Data Visualization and Analytics of Columbia's Website Based on IBM System G

Yue Yu, Chen Xu, Zhongzhu Jiang
Electrical Engineering and Computer Engineering
Columbia University
yy2636@columbia.edu
cx2179@columbia.edu
zj2195@columbia.edu

Abstract— **We write our own crawler to crawl on Columbia University's website to get the link information as our dataset. Then we use the IBM System G as our tool to do the data visualization and some graph characteristic analysis based on the data and the gshell function IBM System G provided.**

Keywords-component; IBM System G; Centrality; PageRank; Data Visualization; Clustering Coefficient; Columbia University

I. INTRODUCTION

The IBM System G is a comprehensive set of graph computing tools. Its key feature compared with the traditional analytic systems is that it is designed to deal with the data linked with each other. And the data on the Internet especially the links on a website fit this feature very well.

We think using a new tool to do some analytics on the university's own website is fun. As there is few data analytics on that, we can show the construction of the university's website and find more about our university in this way.

In this project, we will establish a database about the link information of Columbia University's website. Using System G, we will analyze and visualize the topological structure of our university's websites.

II. RELATED WORKS

In the homework 3 of course EECS 6893 Big Data Analytics, we download and configure the IBM System G Graph Tools. Using the URL links in the Wikipedia data, we create a knowledge graph, try some queries to find relevant terms, and show some visualizations of queries. We know the basics of System G and graph analytics.

In this project, we will go further in System G. More advanced algorithms will be applied to analyze the graph, so that we will find more about our university's websites topological framework. And we collect link information and establish a dataset on our own.

III. SYSTEM OVERVIEW

A. System G

IBM System G is the first complete software stack for all aspects of Graph Computing. Graph database is a tool for efficiently managing large-scale graph data, especially important for contextual and relationship analysis. Graph analytics is important for finding the important vertices or edges that are more central, that are clustered, or that form abnormal patterns. Graphical models are essential to artificial intelligence, information reasoning and predictive analysis, which requires combining many factors to create actionable insights.

We use System G to implement data analytics and data visualization. Centralities, communities and Pagerank are applied in data analytics.

B. Dataset

We use the Scrapy, an open source web crawler framework to write the crawler and crawl the link data from the website. The crawler is written in Python.

The crawler starts crawling from the homepage of the university(http://www.columbia.edu/), and saves the link information into csv file.

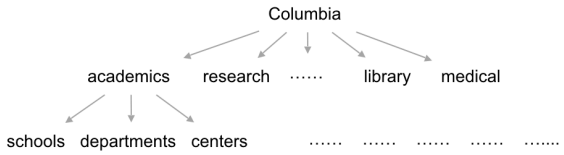

Fig.1 the Columbia University's websites structure

The original data the crawler get is the url, the title of the current page and all the links on the current page and corresponding titles.

	A	B	C	D
	表格 1			
1	url	child_title	child_url	title
2	http://www.columbia.edu/	About Columbia I Columbia University I	http://www.columbia.edu/content/about-(	Columbia University in the City of New York
3	http://www.columbia.edu/	Columbia Global	http://beta.global.columbia.edu/	Columbia University in the City of New York
4	http://www.columbia.edu/	About Columbia I Columbia University I	http://www.columbia.edu/content/about-(	Columbia University in the City of New York
5	http://www.columbia.edu/	Innovative and Compassionate Health I	http://cumc.columbia.edu/patient-care	Columbia University in the City of New York
6	http://www.columbia.edu/	New York City I Columbia News	http://news.columbia.edu/content-topics/	Columbia University in the City of New York
7	http://www.columbia.edu/	A Culture of Lifelong Inquiry I Columbi	http://cumc.columbia.edu/education	Columbia University in the City of New York
8	http://www.columbia.edu/	CU Directory of Classes	http://www.columbia.edu/cu/bulletin/uwb	Columbia University in the City of New York
9	http://www.columbia.edu/	A World Leader in Health Care Researc	http://cumc.columbia.edu/research	Columbia University in the City of New York
10	http://www.columbia.edu/	Research I Columbia News	http://news.columbia.edu/content-topics/	Columbia University in the City of New York
11	http://www.columbia.edu/	Columbia Engineers Build Biologically I	http://engineering.columbia.edu/columbi	Columbia University in the City of New York
12	http://www.columbia.edu/	Columbia Alumni Association	http://alumni.columbia.edu/	Columbia University in the City of New York
13	http://www.columbia.edu/	Giving to Columbia University: Home	http://giving.columbia.edu/	Columbia University in the City of New York
14	http://www.columbia.edu/	Columbia Community Service	http://communityservice.columbia.edu/	Columbia University in the City of New York

Fig. 2 the original data in csv file

After we get the data, we also wrote a Python script to parse the data we get, split it into the vertex information and the edge information. We use the url of the web page as the ID and the title as the property of the vertex, the source of the edge is the url and the target is the child url.

