ANALYSIS OF CO-AUTHORISHIP NETWORK IN GOOGLE SCHOLAR

Project Review

CSE3021-SOCIAL AND INFORMATION NETWORK

FINAL PROJECT REVIEW

ANALYSIS OF CO-AUTHORISHIP NETWORK IN GOOGLE SCHOLAR

Submitted by:

Aviral Sharma (20BCE2918)

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science Engineering

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Under the guidance of:

Prof. Rajkumar R

SCOPE

ABSTRACT

A large number of researchers perform scientific studies in a group, thus forming a network of co-authors. Such a Co-authorship Network can tell a lot about the influence of a researcher in their field of study. Analyzing these networks provide an insight to the workings of complex evolving networks. In this project, an attempt has been made to study the co-authorship network extracted from Google Scholar and filtered based on PageRank by analyzing various centrality measures such as degree centrality, closeness centrality and betweenness centrality. This is achieved through web scrapping using Python and its various modules.

1. INTRODUCTION

Scientific collaboration is one of the main reasons for the advancement of human beings in various frontiers of knowledge. When researchers work together, they create a network among themselves called co-authorship network. Thus, a co-authorship network is a network with researchers as nodes and the links tying them together is the co-authorship. Two researchers are connected if they have worked on same research paper. It is one of the most used methods to study scientific collaboration since the publication of scientific paper is very well documented through the history. Another method to study scientific collaboration is citation network where the nodes are the scientific papers and they are linked if one paper has cited the other one.

It is important to study these co-authorship networks for multiple reasons. It is extremely useful to determine the influence of a particular individual on a field of study. Many scenarios can arise; an organization might want to know which research candidate performs better in a particular field, a staff enrollment panel may want to rank the competitors based on their past records of teamwork and collaboration, a student might want to choose the best advisor and so on. These cases require proper metrics, other than citation metrics like h-index, g-index, number of citations, which can be evaluated using co-authorship network analysis.

To analyze co-authorship network in this project, Google Scholar is used in this project. Google Scholar is a web search engine that indexes the full text or metadata of scholarly literature. Authors publish their academic papers along with their co-authors on Google Scholar, which allows the extraction of a co-authorship network. The details of an author is stored as their Google Scholar author profile. Using web scraping, the co-authors of a particular author can be extracted and a network can be built, which can then be used to perform various Social Network Analysis (SNA) techniques.

PROBLEM STATEMENT

Analyze co-authorship network to find out metrics that predicts an author’s influence and importance in the network.

2. LITERATURE REVIEW

a. Survey of the Existing Models/Works

[1] A.L Barabási, H Jeong, Z Néda, E Ravasz, A Schubert, T Vicsek, Evolution of the social network of scientific collaborations, in Physica A: Statistical Mechanics and its Applications, Volume 311, Issues 3–4, 2002, Pages 590-614, ISSN 0378-4371, https://doi.org/10.1016/S0378-4371(02)00736-7.

This paper studies the collaboration network among scientists who published journals in mathematics and neuro-science over a period of eight years from 1991 to 1998 using three major approaches. First, it uses direct measurements to discover the method by which collaboration network evolves over time. Second, a model based on the parameters found through direct measurements is constructed that provides the predictions required to explain many results. Finally, computer simulations are employed to further solidify the model and understand the gaps in theoretical model and experimental data. This study opens many paths to research upon other networks like the WWW and social networks in general.

[2] Yang Chen, Cong Ding, Jiyao Hu, Ruichuan Chen, Pan Hui, and Xiaoming Fu. 2017. Building and Analyzing a Global Co-Authorship Network Using Google Scholar Data. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1219–1224. DOI: https://doi.org/10.1145/3041021.3053056

In this paper, the authors have used Google Scholar as a platform to collect data of authors and built a global co-authorship network of about 402.39K authors. They applied various metrics of social network analysis such as degree, clustering coefficient and PageRank to study the co-authorship network. By building this global network, they were able to find many unique features of the network. They also found that there is a good correlation between position in the network and h-index.

[3] Mark EJ Newman, Coauthorship networks and patterns of scientific collaboration in Proceedings of the national academy of sciences 101 (suppl 1), 5200-5205 https://doi.org/10.1073/pnas.0307545100

Newman in his paper has collected data of authors from three areas- biology, physics and mathematics. He analysed the co-authorship network among the authors to answer a large number of questions including the number of papers an author writes, average number of co-authors, distance between collaborators and the temporal evolution of networks. He found out that biology contains the greatest number of authors and papers published, with higher number of collaborations as compared to physics and mathematics. He also evaluated the distance between authors, associativity, strength of collaboration, clustering coefficient, betweenness centrality and other metrics.

[4] Perianes-Rodríguez, A., Olmeda-Gómez, C. & Moya-Anegón, F. Detecting, identifying and visualizing research groups in co-authorship networks. Scientometrics 82, 307–319 (2010). https://doi.org/10.1007/s11192-009-0040-z

This paper suggests a method to detect, identify and visualize research groups in a co-authorship network. Research groups are scientists that collaborate on research and share resources and other materials, but not necessarily in the same organization. The groups are detected based on factorial analysis of raw data matrix and similarities in the choice of co-authors. These groups are then visualized using coloured diagrams. They referred to nine Carlos III University of Madrid developments for data and findings fir the Communication Technologies Department for the method.

[5] M. Fujita, H. Inoue and T. Terano, "Searching Promising Researchers through Network Centrality Measures of Co-author Networks of Technical Papers," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 2017, pp. 615-618. doi: 10.1109/COMPSAC.2017.205

The authors in this paper propose that betweenness centrality can be used as an effective metric to search for promising young researchers. They assumed that better researchers possess higher betweenness centrality and that the betweenness centrality of promising scientists will grow over time. They used the data from Japan Science and Technology Agency (JST) to detect growing betweenness. They also showed that the Research Fellows from Japan Society for the Promotion of Sciences (JSPS) who had been evaluated as promising researchers had a growing betweenness centrality.

b. Summary/Gaps/Limitations/Future Work identified in the Survey

The existing studies do not answer all the questions regarding the co-authorship network and there is a large amount of work required to understand and devise useful inferences from such networks.

