CS 595: Assignment #4

Due on Thursday, October 9, 2014

Dr. Nelson 4:20pm

Holly Harkins

Contents

Problem 1

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

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).

Listing 1: Python 1000 Links

```
# -*- encoding: utf-8 -*-

import urllib
import urllib2
import re
import os
import sys
import md5
import BeautifulSoup
import subprocess
from subprocess import call

path="../100Links"

##New Dir for Links
try:
    os.makedirs(path)
except OSError:
    if os.path.exists(path):
        pass
```

Problem 1 continued on next page...

```
file1=open("FinalList.txt","r")
lineCount=0

for line in file1:
    lineCount=lineCount+1

    if lineCount<=100:
        Oneline=line.rstrip("\n")

        ##Beautiful Soup
        try:
            request = urllib2.Request(Oneline)
            response = urllib2.urlopen(request)
            soup = BeautifulSoup.BeautifulSoup(response)

        ##Hash
        test=md5.new(Oneline)
        hashfilename=test.hexdigest()

        internallinklist=[]
        del internallinklist[:]

        ##Inner Links
        for a in soup.findAll('a', attrs={'href': re.compile("^http://")}):
            link2=a['href']

            #No Duplicate Links
            if "png" not in link2 and "jpg" not in link2 and "#" not in link2 and "
                javascript" not in link2 and link2 not in internallinklist:
                internallinklist.append(link2)

        if not internallinklist:
            lineCount=lineCount-1

        else:
            with open(os.path.join(path,hashfilename), 'w') as file4:
                    file4.write("site:")
                file4.write("\n")
                file4.write(Oneline)
                file4.write("\n")
                file4.write("links:")
                file4.write("\n")

                for item in internallinklist:
                    file4.write(item)
                    file4.write("\n")
        sys.exit()

file1.close()
file4.close()
```

Problem 2

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

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.)

Listing 2: Python Memento

```
# -*- encoding: utf-8 -*-
import urllib
import urllib2
import re
import os
import sys
from collections import OrderedDict

path="../100links/"
URIs=dict()
lineCount=0

##Create Graphviz.dot file
graphviz=open("graphviz.dot", "w")
graphviz.write("digraph graphviz {\n")

##Loop all Files in Path
for l in os.listdir(path):

        x=0
        output=""
        output1=""
        output2=""
        site=""
```

```
        with open(path+str(l)) as Onefile:
            for line in Onefile:
                ##Site Name
                if x == 1:
                    site=line.strip()
                    if not site in URIs:
                       URIs[site]=lineCount
                       output1=str(URIs[site])
                       lineCount=lineCount+1
                ##Grab Links
                elif x >= 3:
                    links=line.strip()
                    if not links in URIs:
                        URIs[links]=lineCount
                        output2=str(URIs[links])
                        lineCount=lineCount+1

                        ##Final Output
                        if output1 and output2:
                            output=output+output1+"->"+output2+";"+"\n"
                x+=1

        ##Write to File
        graphviz.write(output)

##Create Ordered l
t=OrderedDict(sorted(URIs.items(), key=lambda t: t[1]))
for item in t.items():
    print str(item[1])+"[label="+item[0]+"]"
    graphviz.write(str(item[1])+"[label="+item[0]+"];\n")
graphviz.write("}")
del URIs

graphviz.close()
```

graphviz.dot file

Problem 3

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.

