Assignment 4

CS595

Introduction to Web Science

Old Dominion University

Computer Science

Due: 11:59 pm Oct 10

Lulwah Alkwai

Question One-

From your list of 1000 links, choose 100 and extract all of the links from those 100 pages to other pages. We're looking for user navigable links, that is in the form of:

```
<A href=foo>bar</a>
```

```
We're not looking for embedded images, scripts, <link> elements,
etc.  You'll probably want to use BeautifulSoup for this.
```

```
For each URI, create a text file of all of the outbound links from
that page to other URIs (use any syntax that is easy for you).  For
example:
```

```
site:
http://www.cs.odu.edu/~mln/
links:
http://www.cs.odu.edu/
http://www.odu.edu/
http://www.cs.odu.edu/~mln/research/
http://www.cs.odu.edu/~mln/pubs/
http://ws-dl.blogspot.com/
http://ws-dl.blogspot.com/2013/09/2013-09-09-ms-thesis-http-mailbox.html
etc.
```

Upload these 100 files to github (they don't have to be in your report).

Answer One-

In this question I created a new path to save the new files in and opened the previous file that had all links. Then I looped in the file and for each link I used beautiful soup to search for links that are not an image or a javascript or an existing link. And save 100 sites and inner links; using the site hash name as file name; in a new file in the created path.

Code-

Attached are:
(a4q1.py): the python code.
(finallinks.txt): the list of 1000 links.
(Allnew100links): all the files containing the site and links.

Question Two-

Using these 100 files, create a single GraphViz dot file of the resulting graph.
Learn about dot at:

```
Examples:
http://www.graphviz.org/content/unix
http://www.graphviz.org/Gallery/directed/unix.gv.txt

Manual:
http://www.graphviz.org/Documentation/dotguide.pdf

Reference:
http://www.graphviz.org/content/dot-language
http://www.graphviz.org/Documentation.php

Note: you'll have to put explicit labels on the graph, see:
https://gephi.org/users/supported-graph-formats/graphviz-dot-format/

(note: actually, I'll allow any of the formats listed here:

https://gephi.org/users/supported-graph-formats/

but dot is probably the simplest.)
```

Answer Two-

Here I opened the path created in Q1 and looped inside the file and for each file I extracted the site name and links and saved in a dictionary,to be used for labels as well as links and saved it in a new dot file that I created.

Code-

Attached are:
(a4q2.py): the python code.
(graphviz.dot): the dot file to be used in Gephi.

Question Three-

Download and install Gephi:

https://gephi.org/

Load the dot file created in #2 and use Gephi to:

- visualize the graph (you'll have to turn on labels)
- calculate HITS and PageRank
- avg degree
- network diameter
- connected components

Put the resulting graphs in your report.

You might need to choose the 100 sites with an eye toward creating a graph with at least one component that is nicely connected. You can probably do this by selecting some portion of your links (e.g., 25, 50) from the same site.

Answer Three-

I installed the tool, it was easy to use, I read the documentation for the tool and looked at a tutorial on Youtube for filtering; called Gephi Tutorial: Filtering Networks,which made it even more clear:
`http://www.youtube.com/watch?v=UrrWA_t1rjc`

The first time I ran the tool on my dot file it gave me a non-connected graph, so I added some search options in the python code on the first question and got a better looking graph.

The initial graph Figure1 was the first look at the links.It actually gave meaningless results. So I used a different layout Figure2 which gave a better view of what we are dealing with.

Then I used different filters and looked at the results. One of the filters I used was the Giant Component and resulted with Figure3 which gave the largest connected components.

Finally, I used the coloring option on edges to have a better view on the connected nodes Figure4.

At the end I used Gephi tool to calculate some statistics, which are listed below:

- HITS Authorities: Figure5
- Hits Hubs: Figure6
- Page Rank: Figure7
- Average degree-Degree distribution: Figure8
- Average Degree-Indegree distribution:Figure9
- Average Degree-Outdegree distribution: Figure10
- Network Diameter-Betweenness Centrality Distribution Figure11
- Network Diameter-Closeness Centrality Distribution: Figure12
- Network Diameter Eccentricity Distribution: Figure13
- Network Diameter Connected Component cc-size-distribution: Figure14.

  ...

Note: Also attached in Q3 File the reports for each part.