1. The proper study of the communities in co-authorship networks, the flow of information among various actors, evolution of the network and other important factors are inadequate.
2. The papers are restricted to a specific field of study and are not common for all the researchers.
3. The use of the information obtained from the network analysis is still undetermined.
4. The studies are yet to be compared at scientific, regional, national and international levels of the network.

3. OVERVIEW OF THE PROPOSED SYSTEM

a. Introduction

To construct the co-authorship network, we require the name of an author. From the author's profile, the list of the co-authors has to be extracted which will be used to construct the network with the authors as nodes and the relationship between them as the edge. Then the coauthors of the coauthors are determined one by one and are added to the network. To achieve this, web scrapping is used. After the construction of the network, various analysis techniques are performed on it and results are plotted.

b. Modules of the Proposed System

To make the network and perform required analysis, the system is divided into four modules, providing reusability and coherence. The details of the modules are discussed here.

i. Web Scraping

Instead of choosing an available dataset, the required data is obtained through scraping the Google Scholar website to get continuous actors in the network. It also allows the control of the number of actors in the group and size of the group. This can be done by recognizing the correct HTML tag attribute that contains the label of the author.

ii. Creating Network and Subnetwork

After data collection through web scraping, the next step is to develop the network by finding the connections between the nodes using a network generator. Further, the data is pruned and the weak nodes are removed to generate a sub network.

iii. Network Analysis

The network analysis is performed by calculating various centrality measures that include degree centrality, betweenness centrality, closeness centrality and eigenvector centrality. PageRank is an example of eigenvector centrality. PageRank was one of the algorithms used by Google to rank its websites. PageRank calculations require a few passes called emphases. These measures are calculated for the network created for each of the node.

iv. Result Visualization

The final co-authorship network is visualized using this module so that it becomes easy for us to observe patterns in the network.

c. PROPOSED SYSTEM MODEL

The proposed framework mainly consists of the web scraping module. It takes the Google Scholar url of the author and the maximum number of nodes to be considered. It then finds all the coauthors of from that link and extracts the url and name for each of the coauthor and adds it to the network. When the maximum number of nodes is reached, the loop breaks and we get a final network of coauthors. Using that network, the centrality measures are calculated for each node and a subgraph of the most prominent actors is also generated by filtering them through page rank algorithm.

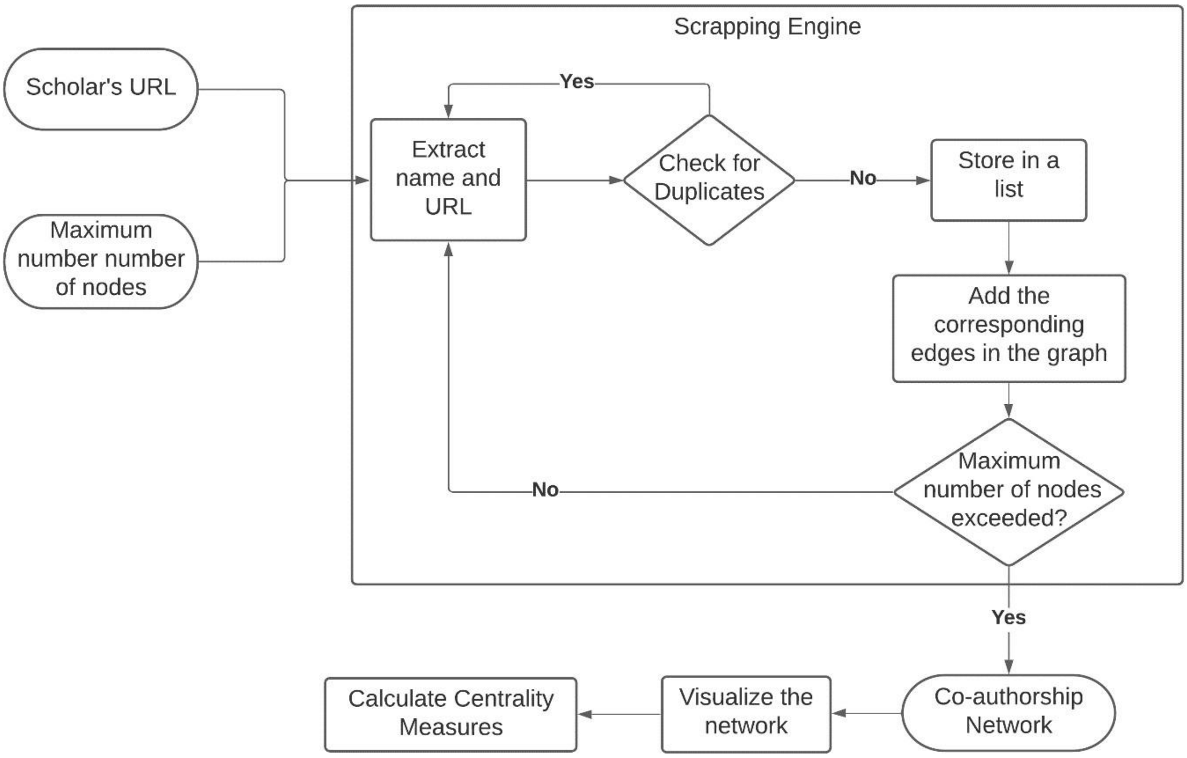


Fig. Scraping Engine and Network Analysis Architecture

The analysis of the network was carried out using centrality measures such as degree centrality, closeness centrality, betweenness centrality and page rank algorithm.

Degree Centrality

$$C_D(G) = \frac{\sum_{i=1}^{|V|}[C_D(v*) - C_D(v_i)]}{|V|^2 - 3|V| + 2}$$

Closeness Centrality

$$C(u) = \frac{1}{\sum_y d(u,y)}$$

Betweenness Centrality

$$B(u) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Page Rank

$$PR(u) = (1-d) + d \times \sum \frac{PR(v)}{N(v)}$$

4. PROPOSED SYSTEM ANALYSIS AND DESIGN

The system consists of a web scrapper, a network generator and network analyser.