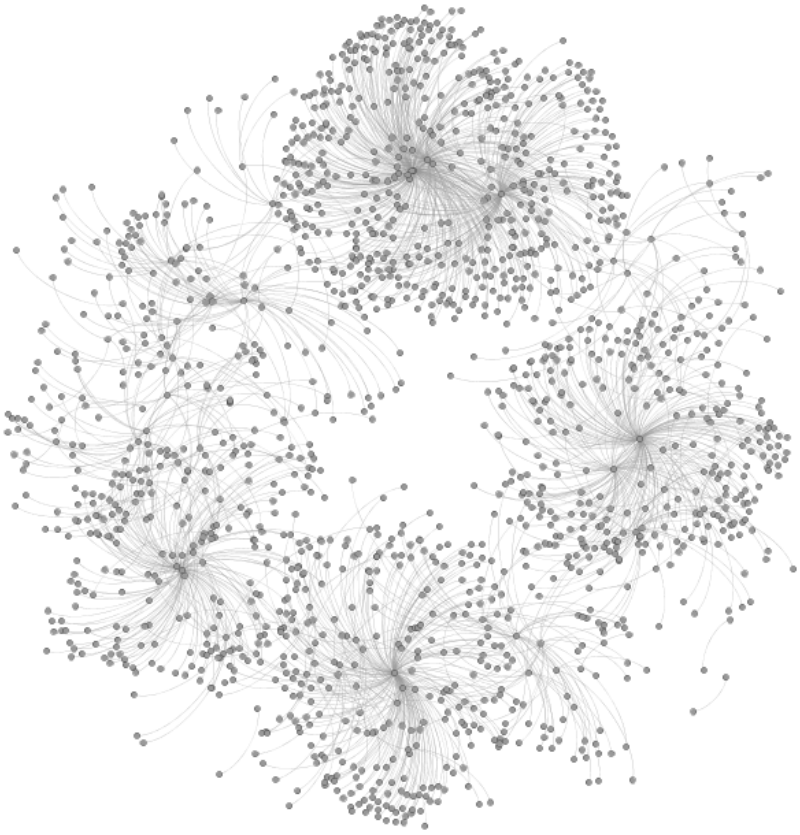

Figure 1: Visualized Graph

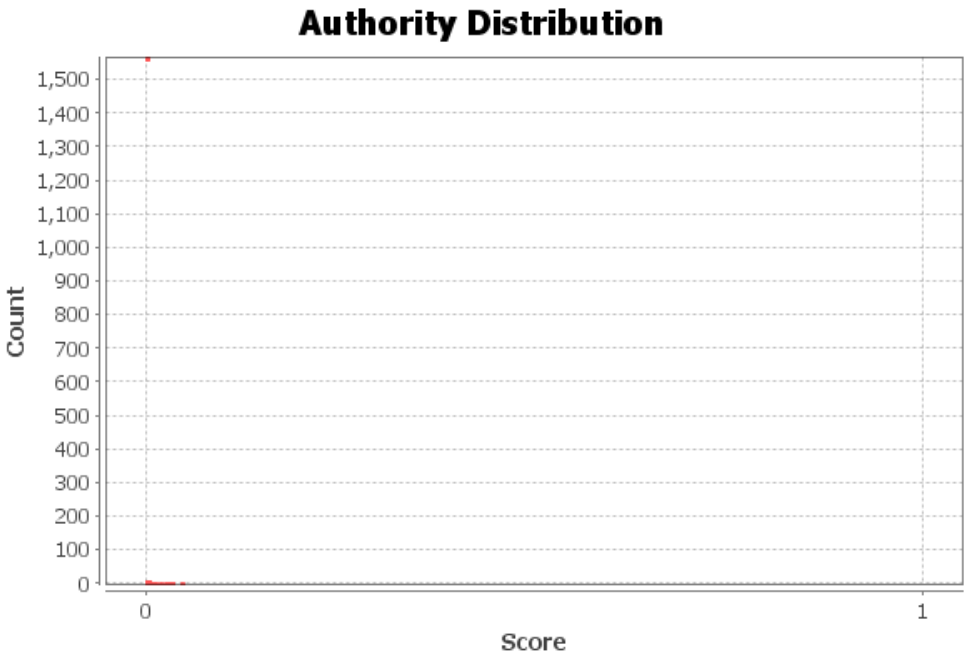

Figure 2: Hits

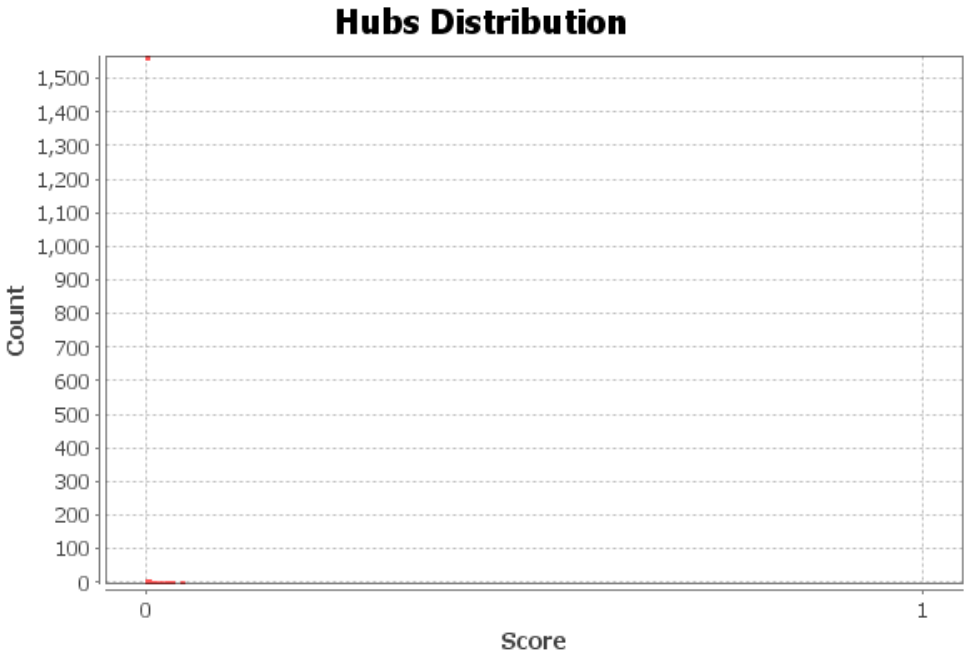

Figure 3: Hits

References

[1] http://www.graphviz.org/content/unix/

