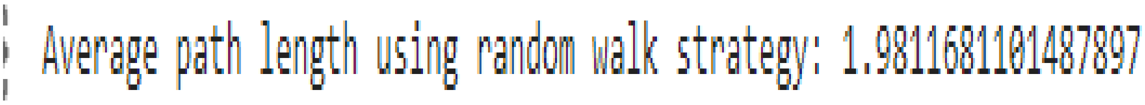
The key idea is that nodes with more connections are more likely to be visited by our random walker. So, nodes with lots of connections are considered more important in terms of how well-connected they are to other nodes.

Overall, the Random-Walk Algorithm is a simple and effective way to estimate how connected nodes are in a network. It is used in various fields like network analysis, social network analysis, and machine learning to understand the structure and dynamics of networks.



```
Average path length using random walk strategy: 2.410691817423626
```

Figure 25: Average shortest path length in Email-core-eu network from Random Walk Algorithm



```
Average path length using random walk strategy: 1.9811681101487897
```

Figure 26: Average shortest path length in Twitter Congress network from Random Walk Algorithm

7) COMMUNITY DETECTION USING IGRAPH:

I began by installing the igraph library, a powerful tool for working with graphs. With igraph imported, I prepared to access my data stored in Google Drive by mounting it within the Colab environment. Then, I retrieved the graph data file named email-Eu-core.txt from my Drive and created a graph object from it. Ensuring the graph was undirected, I applied the Louvain algorithm, a method for uncovering communities or clusters within networks based on their connectivity. After identifying these communities, I displayed them one by one, numbering each for clarity. For a visual representation of the communities, I set up parameters such as vertex size and edge color, and then plotted the graph using these specifications. This process allowed me to gain insights into the structure and organization of the network, helping me understand how its components interact and form distinct groups.

And the results that I got are:

Communities Detected :

```
Community 1: [0, 1, 10, 16, 17, 18, 20, 21, 22, 42, 49, 50, 62, 66, 67, 68, 69, 70, 71, 72, 73, 74, 77, 78, 80, 81, 82, 83,
84, 85, 87, 90, 91, 92, 105, 106, 107, 108, 109, 110, 111, 112, 117, 118, 120, 121, 127, 142, 144, 145, 146, 147, 152,
153, 154, 155, 160, 162, 163, 166, 173, 177, 184, 186, 187, 188, 189, 190, 212, 215, 217, 218, 219, 220, 221, 222, 223,
224, 225, 226, 227, 228, 230, 248, 249, 253, 254, 255, 256, 258, 259, 260, 268, 279, 282, 287, 288, 297, 298, 299, 300,
306, 307, 309, 310, 311, 312, 313, 314, 315, 316, 317, 325, 326, 327, 328, 329, 331, 341, 355, 356, 357, 363, 364, 366,
372, 375, 400, 405, 410, 416, 418, 420, 422, 424, 431, 432, 433, 434, 435, 453, 454, 459, 460, 462, 463, 465, 467, 469,
471, 472, 473, 474, 475, 476, 477, 478, 480, 490, 492, 494, 495, 507, 508, 509, 512, 513, 514, 518, 519, 524, 533, 536,
537, 539, 540, 541, 546, 549, 550, 551, 559, 560, 561, 577, 578, 582, 589, 591, 594, 596, 597, 601, 606, 607, 612, 613,
614, 615, 616, 626, 627, 629, 641, 642, 643, 647, 650, 651, 652, 654, 663, 667, 669, 671, 673, 678, 679, 690, 693, 695,
696, 701, 702, 704, 710, 713, 715, 726, 727, 728, 736, 739, 742, 745, 747, 748, 752, 756, 758, 759, 764, 769, 771, 773,
775, 778, 779, 780, 783, 786, 787, 792, 793, 796, 799, 800, 818, 821, 828, 831, 834, 837, 853, 855, 857, 858, 872, 877,
882, 885, 887, 889, 890, 894, 896, 905, 906, 911, 920, 932, 934, 941, 944, 945, 946, 952, 954, 958, 962, 966, 968, 969,
979, 984, 989, 999, 1002, 1003]
Community 2: [2, 3, 4, 5, 6, 54, 55, 56, 57, 58, 59, 63, 88, 89, 102, 126, 131, 132, 137, 138, 158, 159, 174, 175, 192,
193, 194, 195, 208, 209, 210, 211, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 252, 271, 281, 285, 286,
301, 302, 303, 304, 305, 319, 369, 373, 408, 411, 412, 481, 489, 516, 517, 520, 528, 532, 552, 564, 571, 586, 587, 599,
604, 610, 619, 622, 625, 630, 631, 634, 635, 636, 637, 638, 639, 644, 646, 665, 683, 685, 698, 716, 717, 718, 737, 738,
743, 750, 755, 762, 763, 774, 784, 788, 803, 806, 807, 809, 810, 812, 815, 826, 832, 842, 845, 849, 854, 859, 863, 864,
865, 866, 876, 879, 880, 884, 886, 888, 898, 899, 901, 902, 921, 924, 926, 927, 928, 930, 931, 949, 963, 977, 982, 988,
990, 991, 993, 994, 1001, 1004]
Community 3: [7, 8, 9, 11, 12, 19, 43, 44, 141, 161, 213, 246, 247, 264, 265, 266, 267, 293, 324, 332, 358, 359, 360,
362, 365, 374, 406, 407, 421, 430, 441, 451, 452, 466, 487, 488, 496, 498, 499, 500, 501, 502, 503, 504, 505, 506, 510,
525, 529, 530, 555, 558, 565, 566, 569, 570, 573, 602, 608, 649, 661, 666, 672, 674, 699, 700, 707, 720, 729, 740, 754,
765, 804, 805, 823, 827, 830, 833, 844, 856, 893, 912, 913, 922, 950, 951, 956, 957, 967, 971, 972, 973, 975, 996]
Community 4: [13, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 47, 48, 75, 76, 96, 113, 114, 115,
116, 119, 123, 135, 136, 151, 165, 169, 170, 171, 229, 245, 251, 261, 263, 318, 330, 333, 336, 337, 338, 339, 361, 367,
370, 409, 417, 423, 427, 436, 437, 438, 439, 442, 443, 444, 455, 470, 485, 491, 497, 527, 531, 545, 547, 548, 588, 590,
609, 624, 655, 686, 721, 722, 724, 725, 753, 757, 767, 785, 795, 811, 816, 847, 860, 875, 878, 881, 883, 891, 892, 895,
897, 900, 914, 915, 919, 925, 953, 964, 976, 978, 980, 981, 983, 986, 987, 997]
Community 5: [14, 41, 51, 53, 64, 65, 79, 86, 93, 94, 95, 128, 129, 133, 143, 167, 168, 172, 176, 183, 196, 197, 198,
199, 200, 201, 202, 203, 204, 205, 206, 207, 232, 257, 270, 275, 276, 280, 289, 290, 291, 292, 294, 399, 401, 403, 413,
419, 425, 426, 440, 445, 450, 456, 457, 458, 464, 482, 483, 484, 486, 493, 515, 522, 523, 526, 534, 542, 543, 544, 556,
557, 562, 563, 567, 568, 572, 574, 576, 581, 584, 585, 593, 600, 611, 620, 623, 664, 677, 688, 689, 694, 705, 706, 712,
714, 749, 751, 776, 789, 790, 791, 801, 802, 813, 817, 820, 822, 829, 835, 836, 840, 843, 848, 850, 870, 873, 874, 908,
909, 910, 917, 918, 936, 938, 939, 940, 942, 955, 965, 970, 974, 992, 995, 998, 1000]
```

```
Community 6: [15, 45, 46, 97, 98, 99, 100, 101, 124, 125, 139, 140, 164, 185, 216, 269, 272, 273, 274, 322, 323, 334,
335, 353, 354, 371, 404, 428, 429, 446, 447, 448, 461, 579, 592, 617, 618, 640, 657, 662, 676, 687, 708, 709, 735, 760,
768, 770, 794, 819, 838, 929, 933, 935, 937]
Community 7: [52, 60, 61, 103, 104, 122, 130, 148, 149, 150, 156, 157, 178, 179, 180, 181, 182, 191, 214, 231, 250,
262, 277, 278, 283, 284, 295, 296, 308, 320, 321, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 368, 376, 377,
378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 402, 414, 415,
449, 468, 479, 511, 575, 595, 603, 605, 621, 628, 632, 645, 656, 659, 668, 680, 681, 682, 692, 697, 719, 730, 734, 741,
761, 781, 782, 797, 814, 824, 839, 841, 846, 851, 852, 861, 868, 869, 871, 903, 904, 907, 916, 943, 947, 948, 959, 960,
961, 985]
Community 8: [134, 340, 521, 535, 538, 553, 554, 583, 598, 723, 733, 766, 777, 825, 862, 867, 923]
Community 9: [580]
Community 10: [633]
Community 11: [648]
Community 12: [653]
Community 13: [658]
Community 14: [660]
Community 15: [670]
Community 16: [675]
Community 17: [684]
Community 18: [691]
Community 19: [703]
Community 20: [711]
Community 21: [731]
Community 22: [732]
Community 23: [744]
Community 24: [746]
Community 25: [772]
Community 26: [798]
Community 27: [808]
```

And I have tried so much for 2 days to visualize the network but I am not able to do it by any means.