Web Scrapper

The web scrapper is implemented using beautifulSoup. The response of the url of the author is fetched using requests module and passed to BeautifulSoup. The HTML parser parses the response of the webpage, which is then used to get the necessary information.

The name of the author is situated in a div tag with id “gsc_prf_in”.

```
<div id="gsc_prf_in">Matthijs J. Warrens</div>
```

Fig. div tag containing the name of the author

The list of the coauthors is stored in an anchor tag <a> with tabindex value of -1.

```
https://scholar.google.com/citations?user=tvmxzygAAAAJ&hl=en&oi=sra
Matthijs J. Warrens
******
14
['http://scholar.google.nl/citations?user=-U1TQ5gAAAAJ&hl=en', 'http://scholar.google.nl/citatior
http://scholar.google.nl/citations?user=-U1TQ5gAAAAJ&hl=en
Roel J. Bosker
******
http://scholar.google.nl/citations?user=x9ukpr0AAAAJ&hl=en
Davide Chicco
******
http://scholar.google.nl/citations?user=ovoyC-kAAAAJ&hl=en
Giuseppe Jurman
******
http://scholar.google.nl/citations?user=Y3ZmEWwAAAAJ&hl=en
Hanke Korpershoek
******
http://scholar.google.nl/citations?user=08KXrAIAAAAJ&hl=en
Willem J. Heiser
******
http://scholar.google.nl/citations?user=5eiUuO0AAAAJ&hl=en
```

```
http://scholar.google.nl/citations?user=5eiUuO0AAAAJ&hl=en
Jan H. van Driel
******
http://scholar.google.nl/citations?user=_d9scVEAAAAJ&hl=en
Rink Hoekstra
******
http://scholar.google.nl/citations?user=4RvhI-AAAAAJ&hl=en
Peter M Kruyen
******
http://scholar.google.nl/citations?user=l7s7CjcAAAAJ&hl=en
Lorenza Colzato
******
http://scholar.google.nl/citations?user=JR_msDAAAAAJ&hl=en
Marie Stadel
******
http://scholar.google.nl/citations?user=a4X7AFcAAAAJ&hl=en
Judith K. Daniels
******
http://scholar.google.nl/citations?user=Wxaac_gAAAAJ&hl=en
Bertus F. Jeronimus
```

```
******
http://scholar.google.nl/citations?user=a4X7AFcAAAAJ&hl=en
Judith K. Daniels
******
http://scholar.google.nl/citations?user=Wxaac_gAAAAJ&hl=en
Bertus F. Jeronimus
******
http://scholar.google.nl/citations?user=NRKUWiYAAAAJ&hl=en
Jeroen de Mast
******
136
more than 100 people now, break
```

Network Generator

The network is generated using the networkx library. The list of all the co-authors is generated using the web scraper and stored in a dictionary, which is used to create the network. A subnetwork is also generated with the nodes with page rank higher than the average value, which represents the graph of authors with higher impact.

Network Analysis

The analysis of network is done using centrality measures like degree centrality, closeness centrality, betweenness centrality and page rank algorithm. All the algorithms are available in the network library.

Degree centrality is the number of edges incident on a particular node. It is one of the mostbasic forms of centrality measures.

Closeness centrality of a node refers to the average length of the shortest path between the node and the remaining nodes in the graph. The node which is closer to the greatest number of nodes has higher value of closeness centrality. Closeness centrality is a useful measure that estimates how fast the flow of information would be through a given node to other nodes. Closeness centrality measures how short the shortest paths are from node i to all nodes.

Betweenness centrality of a node is the number of times a node passes through the shortest distance between any nodes in the graph. The nodes with higher betweenness centrality affect the flow of the information in the network.

PageRank algorithm developed by Google uses eigenvector centrality to find out the influence of a node in a graph. It was developed originally to rank the web pages.

The resulting graph is plotted using networkx library. The size of the nodes in the graph are drawn according to the degree of the node, which makes it easier to visualize the graph.

5. IMPLEMENTATION

The project was implemented on VS Code using Python. The details of themodules used are presented here.

- Programming Language: Python
- Platform: VS Code
- Modules used:
    - matplotlib
    - networkx
    - pandas
    - requests
    - BeautifulSoup
    - seaborn
    - defaultdict
    - urllib3

The main libraries used in the project are

NetworkX is a python library used to create, analyze and manipulate networks. It is used to make networks in various formats and visualize them. It also has built-in methods to perform various calculations on a network.

matplotlib is a python library to create various types of visualizations such as graphs and plots.

seaborn is another python library to visualize data using various plots. It is based on matplotlib and contains many additional attractive formats.

Beautiful Soup is used to parse the data out of HTML and XML files. It can be used to search the contents of tags in the files with ease.

The variable scholar is assigned the url of the author and the number of nodes to be considered was set to 100.

scholar = 'https://scholar.google.com/citations?user=x6fNSxcAAAAJ&hl=en'