The size of the data we get is 223.8MB, 107,777 vertices and 933,763 edges.

IV. ALGORITHM

A. *Centrality Analysis*

The vertex's centrality can indicate its importance in the network. The centrality of a node can be measured using degree centrality, closeness centrality, and betweenness centrality. We conduct the centrality analysis based on the measurement we mentioned above and export the graph into csv file using python script to do ranking.

In the gShell, the shortest distance between a vertex and any other vertex in the graph is calculated using Dijkstra's algorithm. We don't take the edge weights into consideration since we don't get the weight information.

Degree Centrality

Degree centrality, which is defined as the number of links incident upon a node. The degree can be interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network. We usually define two separate measures of degree centrality, namely indegree and outdegree.

Indegree is a count of the number of ties directed to the node and outdegree is the number of ties that the node directs to others.

Closeness Centrality

In connected graphs there is a natural distance metric between all pairs of nodes, defined by the length of their shortest paths. The more central a node is the lower its total distance from all other nodes.

Betweenness centrality

Betweenness centrality quantifies the number of times a node acts as a bridge along the shortest path between two other nodes.

B. *Clustering Coefficient*

In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together.

The function in gShell computes the local clustering coefficient for every vertex.

The local clustering coefficient of a vertex (node) in a graph quantifies how close its neighbors are to being a complete graph

.

C. *PageRank*

PageRank is an algorithm used by Google Search to rank websites in their search engine results. The rank value indicates an importance of a particular page. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

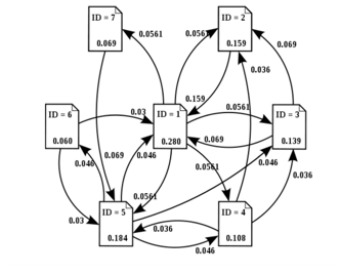

Fig. 3 PageRank

V. EXPERIMENT RESULTS

A. Create the graph

Since the link on the web page has the direction itself, we treat the graph of the link as a directed graph.

The environment we use is gShell. gShell is a shell-like environment on top of IBM System G Graph Store. It allows users to create, update and query multiple graph stores simultaneously and supports both directed and undirected graphs.

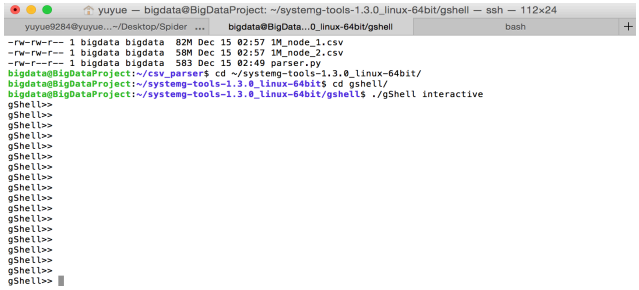

Fig. 4 gShell interactive

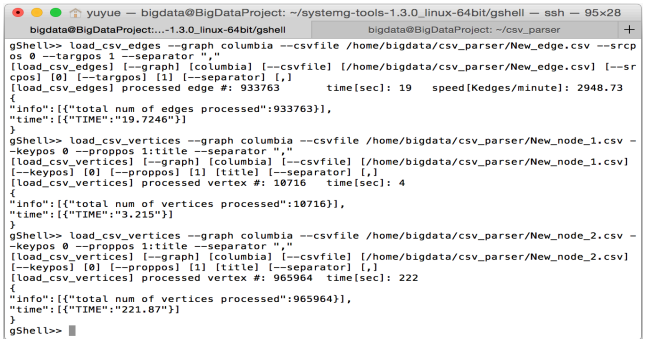

Fig. 5 load vertices and edges from csv files

```
gShell>> create --graph columbia --type directed
[create] [--graph] [columbia] [--type] [directed]
{
"info":[{"MESSAGE":"store [columbia] is created!"}]
}
gShell>>
```

Fig. 6 create the graph

Use the open command to load the graph to the memory to accelerate the processing speed.

```
gShell>> open --graph columbia
[open] [--graph] [columbia]
{
"info":[{"MESSAGE":"store is opened"}]
}
```

Fig. 7 open the graph

B. Data visualization

The following pictures are the visualization of the data we get, but we found that if the data has too many vertices and edges, the visualization will fail. so we used a rather smaller dataset to do this part instead. We used the IBM System G Lite to do the visualization.

