GEORGETOWN UNIVERSITY

Page Rank on AWS Cloud

ANLY502 - Final Report
Prof. Marck Vasiman

Team Name: Kungfu Four
***Team Member**: Jianing Sun, Yi Xiang, Yiran Liu, Wen Li*

Contents

1 Executive Summary

In order to learn the connection between websites, the PageRank algorithm is applied to raw data in this project. Pyspark is used as a tool for distributed computing of 50GB raw data files.

The main job here was to preprocess raw data, and turned it to “node.parquet” and “edge.parquet” being stored in S3 bucket for the use of PageRank algorithm. And then, the PageRank algorithm was implemented with a pipeline containing both Mapper and Reducer. Finally, we end up with a visualization to get some insights into the networks properties, trying different tools including Python, R Shiny, R, because PySpark was not already able to draw network graphs and displayed it on an html.

The main result from this project is a clean node & edge data file, pagerank scores for every website, and the visualized distribution of the network overall. From these results, we found that there are three websites with the same largest page rank total rank. In addition, some interesting findings catched our eyes, such as links with more Web Ads tend to have higher importance results which is agreed with our assumption that Web Ads is an efficient way to attract users’ attention and increase the website clicks.

2 Introduction

The Word Wide Web creates many new challenges for information retrieval with tremendous web page information online. Current estimations about this topic are more than 150 million web pages which could be doubled within a year. In addition, the complexity of information online ranging from a wide broad of topics makes the web page even more challengeable for information retrieval. Moreover, it is an essential ability for search engines contented with inexperienced users and pages engineered to manipulate search engine ranking functions.

However, the Word Wide Web is all about hypertext and offers auxiliary information on the page content of web URLs, such as hyperlink. The main goal of our project is implementing page rank algorithms on AWS cloud. We take advantage of the link structure of the Web to produce a global 'essential' ranking of every web page which is called PageRank. It is an important method to help users and search engines keep track of important information from the vast amount of the World Wide Web.

3 Method

3.1 Data Collection

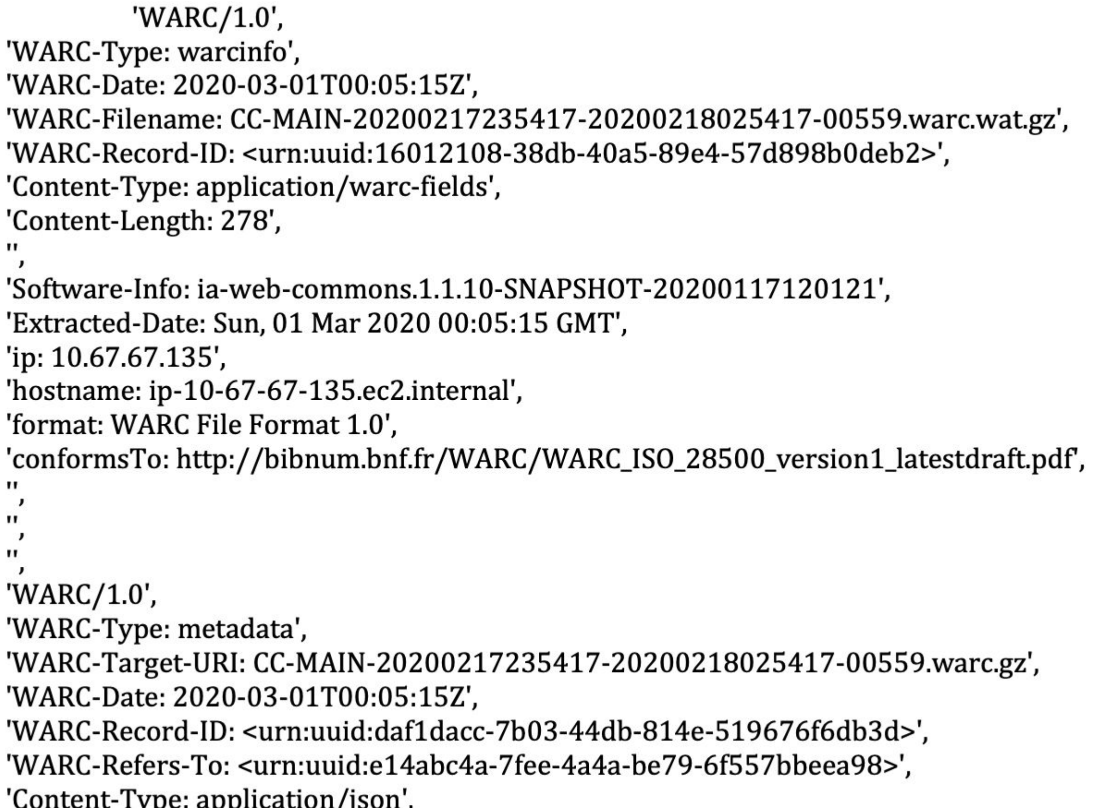
We extracted the latest raw data February 2020 crawl archive from the Common Crawl website which is a nonprofit organization that crawls the web and freely provides its archives and datasets to the public. To better explore and extract url links and hyperlinks in each web page, we collected the dataset in WARC segments files type. We download the dataset using S3 listed on our code file. Finally, we chose 50GB out of 180GB datafile in this S3 link.