nodes=100

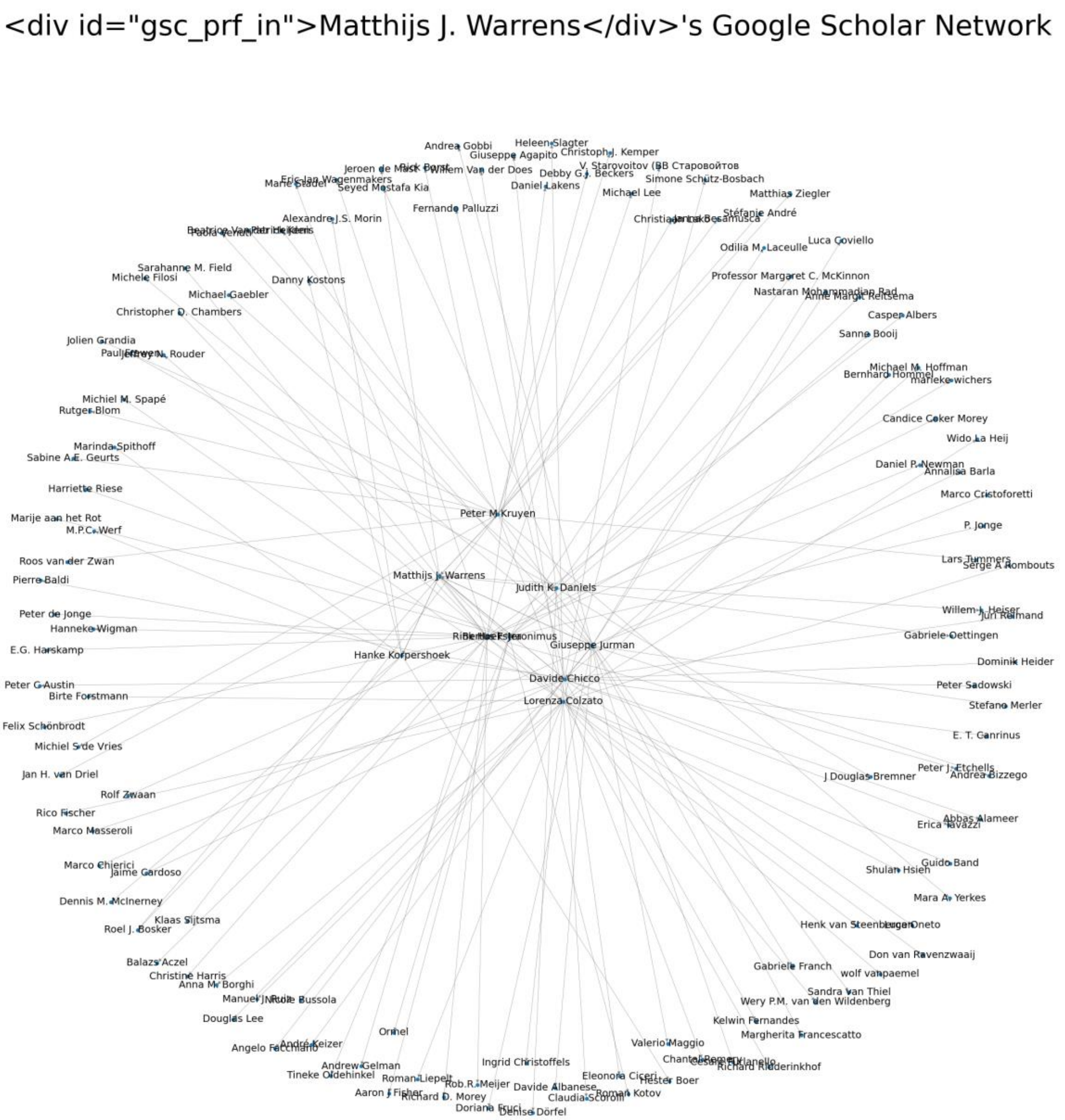
The author corresponding to the link is Matthijs j. Warrens

Google drive link:

[https://drive.google.com/drive/folders/1uf8uMTdFg
HVW9hu1wpYwrsjg6omGf37g?usp=share_link](https://drive.google.com/drive/folders/1uf8uMTdFgHVW9hu1wpYwrsjg6omGf37g?usp=share_link)

6. RESULTS AND DISCUSSION

Upon running the code, the network of 204 nodes was made with 100 nodes considered for the analysis. The network is visualized and presented in the figure below.