[2] http://www.graphviz.org/Gallery/directed/unix.gv.txt/

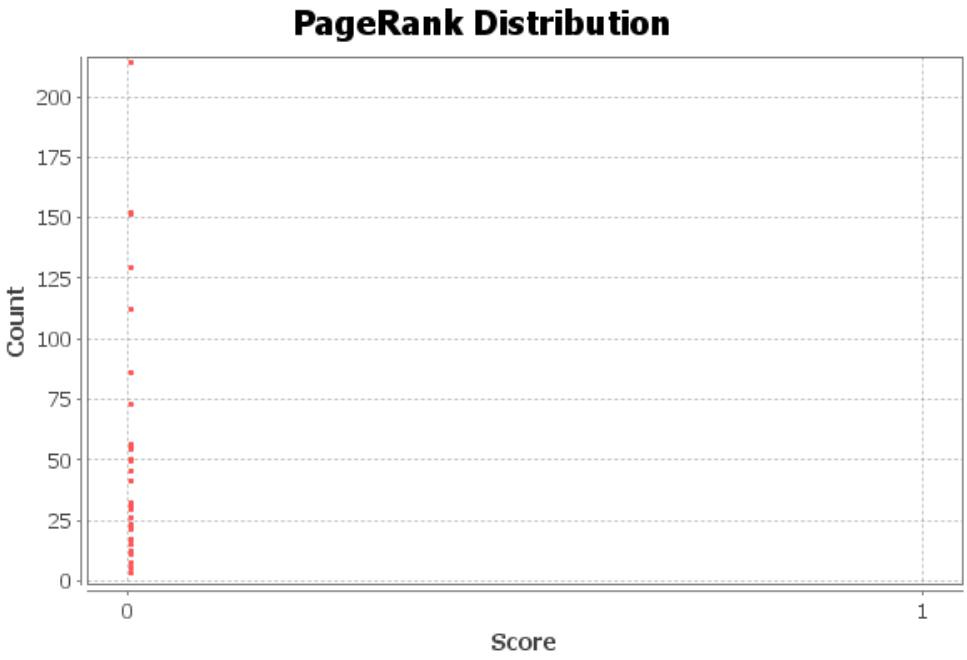

Figure 4: PageRank

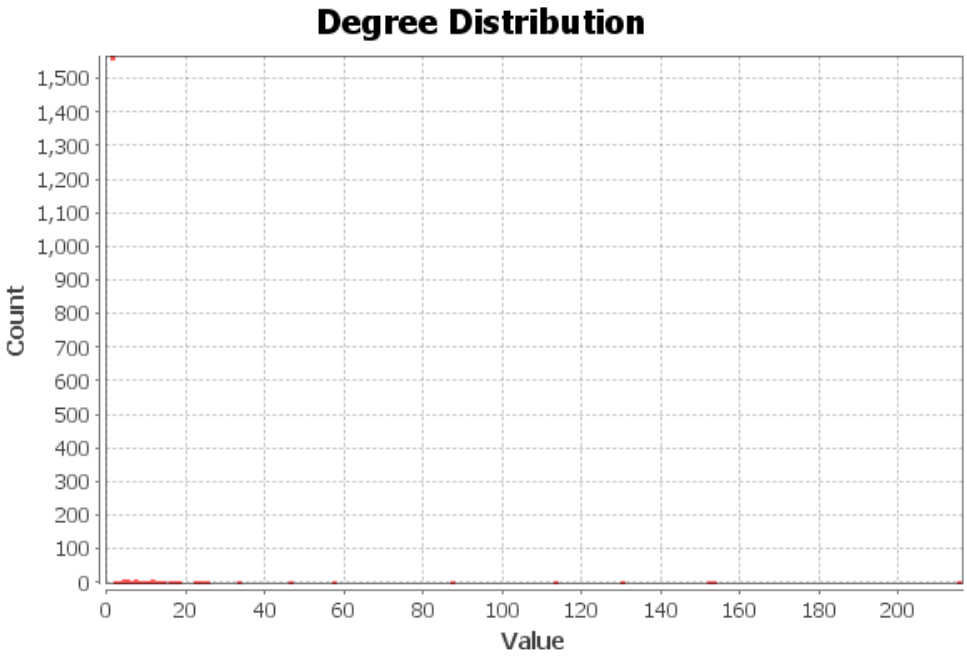

Figure 5: Average Degree

[3] http://www.graphviz.org/Documentation/dotguide.pdf/

[4] http://www.graphviz.org/content/dot-language/