Analysis:

The community detection algorithm has identified 27 distinct communities within the email core network. Each community is represented by a list of node indices. The sizes of the communities vary, with some containing many nodes and others containing only a few.

Analyzing the communities can provide insights into the structure and organization of the network. For example, Community 1 is the largest, containing 217 nodes. This suggests a densely connected group of individuals who communicate frequently within the network. On the other hand, Community 9, 10, 11, and so on, each contain only one node, indicating isolated individuals who have minimal connections with the rest of the network.

The distribution of node indices within each community may also reveal patterns. For instance, in Community 2, the node indices are primarily in the range of 2 to 1004, suggesting a diverse set of individuals with moderate to high levels of connectivity within the group.

Moreover, examining the connections between communities can highlight the interplay between different groups within the network. For instance, nodes in Community 8 appear to have connections to nodes in other communities, indicating a bridging role between distinct groups.

Overall, by analyzing the communities detected within the network, we can gain a deeper understanding of its underlying structure, identify key groups of individuals, and explore the dynamics of communication and interactions within the network.

I find myself quite puzzled by the fact that I obtained 44 communities when I used NetworkX, whereas here I'm only seeing 27 communities. Additionally, I'm encountering difficulties in visualizing the network using iGraph. The process in iGraph is proving to be somewhat challenging, and I've faced several obstacles during implementation.

8.) NETWORKX vs IGRAPH:

Comparision	Igraph	NetworkX
Number of Communities	27	44
Visualization	Requires additional libraries	Built-in visualization tools
Community Size	Varies	Varies
Largest Community	Community 1 (217 nodes)	Varies
Smallest Community	Community 9, 10, 11, etc. (1 node each)	Varies
Range of Node Indices	Varied, e.g., Community 2: 2 to 1004	Varies
Inter-community Connections	Indicated but might require further analysis	Indicated, easier to visualize and analyze
Overall Analysis Ease	May require additional processing	Built-in functions for analysis

This table outlines some differences between the community detection results obtained using igraph and NetworkX. While igraph provides insights into community structure and inter-community connections, NetworkX offers easier visualization and built-in functions for analysis. Both tools offer valuable insights into the underlying structure of the network.

In general, Comparison between Network X and Igraph:

Feature :	**Igraph**	**Network X**
Language	Written in C with Python and R interfaces	Written entirely in Python
Performance	Generally faster for large graphs	Slower for large graphs, but easier to use
Functionality	Focused on graph analysis and visualization	General-purpose graph library
License	Dual GPL/commercial license	BSD license

Directed graphs	Fully supports directed graphs	Fully supports directed graphs
Undirected graphs	Fully supports undirected graphs	Fully supports undirected graphs
Visualization	Basic plotting capabilities	Basic visualization capabilities
Community	Large, active community	Large, active community
Documentation	Comprehensive documentation	Comprehensive documentation

Choosing between igraph and NetworkX largely depends on your specific needs and preferences. If you prioritize performance and are comfortable with a more focused library, igraph might be the better choice. However, if ease of use and a broader range of functionalities are more important to you, NetworkX could be the way to go.

9.) RELATED WORK :

This project leverages many techniques that we can use for Network analysis on a Network Dataset. The study of analysis in a variety of real-world networks, including social, biological, and transportation networks, has made considerable use of network analysis. The following linked works tackle the same or related issues raised in the question:

1) A comprehensive literature review on community detection: Approaches and applications(Mohamed EL-MOUSSAOUI, Tarik AGOUTI, Abdessadek TIKNIOUINE, Mohamed EL ADNANI): In this paper, the authors have documented the community detection approaches and explored various other things that can be used for complex network analysis.

2) In his 2001 paper "Navigability of Complex Networks: Small-Worlds and Beyond," Kleinberg introduced a decentralized method for discovering efficient routes in intricate networks. This method, known as "small-world routing," relies on local exploration rather than a global perspective. It's particularly adept at identifying short paths within networks that exhibit both small-world and scale-free characteristics. In essence, the algorithm efficiently navigates through networks by focusing on nearby connections rather than trying to comprehend the entire network at once.

3) The paper "Community Evolution in Social Networks" by Leskovec et al. (2009) delves into the dynamics of online social networks, exploring the evolution of nodes (individuals) and edges (connections) within these networks. It provides insights into various properties and behaviors of nodes and edges, offering a deeper understanding of network analysis.