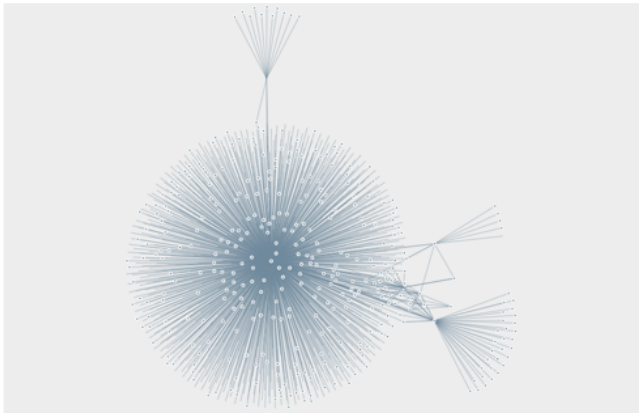

Fig. 8 Center: “http://www.columbia.edu/content/az-index.html” Depth: 2

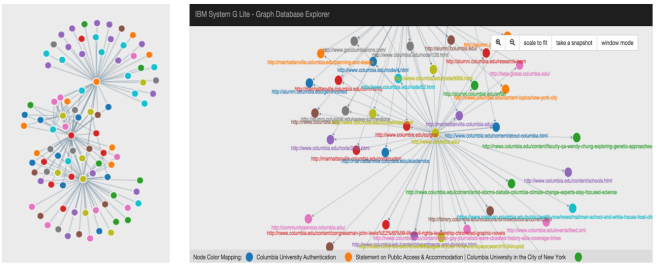

Fig. 9 Center: “http://www.columbia.edu/” Depth: 2

C. Data Analysis

Centrality

We can use the command to find the degree which has the largest degree.

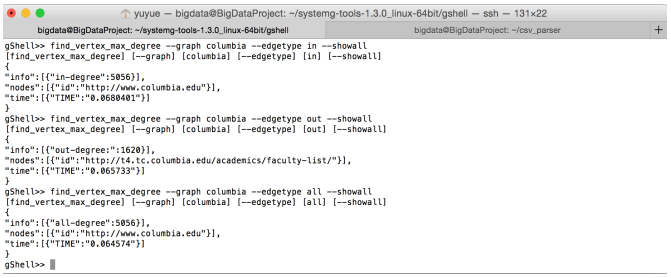

Fig. 10 run degree centrality

From the result we can see that the vertex with the most out degree (1620) is http://t4.tc.columbia.edu/academics/faculty-list/, and the one with the most in degree(5096) is http://www.columbia.edu/.

Then we can use the find neighbor command to view in detail.