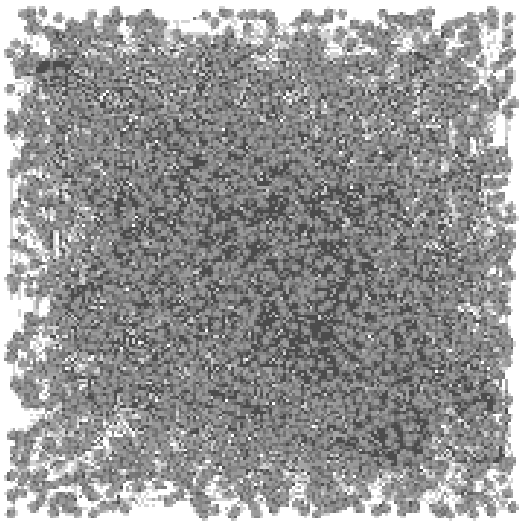

Figure 1: Initial View

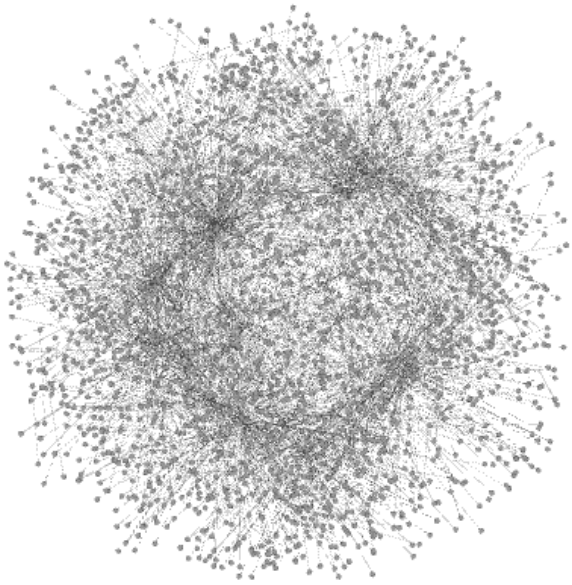

Figure 2: Using Yifan Hu' Layout

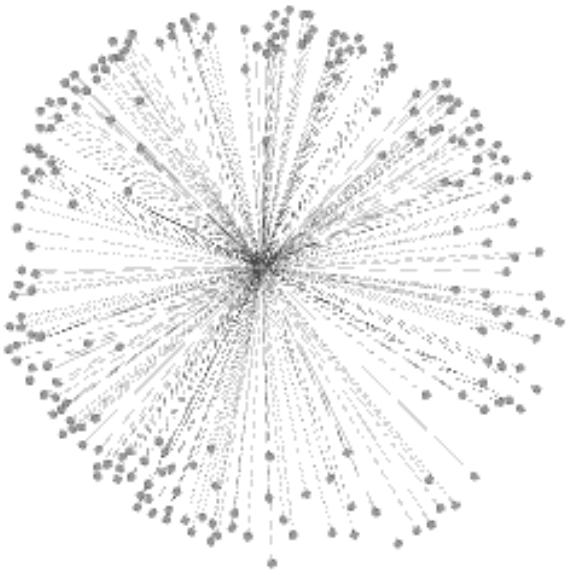

Figure 3: Using Filters

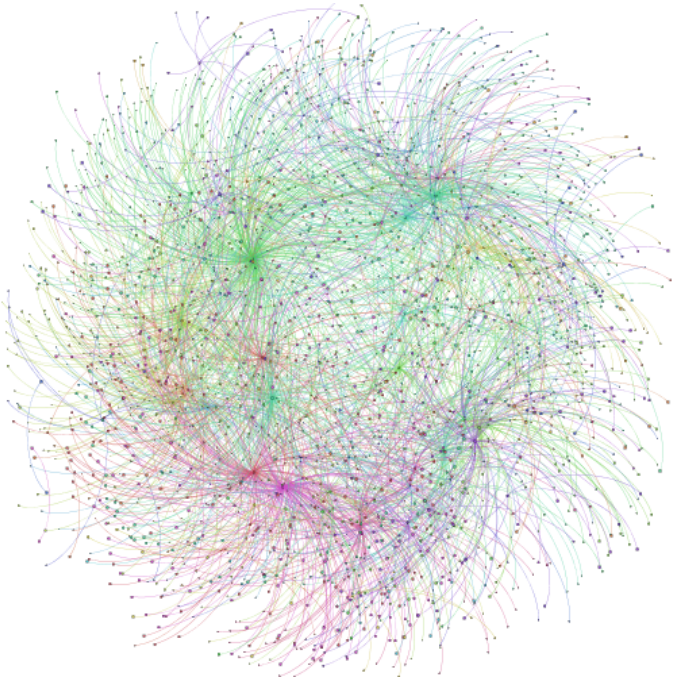

Figure 4: Color Nodes

Figure 5: Hits Authorities