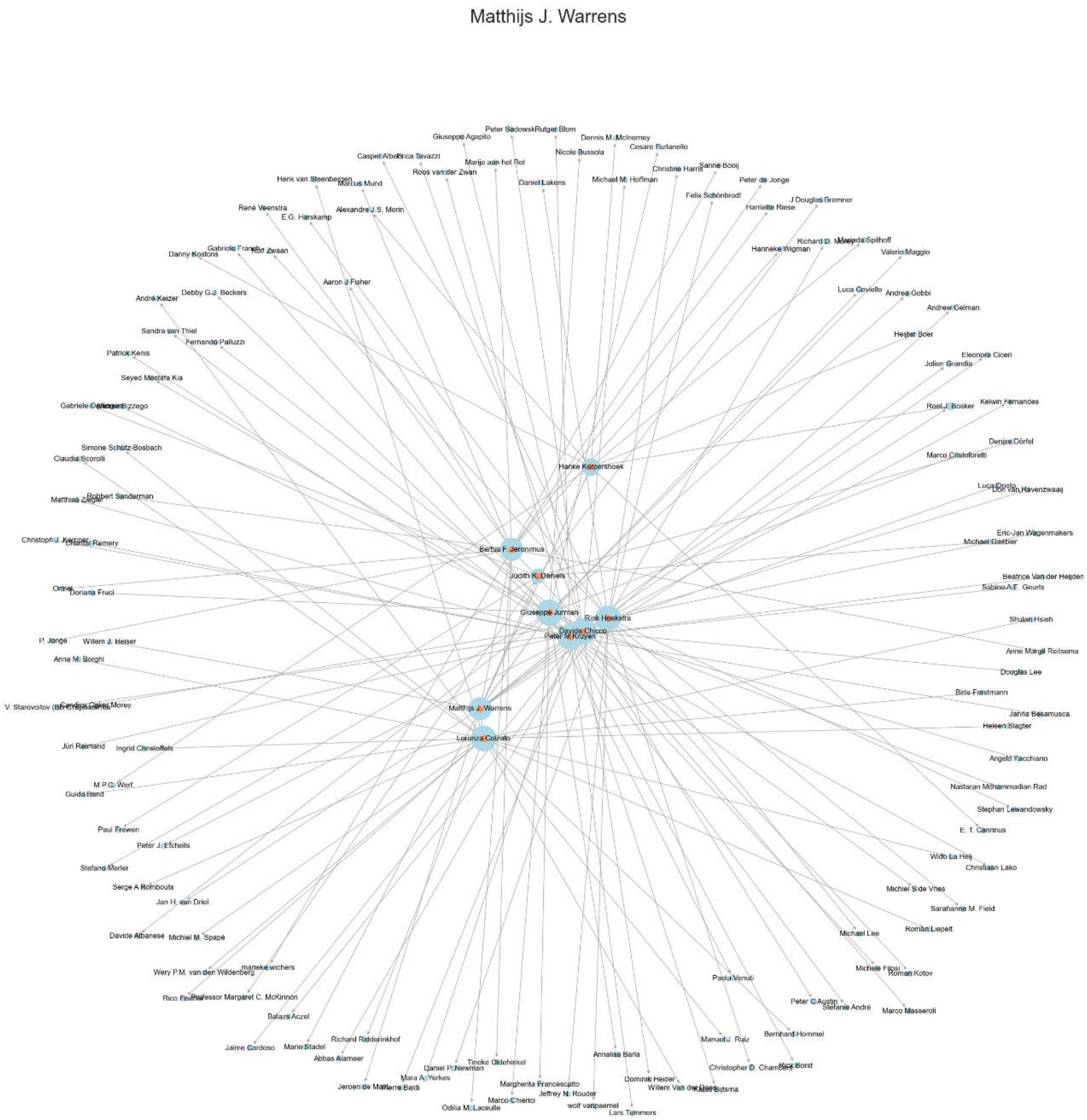


The degree distribution is presented in a histogram. Most of the nodes have a degree of 1 because the scrapping was stopped upon reaching the maximum number of nodes and the nodes discovered later do not have any other neighbours.

The important subnetwork is also created by finding the nodes which are more prominent in the network. The resting subnetwork is also visualized here.

Here, Co-authors are in light blue color.

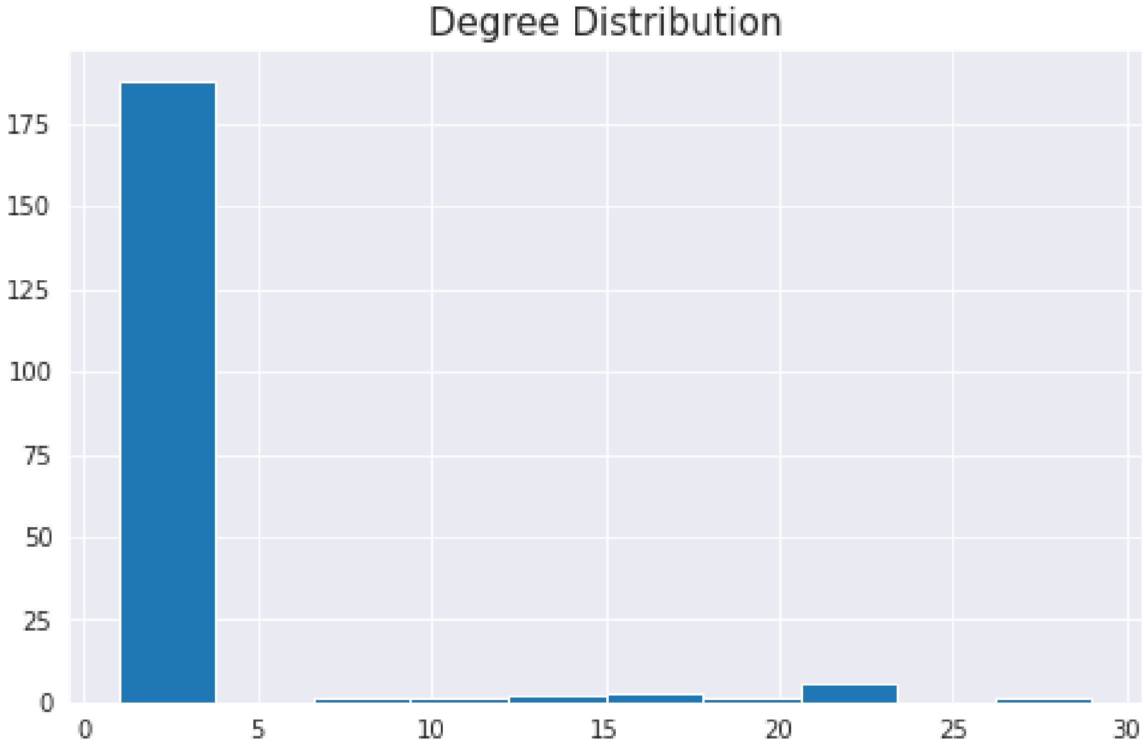
Nodes having high degree are in orange color.