[5] http://www.graphviz.org/Documentation.php/

Centrality Distribution.png

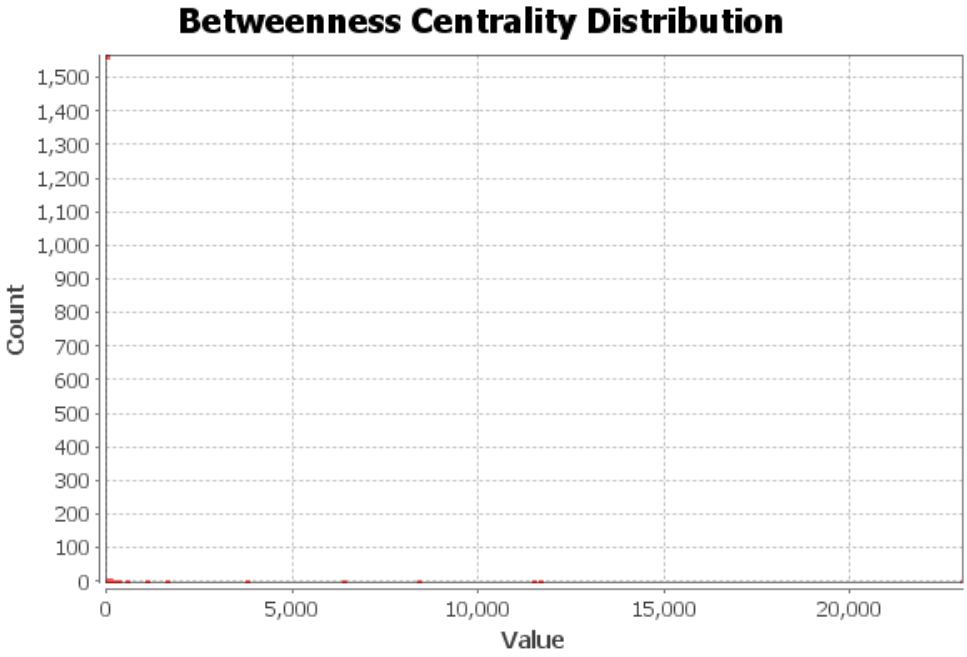

Figure 6: Network Diameter

Centrality Distribution.png

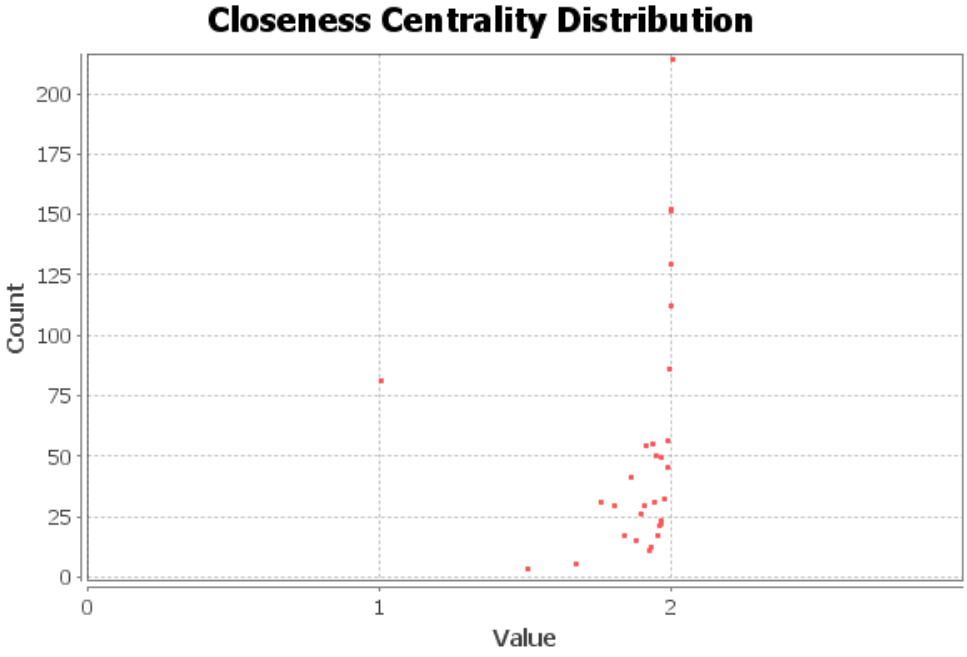

Figure 7: Network Diameter