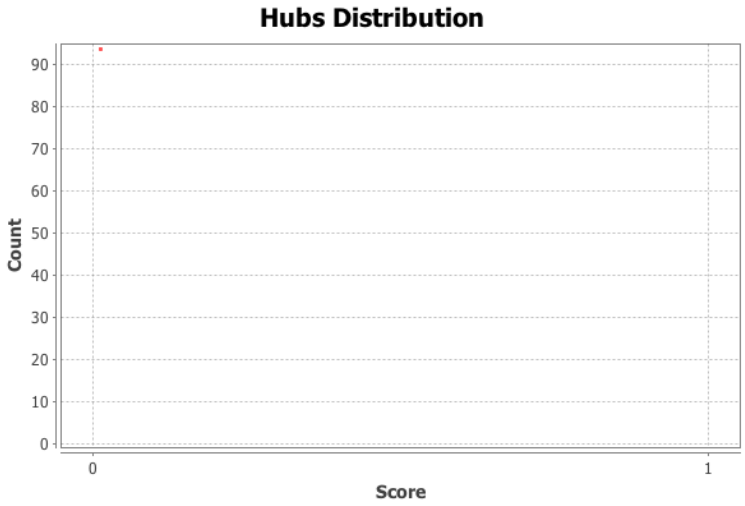

Figure 6: Hits Hubs

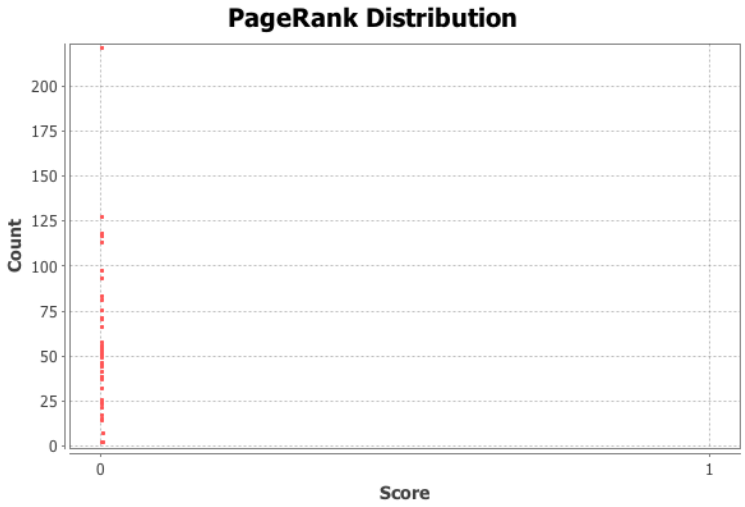

Figure 7: Page Ranks

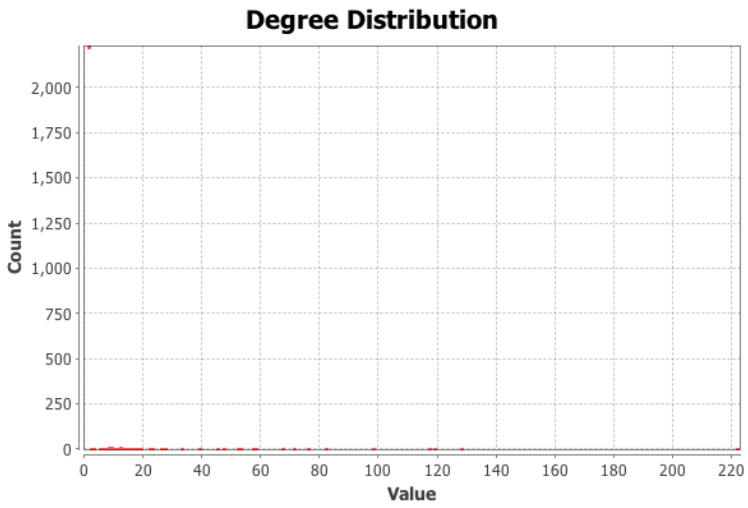

Figure 8: Average Degree: Degree Distribution

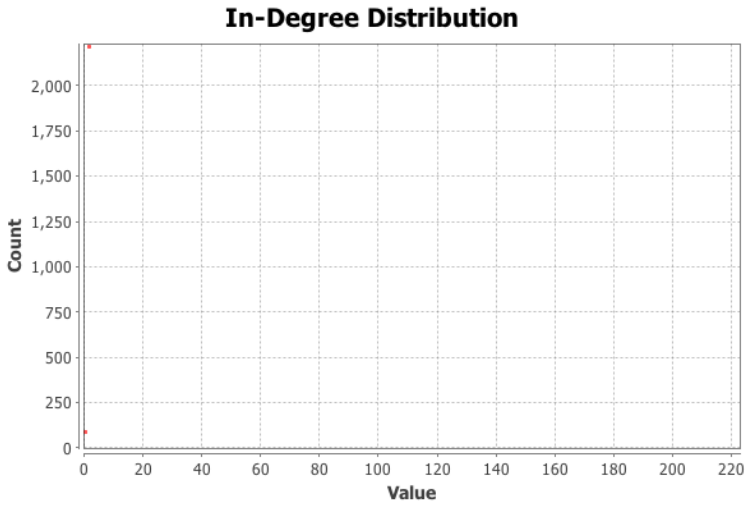

Figure 9: Average Degree: Indegree Distribution