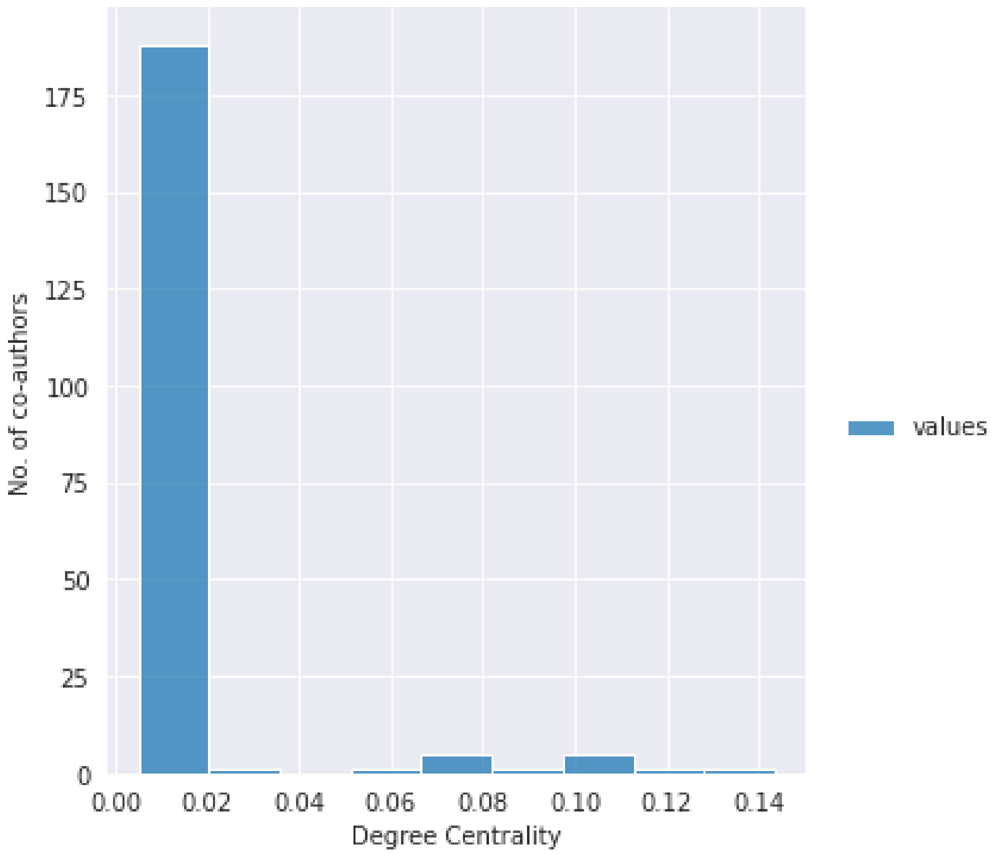


Similarly, histograms were made for the degree centrality, closeness centrality and betweenness centrality. The nodes at the outer lever have lower values of the centrality measures because of the fact that those nodes do not have any neighbours other than the nodes through which they were discovered in the scraping engine. The average values of the centrality measures are given here.

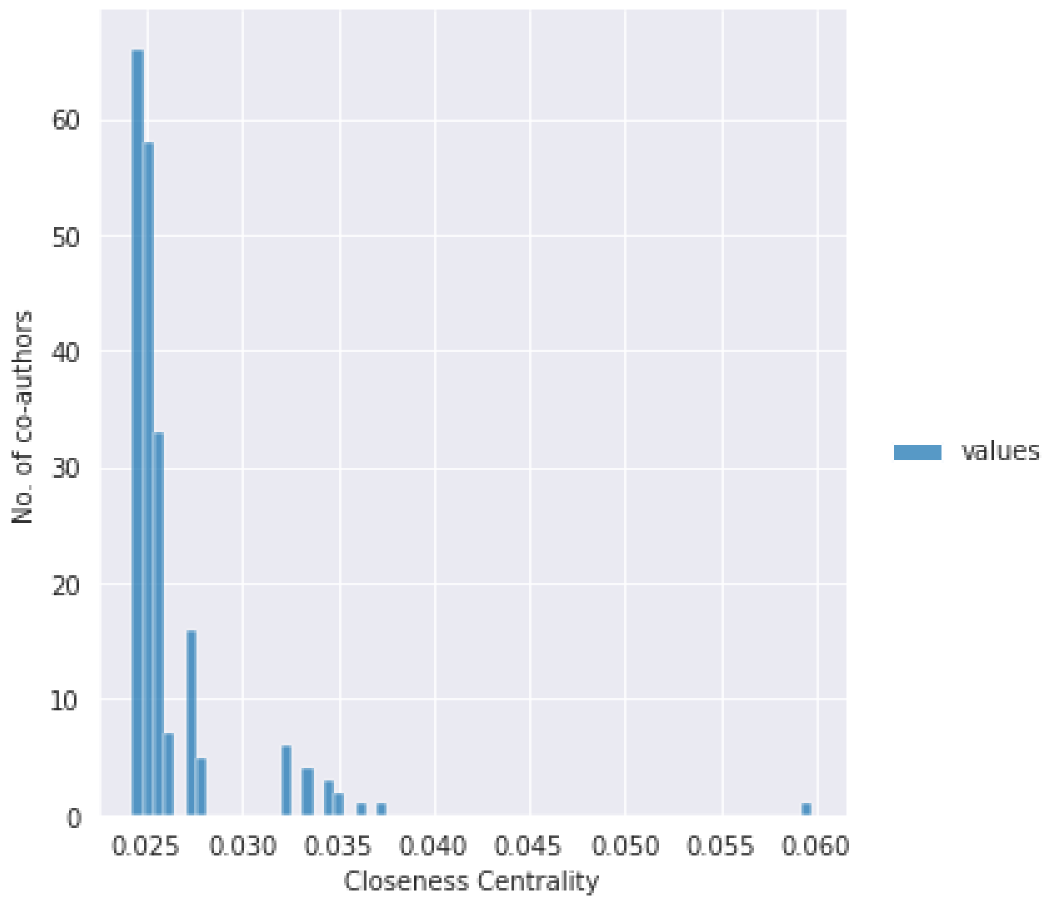
Average Degree Centrality: 0.0155887230514095