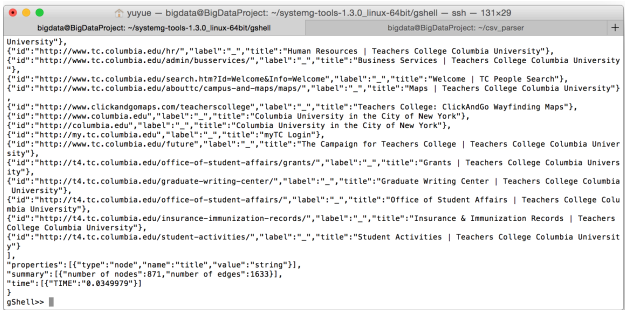

Fig. 11 find neighbors
http://t4.tc.columbia.edu/academics/faculty-list/

We show the top 10 web pages that has the most indegree (below) and the outdegree (above). The 1st and 2nd page which has the largest indegree are actual the same page (main page of the university).

```
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 4 0
['http://t4.tc.columbia.edu/academics/faculty-list/', '1620']
['http://academiccommons.columbia.edu/catalog/browse/subjects', '1513']
['http://www.columbia.edu/content/az-index.html', '1490']
['http://allivyfair.ei.columbia.edu/recruiters/', '1424']
['http://allivyfair.ei.columbia.edu/recruiters/category/masters/', '1162']
['http://allivyfair.ei.columbia.edu/recruiters/category/u-s-citizenpermanent-resident/', '1134']
['http://allivyfair.ei.columbia.edu/recruiters/category/bachelors/', '1134']
['http://www8.gsb.columbia.edu/courses/career-track/entrepreneurship-social-entrepreneurship', '1068']
['http://www8.gsb.columbia.edu/courses/divisions/management', '972']
['http://allivyfair.ei.columbia.edu/recruiters/category/permanent-full-time/', '967']
['http://allivyfair.ei.columbia.edu/recruiters/category/environmental/', '966']
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 5 0
['http://www.columbia.edu', '5056']
['http://www.columbia.edu/', '4117']
['http://medren.columbia.edu', '3890']
['http://www8.gsb.columbia.edu/centennial', '2450']
['http://medren.columbia.edu/feed/', '2207']
['http://library.columbia.edu/about.html', '1953']
['http://library.columbia.edu/research/askalibrarian.html', '1882']
['http://www8.gsb.columbia.edu/researcharchive/', '1645']
['http://www.tc.columbia.edu', '1630']
['http://www8.gsb.columbia.edu/about-us', '1604']
['http://www8.gsb.columbia.edu/cbs-directory', '1589']
```

Fig. 12 results of degree centrality

The results of closeness centrality are as follows.

```
gShell>> analytic_closeness_centrality --graph columbia --ignoreedgeweight
[analytic_closeness_centrality] [--graph] [columbia] [--ignoreedgeweight]
"",
"summary":[
{"stat":"avg","metric":"analytic_closeness","value":114.433197},
{"stat":"max","metric":"analytic_closeness","value":4153.108398},
{"stat":"min","metric":"analytic_closeness","value":0.000000}
],
"time":[{"TIME":"237.138"}]
}
gShell>>
```

Fig. 13 analytic closeness centrality

We show the top 10 which has the highest and lowest (other than 0) closeness centrality web pages, since the gShell uses the Opsahl 2010 formula: C(i) = sum(1/shortest_distance(i, j)) for all j!=i to calculate the closeness centrality. The higher the value is, the web page is more in important in the graph.

```
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 6 0
['https://www.careereducation.columbia.edu/findajob/howtointernship/startupfund/siffaqs', '4153.108398']
['https://www.careereducation.columbia.edu/findajob/howtointernship/columbia-engineering-internship-fund/faq', '4152.941895']
['https://www.careereducation.columbia.edu/findajob/howtointernship/alumni/apif-faq', '4152.941895']
['https://www.careereducation.columbia.edu/CCEandEmployers/EmployersOnCampus', '4093.360840']
['https://www.careereducation.columbia.edu/findajob/intlresources/intloppor', '4059.907959']
['https://www.careereducation.columbia.edu/resources/industry/eng', '4030.931396']
['https://www.careereducation.columbia.edu/resources/leadership-development-programs', '3944.860840']
['https://www.careereducation.columbia.edu/resources/industry/media/film', '3889.871582']
['https://www.careereducation.columbia.edu/resources/industry/media/publishing', '3888.038574']
['https://www.careereducation.columbia.edu/resources/industry/arts/arts-management', '3887.205811']
['https://www.careereducation.columbia.edu/findajob/gap-year/sample-organizations', '3884.528320']
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 6 1
['http://library.cumc.columbia.edu/contact-us', '1.500000']
['http://www.journalism.columbia.edu/event/15137class=more', '2.000000']
['http://www.ldeo.columbia.edu/LCSN/Eq/Ena/980925.html', '2.000000']
['http://www.ldeo.columbia.edu/LCSN/Eq/Ena/20000210.html', '2.000000']
['http://www.columbia.edu/~gwt2102/index.html', '2.000000']
['http://calendar.columbia.edu/sundial/webapi/get.php?br=ais&con=embedded&id=23888&vt=detail', '2.000000']
['http://calendar.columbia.edu/sundial/webapi/get.php?br=ais&con=embedded&id=31267&vt=detail', '2.000000']
['http://calendar.columbia.edu/sundial/webapi/get.php?br=ei_brand&con=standalone&id=31431&vt=detail', '2.000000']
['http://calendar.columbia.edu/sundial/webapi/get.php?br=ais&con=embedded&id=28715&vt=detail', '2.000000']
['http://calendar.columbia.edu/sundial/webapi/get.php?br=ais&con=embedded&id=30215&vt=detail', '2.000000']
['https://calendar.columbia.edu/sundial/webapi/get.php?br=default&id=80638&vt=detail', '2.000000']
```