3.2 Data Format

The WARC format is the raw data from the crawl, providing a direct mapping to the crawl process. Not only does the format store the HTTP response from the websites it contacts (WARC-Type: response), it also stores information about how that information was requested (WARC-Type: request) and metadata on the crawl process itself (WARC-Type: metadata).

For the HTTP responses themselves, the raw response is stored. This not only includes the response itself, what you would get if you downloaded the file, but also the HTTP header information, which can be used to glean a number of interesting insights.

Below is the JSON table example our data files



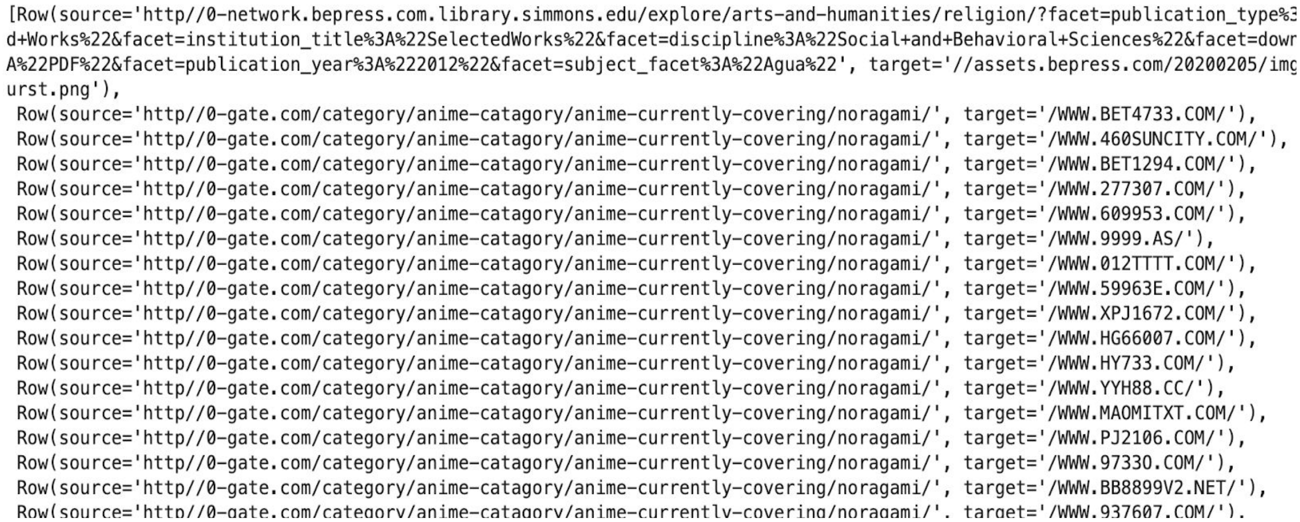
```
            'WARC/1.0',
'WARC-Type: warcinfo',
'WARC-Date: 2020-03-01T00:05:15Z',
'WARC-Filename: CC-MAIN-20200217235417-20200218025417-00559.warc.wat.gz',
'WARC-Record-ID: <urn:uuid:16012108-38db-40a5-89e4-57d898b0deb2>',
'Content-Type: application/warc-fields',
'Content-Length: 278',
'',
'Software-Info: ia-web-commons.1.1.10-SNAPSHOT-20200117120121',
'Extracted-Date: Sun, 01 Mar 2020 00:05:15 GMT',
'ip: 10.67.67.135',
'hostname: ip-10-67-67-135.ec2.internal',
'format: WARC File Format 1.0',
'conformsTo: http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf',
'',
'',
'',
'WARC/1.0',
'WARC-Type: metadata',
'WARC-Target-URI: CC-MAIN-20200217235417-20200218025417-00559.warc.gz',
'WARC-Date: 2020-03-01T00:05:15Z',
'WARC-Record-ID: <urn:uuid:daf1dacc-7b03-44db-814e-519676f6db3d>',
'WARC-Refers-To: <urn:uuid:e14abc4a-7fee-4a4a-be79-6f557bbeea98>',
'Content-Type: application/json',
```