[6] https://gephi.org/users/supported-graph-formats/graphviz-dot-format/

[7] https://gephi.org/users/supported-graph-formats/

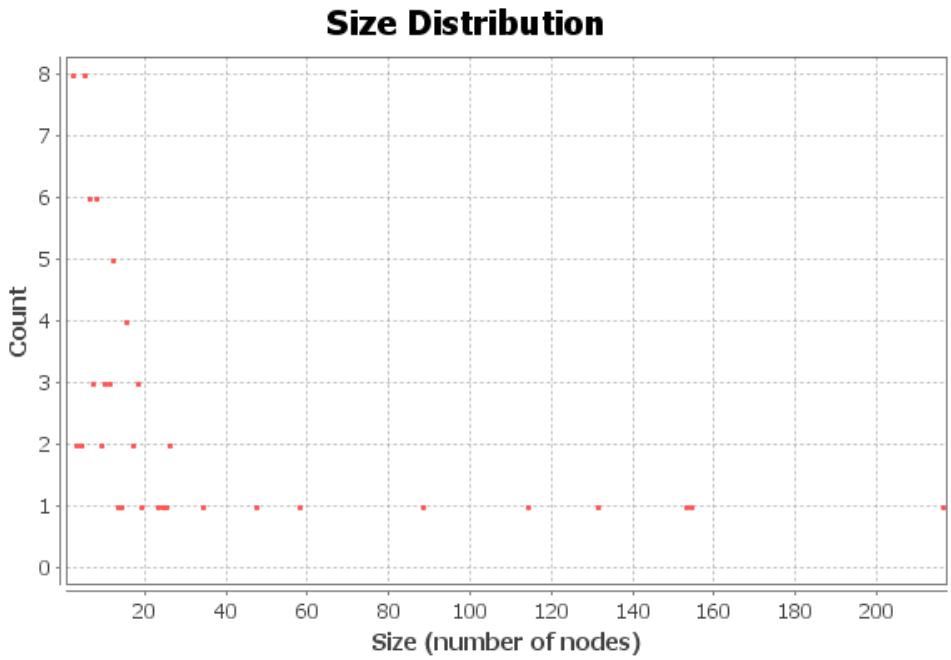

Figure 8: Connected Components

[8] http://www.youtube.com/watchvUrrWAt1rjc/