Fig. 14 results of closeness centrality

The results of betweenness centrality are as follows.

```
gShell>> analytic_betweenness_centrality --graph columbia
[analytic_betweenness_centrality] [--graph] [columbia]
"",
"summary":[
{"stat":"avg","metric":"analytic_betweenness","value":0.000000},
{"stat":"max","metric":"analytic_betweenness","value":0.001371},
{"stat":"min","metric":"analytic_betweenness","value":0.000000}
],
"time":[{"TIME":"733.358"}]
}
```

Fig. 15 analytic betweenness centrality

We also give the ranking result of the betweenness centrality. The higher the value is, the more the shortest path is going through it. We can see in this case the homepage of the university is the center of the web this also correspond with it has the largest in degree.

```
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 1 0
['http://www.columbia.edu/', '0.001371']
['http://www.columbia.edu/node/1846.html', '0.001344']
['http://www.columbia.edu/content/az-index.html', '0.001316']
['http://library.columbia.edu/subject-guides.html', '0.001136']
['http://library.columbia.edu/subject-guides/africa.html', '0.001068']
['http://library.columbia.edu/locations/business.html', '0.000942']
['https://clio.columbia.edu/', '0.000925']
['http://www8.gsb.columbia.edu/faculty-research/research/resources', '0.000907']
['https://clio.columbia.edu/academic_commons', '0.000900']
['http://academiccommons.columbia.edu/catalog/browse/subjects', '0.000735']
['http://library.columbia.edu/news.html', '0.000706']
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 1 1
['http://cuit.columbia.edu/cuit/it-security-practices/using-strong-passwords', '0.000001']
['http://cuit.columbia.edu/cuit/it-security-resources/social-networking-dos-donts', '0.000001']
['http://cuit.columbia.edu/cuit/it-security-practices/filesharing-networks', '0.000001']
['http://cuit.columbia.edu/cuit/it-security-practices/anti-malware', '0.000001']
['http://cuit.columbia.edu/cuit/it-security-resources/cuit-security-how-tos', '0.000001']
['http://cuit.columbia.edu/cuit/it-security-practices/information-security-tips-tricks/finding-your-ip-mac-addresses', '0.000001']
['http://library.columbia.edu/locations/burke/rare_books.html', '0.000001']
['http://library.columbia.edu/technology/tools/libx_tool.html', '0.000001']
['http://www.columbia.edu/cu/arthistory/', '0.000001']
['https://blogs.cul.columbia.edu/outdoorsculpture/', '0.000001']
['http://library.columbia.edu/locations/science-engineering/services.html', '0.000001']
```

Fig. 16 results of betweenness centrality

(1) Clustering Coefficient

```
gShell>> analytic_clustering_coefficient --graph columbia
[analytic_clustering_coefficient] [--graph] [columbia]
"",
"summary":[
{"stat":"avg","metric":"analytic_coefficient","value":0.101254},
{"stat":"max","metric":"analytic_coefficient","value":1.000000},
{"stat":"min","metric":"analytic_coefficient","value":0.000000}
],
"time":[{"TIME":"16.7218"}]
}
```

Fig. 17 calculate clustering coefficient

From the results, we can see some small websites are more likely to cluster together.

```
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 2 0
['http://irvinginstitute.columbia.edu/cusp', '0.999517']
['http://spa.columbia.edu', '0.999517']
['http://ps.columbia.edu/CERS/discovery-process', '0.999517']
['http://cumc.columbia.edu/dept/taub/genomicscore', '0.999517']
['http://www.hhs.gov/ohrp', '0.999517']
['http://www.nih.gov/health/clinicaltrials/index.htm', '0.999517']
['http://www.columbiastemcell.org', '0.999517']
['http://irvinginstitute.columbia.edu/tgmouse', '0.999517']
['http://www.ncbi.nlm.nih.gov/sites/entrez?db=PubMed', '0.999517']
['http://www1.wfubmc.edu/wfupc', '0.999517']
['http://irvinginstitute.columbia.edu/resources/request_form_login.html', '0.999517']
bigdata@BigDataProject:~/csv_parser$ python sort.py columbia 2 1
['http://pocketknowledge.tc.columbia.edu/home.php/editprofile', '0.000001']
['http://academiccommons.columbia.edu/catalog/browse/subjects', '0.000021']
['http://www.tc.columbia.edu/az.htm?Id=AZ+List&Info=AZ+List', '0.000034']
['http://www.tc.columbia.edu/az.htm?resources=staff', '0.000034']
['http://www.tc.columbia.edu/az.htm?resources=students', '0.000034']
['http://www.tc.columbia.edu/calendar.htm', '0.000046']
['http://www.tc.columbia.edu/housing/', '0.000046']
['http://www.tc.columbia.edu/administration/security', '0.000046']
['http://www.tc.columbia.edu/webviewer/', '0.000046']
['http://www.tc.columbia.edu/facilities/', '0.000046']
['http://www.tc.columbia.edu/admissions/index.asp?Id=Prospective+Students&Info=On-Campus+Events#3-tab', '0.000046']
```