Average Closeness: 0.017139364890078987

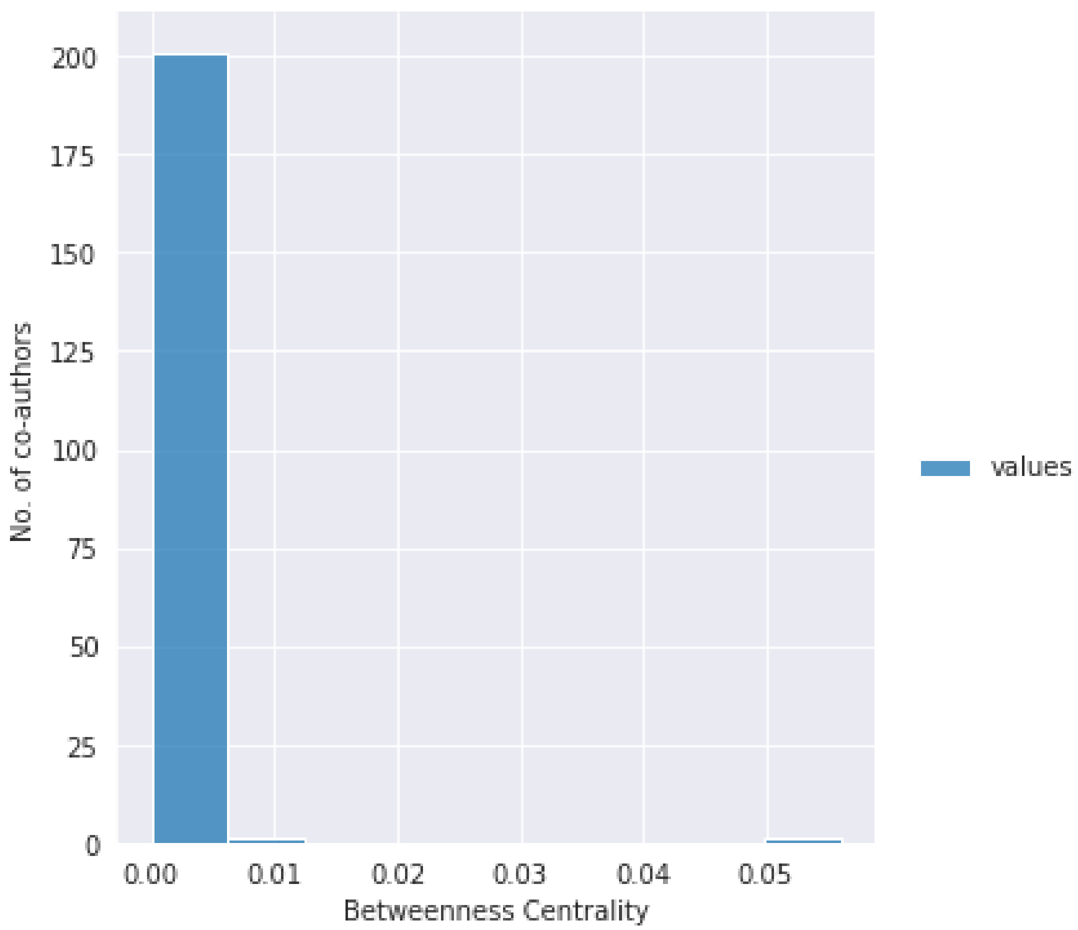




Average Betweenness: 0.0003893099607345318



No. of co-authors
60
50
40
30
20
10
0
0.025
0.030
0.035
0.040
0.045
0.050
0.055
0.060
Closeness Centrality
values



No. of co-authors
200
175
150
125
100
75
50
25
0
0.00
0.01
0.02
0.03
0.04
0.05
Betweenness Centrality
values

7. CONCLUSION, LIMITATIONS AND SCOPE OF FUTURE WORK

Froom this project, the network of co-authors was constructed and analyzed. The values of degree centrality, closeness centrality, betweenness centrality and page rank was calculated for each of the nodes and a sub network of the most important actors was constructed. From this study, the uses of co-authorship network can be realized and the information obtained can be employed in various fields.

The system made in this project can be improved in many ways. The current implementation only records the top twenty co-authors of a person. To get the list of all co-authors, other methods have to be devised or another scrapper must be used. The number of nodes can be increased to make the network bigger and analyze the communities that are formed in the network. The evolution of network over time is also not considered in this project and can be an interesting addition to this project, providing useful insights in the network.

8. REFERENCES

[1] A.L Barabási, H Jeong, Z Néda, E Ravasz, A Schubert, T Vicsek, Evolution of the social network of scientific collaborations, in Physica A: Statistical Mechanics and its Applications, Volume 311, Issues 3–4, 2002, Pages 590-614, ISSN 0378-4371, https://doi.org/10.1016/S0378-4371(02)00736-7.

[2] Yang Chen, Cong Ding, Jiyao Hu, Ruichuan Chen, Pan Hui, and Xiaoming Fu. 2017. Building and Analyzing a Global Co-Authorship Network Using Google Scholar Data. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1219–1224. DOI: https://doi.org/10.1145/3041021.3053056

[3] Mark EJ Newman, Coauthorship networks and patterns of scientific collaboration in Proceedings of the national academy of sciences 101 (suppl 1), 5200-5205 https://doi.org/10.1073/pnas.0307545100

[4] Perianes-Rodríguez, A., Olmeda-Gómez, C. & Moya-Anegón, F. Detecting, identifying and visualizing research groups in co-authorship networks. Scientometrics 82, 307–319 (2010). https://doi.org/10.1007/s11192-009-0040-z

[5] M. Fujita, H. Inoue and T. Terano, "Searching Promising Researchers through Network Centrality Measures of Co-author Networks of Technical Papers," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 2017, pp. 615-618. doi: 10.1109/COMPSAC.2017.205