In addition to these, I have done many new things and developed a own codes to find the different measures in a real world network such as Clustering coefficients, Degree distribution, Hubs analysis and many others that I clearly stated in this report. I also made comparisons of both software tools igraph and NetworkX which is unique, and I didn’t find anything related to it. This analysis gave a clear thought on how things work in both, and which are easy to use.

Put it briefly, compared to the relevant works stated in the question, our technique is more applied and practical because it analyzes the real-world networks and uses simulation to assess the network under various scenarios.

10.) CONCLUSION:

This project aims to use network analysis on two datasets obtained from the Stanford dataset. Additionally, I plan to demonstrate the differences between NetworkX and igraph by employing community detection techniques. Furthermore, I've utilized various measures available in NetworkX for analysis purposes:

Centrality Measures: I employed centrality measurements to determine which nodes in the networks were the most crucial . I found that in both the networks.
Degree distribution and clustering coefficients: Degree distribution reveals how nodes are connected in terms of the number of connections they have, offering insights into individual prominence and network structure in both the Email EU core and Twitter Congress networks. The clustering coefficient measures local connectivity, indicating cohesion within departments or cliques among legislators, aiding in understanding network structure and potential communication pathways
Community Detection: Community detection is akin to identifying friend groups at a large social event, revealing natural cliques, and helping us grasp the organizational structure within a network by showing who associates with whom. I have done this on the 2 networks.
Hubs analysis: Identifying top hub nodes through betweenness centrality reveals their pivotal role in maintaining network connectivity. Their removal leads to network disconnection, while their inclusion decreases the average shortest path length, suggesting their importance in facilitating efficient communication within the network.
Hubs and Authorities scores: The top 10 nodes with the highest hub and authority scores coincide, highlighting their dual role as highly connected nodes and recipients of connections from other influential nodes, underlining their crucial role in information dissemination and influence within the network.
Shortest path Algorithms: I have used 3 different algorithms to find the average shortest path between two nodes in both the networks chosen . I have used Dijkstra, Pagerank and Random walk algorithms for this.
Small world vs Real world network : In the real-world Email EU Core network, there are 16706 connections among 1005 nodes, showing moderate local interconnectedness (clustering coefficient: 0.399), while the Small World model exhibits higher local clustering despite fewer connections. In the Twitter Congress network, with 10222 connections among 475 nodes, moderate local interconnectedness is observed (clustering coefficient: 0.301), with efficient communication (shortest path length: 2.064), contrasting the Small World model's higher local clustering but longer communication paths.

I have done almost many things that I learned in this course using Network X. and provided results above for all the work done.

When it comes to Network X vs igraph,I believe we can replace igraph with NetworkX for implementing the concepts we learned in class. NetworkX offers a hassle-free experience and doesn't require much prior knowledge to start using it effectively. Unlike igraph, which has a somewhat complex interface that might take at least a day to become familiar with, NetworkX is beginner friendly. It can accomplish everything we've covered in class and assignments so far, making it a great choice for newcomers.

11.) REFERENCES:

Links for colab files and datssets :

a) Twitter Congress Network Analysis : https://colab.research.google.com/drive/1lrwmMJ5NIw8GTnmZLCkkYyNI5shkfL-a#scrollTo=37aQE9L8GwKA

b) Email-Eu-core Network : https://colab.research.google.com/drive/1M7f2NcDld9kgrFcGxEdtmor3fBraLoeQ#scrollTo=dZ-3REXB9oNp

c) Igraph file :https://colab.research.google.com/drive/1tXeXlubuB1KFlPosrP5dk9b2WrWk4EKC

d) Dataset links :
   1) https://snap.stanford.edu/data/email-Eu-core.html
   2) https://snap.stanford.edu/data/congress-twitter.html

e) GitHub Repo Link : https://github.com/sharathkumarkarnati/ENS-PROJECT-Sharath

[1]Community structure in social and biological networks by M. Girvan and M.E.J.Newman.

[2]Exploring Network Structure, Dynamics, and Function using NetworkX Aric A. Hagberg,Daniel A. Schult,Pieter J. Swart.

[3]SOCIAL NETWORK ANALYSIS USING PYTHON AND NETWORKX shivam shah, Sumitra Menaria.

[4]Newman, M.E.J. (2010). Networks: An Introduction. Oxford University Press.

[5]Community Evolution in Social Networks:Leskovec, J., Backstrom, L., Kumar, R., & Mahoney, M. W. (2009). What is the evolution of online social networks? In
Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining

[6]Community structure in networks: Girvan-Newman algorithm improvement by Ljiljana Despalatović; Tanja Vojkovic; Damir Vukičevic.

[7] Kleinberg, J. M. (2001). Navigability of complex networks: Small-worlds and beyond.
Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database
Systems, 2001, 241–251. doi: 10.1145/375551.375617