Fig. 18 Ranking the data by the clustering coefficient (ignore)

(2) PageRank

```
gShell>> analytic_pagerank --graph columbia
[analytic_pagerank] [--graph] [columbia]
{
"info":[{"MESSAGE":"num of iterations: 3"}],
"time":[{"TIME":"0.656291"}]
}
```

Fig. 19 calculate PageRank

From the result we can see that the main page which has the most indegree also has the highest PageRank value.

```
bigdata@BigDataProject:~/csv_parser$ python sort_pr.py columbia 1 0
['http://www.columbia.edu/', '0.001371']
['http://www.columbia.edu/node/1846.html', '0.001344']
['http://www.columbia.edu/content/az-index.html', '0.001316']
['http://library.columbia.edu/subject-guides.html', '0.001136']
['http://library.columbia.edu/subject-guides/africa.html', '0.001068']
['http://library.columbia.edu/locations/business.html', '0.000942']
['https://clio.columbia.edu/', '0.000925']
['http://www8.gsb.columbia.edu/faculty-research/research/resources', '0.000907']
['https://clio.columbia.edu/academic_commons', '0.000900']
['http://academiccommons.columbia.edu/catalog/browse/subjects', '0.000735']
['http://library.columbia.edu/news.html', '0.000706']
```

Fig. 20 Top 10 pages which have the highest PageRank value

D. Some problems

(1) Problems with Analysis

The link data we get is still not enough. Lots of the link information for the departments is missing.

When processing the vertex data, we find that the "http://www.columbia.edu/" and "http://www.columbia.edu" are treated as the different node since the url is slightly different but they are actual the same page.

Also, some redirecting link is not handled properly,http://www.columbia.edu/cu/lweb is the link to library but it is stored as a different vertex.

All these things will affect the accuracy of the analysis.

(2) Problems with System G

We found that if we add the restart at the end of the command, it will work.

Some time the graph will disappear itself.

When using the python script to create the graph, it can't be accessed using gshell.

The vertex id some time is not shown properly, "http://www.columbia.edu/" becomes "http:".

When the file contains too many nodes and edges, the visualization will fail.

```
gShell>> analytic_pagerank --graph columbia --prop "page_rank"
[analytic_pagerank] [--graph] [columbia] [--prop] [page_rank]
terminate called after throwing an instance of 'int'
Aborted (core dumped)
```

Fig. 21 System G error

VI. CONCLUSION

According to the graph analysis, we show something about the Columbia University's websites topological framework. We also find the importance of different websites by using centrality, cluster coefficient and PageRank analysis. From the data visualization, we can find special relationships between different website links in detail. Furthermore, we can also see the priority and importance of all the website links in the whole website system.

In the future, we can further improve the dataset to analyze more accurately. And we can use it to help figure out some problems like low click rate or difficulty of access in terms of topological structure.

ACKNOWLEDGMENT

We would like to thank Professor CHING-YUNG LIN and all TAs for providing keen insights and keeping us on track.

REFERENCES

[1] Description of system G. (2014). Retrieved December 21, 2015, from http://systemg.research.ibm.com/doc/gshell.html.

[2] Definition and description of centrality. (2015). Retrieved December 21, 2015, from https://en.wikipedia.org/wiki/Centrality.

[3] Description of scrapy. (2013). Retrieved December 21, 2015, from http://scrapy.org/.

[4] Definition and description of pagerank. (2015). Retrieved December 21, 2015, from https://en.wikipedia.org/wiki/PageRank.

[5] Description of gshell. (2014). Retrieved December 21, 2015, from http://systemg.research.ibm.com/doc/gshell.html.