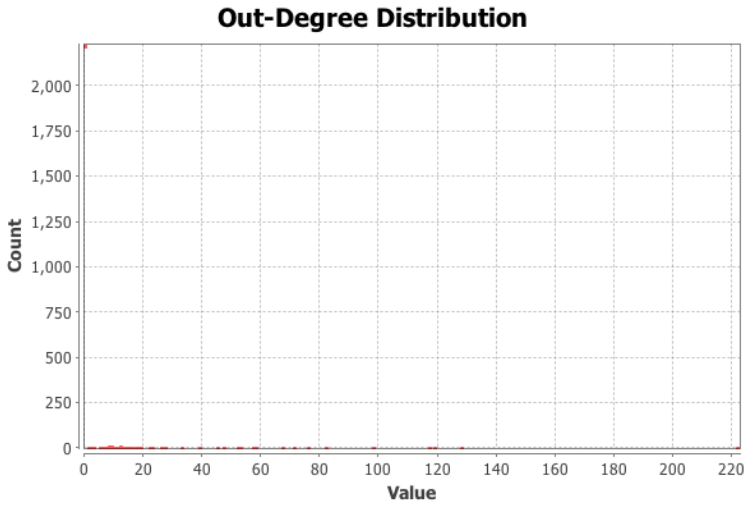

Figure 10: Average Degree: Outdegree Distribution

Centrality Distribution.png

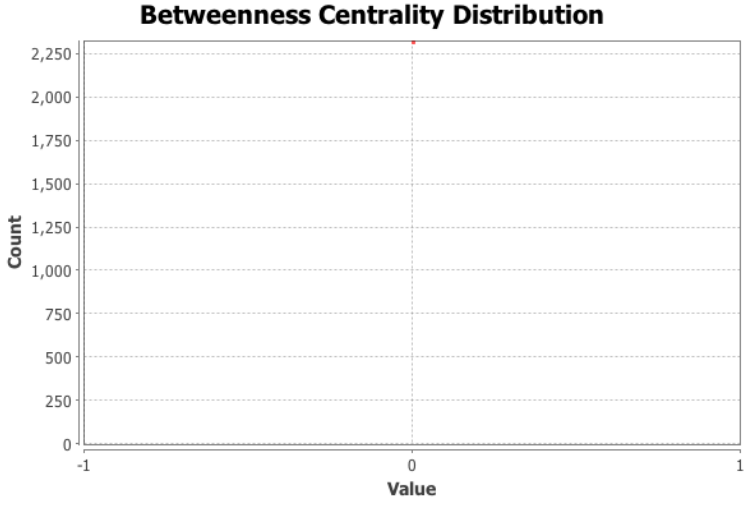

Figure 11: Network Diameter: Betweenness Centrality Distribution

Centrality Distribution.png

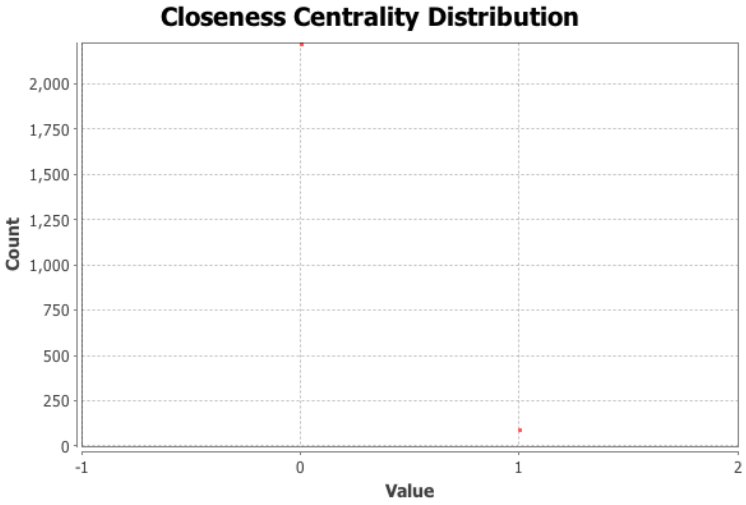