APPENDIX

SAMPLE CODE:

```
import matplotlib.pyplot as plt
import networkx as nx
import urllib3
from bs4 import BeautifulSoup
from collections import defaultdict
import requests
import seaborn as sns
import pandas as pd


scholar = 'https://scholar.google.com/citations?user=x6fNSxcAAAAJ&hl=en'
nodes=100


def getGraph(seed, Nmax):
    urls = defaultdict(int)
    urls[seed]+=1
    newUrls = [seed]
    G = nx.DiGraph()

    def coAuthors(url):
        print(url)
        coUrls = []
        coNames = []
        response = requests.get(url)
        soup = BeautifulSoup(response.content, "html.parser")
        s = soup.body.findAll('a', {"tabindex": "-1"})
        egoName = soup.body.find('div', {"id": "gsc_prf_in"}).text
        print(egoName)
        print("******")
        if s:
            for i in s:
                if i.text=="Sort by citations" or i.text=="Sort by year" or i.text=="Sort by title":
                  continue
                coNames.append(i.text)
                #for network plot
                coUrls.append('http://scholar.google.nl'+ i['href'])
        for j in coUrls:
            urls[j] += 0
        for m in coNames:
            G.add_edge(egoName.split(',')[0], m.split(',')[0], weight = 1)
        return coUrls

    while newUrls:
        for k in urls.keys():
            # update url.values() first
```

```
            urls[k] += 1
        addUrls = []
        # get new-added authors, may have duplications.
        for i in newUrls:

            #coAuthors(i)
            coUrls = coAuthors(i)
            if coUrls:
                for j in coUrls:
                    addUrls.append(j)
        for m in set(addUrls):
            # get rid of the duplications
            urls[m] += 0
        newUrls = [k for k, v in urls.items() if v <= 1]
        # This is for updating the new coauthors and avoid the deadloop: a->b->a->......
        addUrls = []
        print(len(urls.keys()))
        if len(urls.keys()) > Nmax:
            print('more than '+str(Nmax)+' people now, break')
            break
        print(newUrls)
    return G



def getName(seed, Nmax):
    #urls = defaultdict(int)
    G = nx.DiGraph()
    response = requests.get(seed)
    soup = BeautifulSoup(response.content, "html.parser")
    egoName = soup.body.find('div', {"id": "gsc_prf_in"}).text
    print(egoName)
    return egoName


#get Name of the author
name = getName(scholar, nodes)


#get coauthor List
g = getGraph(scholar, nodes)


def plot_(g):
    plt.figure(figsize = (30, 30))
    pos = nx.spring_layout(g)
    nx.draw_networkx_labels(g, pos, font_color='k', font_size = 14)
    nx.draw(g, pos, node_size = 20, edge_color = 'grey', width = 0.4, arrows = True)
    #str1=name
    str2="'s Google Scholar Network"
```

```
    myTitle=str(name)+str2
    plt.title(myTitle, fontsize=40)
    #plt.xticks([])
    #plt.yticks([])
    plt.savefig("impnetwork.png")
    plt.show()
    #return plt


plot_(g)


#create Important subnetwork
def page_Rank(G):
    print("Google Page Rank Algoritham\n")
    rank = nx.pagerank(G)
    for k,v in rank.items() :
        print("Page Rank Centrality :",k,'\t',v)

    r = [x for x in rank.values()]
    rsum = sum(r)
    rlen = len(r)
    rfac = rsum/rlen
    Gt = G.copy()

    for k, v in rank.items():
        if v < rfac:
            Gt.remove_node(k)
    return Gt


Gt = page_Rank(g)


plot_(Gt)


mylist=[]
degree = g.degree()
for k,v in degree :
    print('Degree of Each Node :',k,'\t',v)
    mylist.append(v)


plt.figure(figsize = (8, 5))
plt.hist(mylist)
plt.title("Degree Distribution", fontsize = 15)
plt.show()


centality_value = []
```

```
centrality = nx.degree_centrality(g)
for each in centrality.items():
    print('Degree Centrality: ', each[0], '\t', each[1])
    centality_value.append(each[1])


avg_centrality = sum(centrality.values())/len(centrality)
print('Average Degree Centrality: ', avg_centrality)


#create dataframe
import pandas as pd
y=list(centrality.values())
x=list(centrality.keys())
data = {'keys':x,'values':y}
df1 = pd.DataFrame.from_dict(data)

sns.set_style('darkgrid')
a=sns.displot(df1)
a.set_axis_labels(x_var="Degree Centrality", y_var="No. of co-authors")


close = nx.closeness_centrality(g)
for each in close.items():
    print('Closeness Centrality: ', each[0], '\t', each[1])


avg_closeness = sum(close.values())/len(close)
print('Average Closeness: ', avg_closeness)


y=list(close.values())
x=list(close.keys())
data = {'keys':x,'values':y}
df1 = pd.DataFrame.from_dict(data)

sns.set_style('darkgrid')
a=sns.displot(df1)
a.set_axis_labels(x_var="Closeness Centrality", y_var="No. of co-authors")


btwn = nx.betweenness_centrality(g, weight='weight')
for each in btwn.items():
    print('Betweeness Centrality: ', each[0], '\t', each[1])


avg_betweenness = sum(btwn.values())/len(btwn)
print('Average Betweenness: ', avg_betweenness)


y=list(btwn.values())
x=list(btwn.keys())
```

```
data = {'keys':x,'values':y}
df1 = pd.DataFrame.from_dict(data)


sns.set_style('darkgrid')
a=sns.displot(df1)
a.set_axis_labels(x_var="Betweenness Centrality", y_var="No. of co-authors")


def plot2(g, strng):
  plt.figure(figsize=(35, 35))

  # 1. Create the graph
  df=nx.to_pandas_edgelist(g)
  #print(df)
  # 2. Create a layout for our nodes
  layout = nx.spring_layout(g,iterations=50)

  # 3. Draw the parts we want
  nx.draw_networkx_edges(g, layout, edge_color='#AAAAAA')

  coauthor = [node for node in g.nodes() if node in df.target.unique()]
  size = [g.degree(node) * 80 for node in g.nodes() if node in df.target.unique()]
nx.draw_networkx_nodes(g, layout, nodelist=coauthor, node_size=size, node_color='lightbl
ue')

  people = [node for node in g.nodes() if node in df.source.unique()]
nx.draw_networkx_nodes(g, layout, nodelist=people, node_size=100, node_color='#AAAA
AA')

  high_degree_people = [node for node in g.nodes() if node in df.source.unique() and g.degree
(node) > 1]
  nx.draw_networkx_nodes(g, layout, nodelist=high_degree_people, node_size=100, node_co
lor='#fc8d62')

  coauthors_dict = dict(zip(coauthor, coauthor))
  nx.draw_networkx_labels(g, layout, labels=coauthors_dict)

  plt.axis('off')

  plt.title("Justin Wolfer's Google Scholar Network", fontsize=30)
  # 4. Tell matplotlib to show it
  plt.savefig(strng)
  plt.show()
plot2(g, "network.png")


plot2(Gt, "impnetwork.png")
```