Figure 12: Network Diameter: Closeness Centrality Distribution

Distribution.png

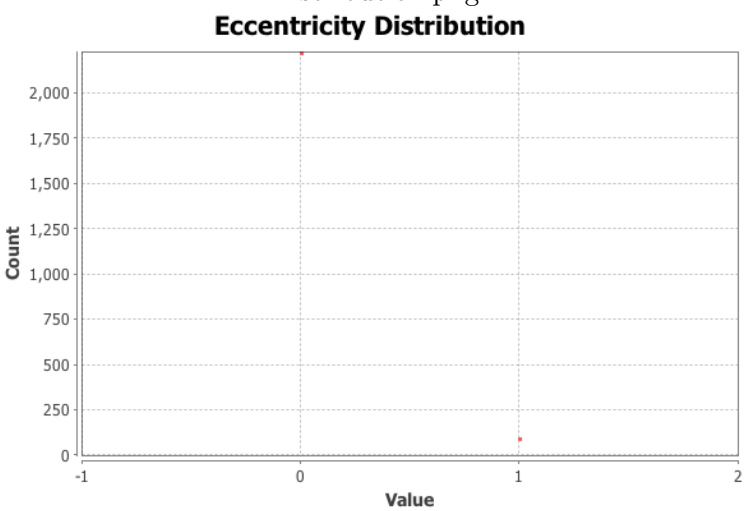

Figure 13: Network Diameter: Eccentricity Distribution

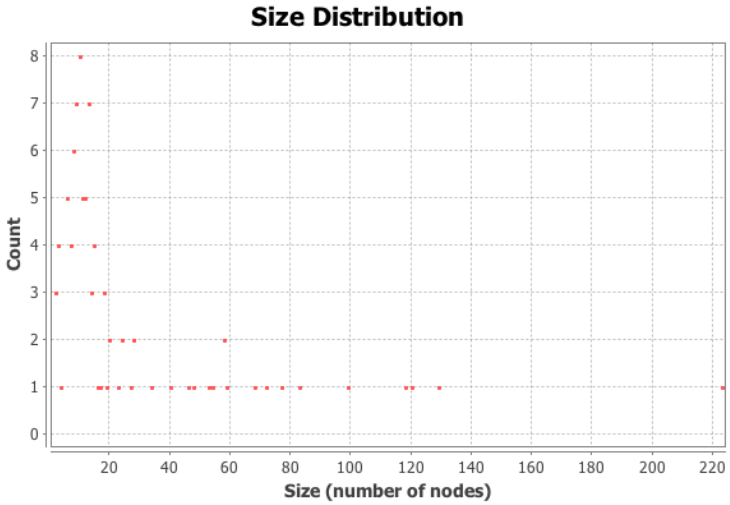

Figure 14: Network Diameter: Connected Component cc-size-distribution