Table 1 Data Example for WARC format

3.3 Data Preprocessing

3.3.1 Extract Web Page URLs and Hyperlink URLs

Using regular expression and JSON file manipulation, we extracted URL links and Hyperlinks for each web page and created graphs for these whole urls taking advantage of the JSON Format of Web Pages. In addition, in order to implement page rank algorithms, we need to preprocess our URLs regarding graph nodes and create edges between source URL and target URLs. Therefore, we created graph nodes using increasing index methods in Python and drew an edge between two nodes to identify the relationship between two URLs if there is any hyperlink in our current web page file. Below is the intermediate data example of edges files for extracted URLs with source and target.

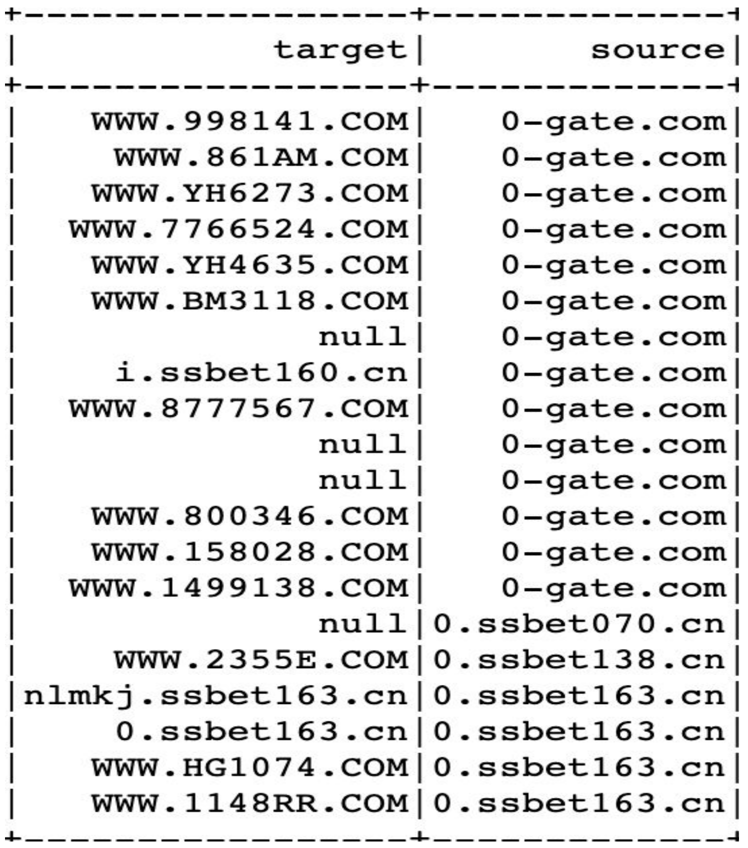


```
[Row(source='http//0-network.bepress.com.library.simmons.edu/explore/arts-and-humanities/religion/?facet=publication_type%3
d+Works%22&facet=institution_title%3A%22SelectedWorks%22&facet=discipline%3A%22Social+and+Behavioral+Sciences%22&facet=dow
A%22PDF%22&facet=publication_year%3A%222012%22&facet=subject_facet%3A%22Agua%22', target='//assets.bepress.com/20200205/img
urst.png'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.BET4733.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.460SUNCITY.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.BET1294.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.277307.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.609953.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.9999.AS/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.012TTTT.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.59963E.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.XPJ1672.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.HG66007.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.HY733.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.YYH88.CC/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.MAOMITXT.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.PJ2106.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.97330.COM/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.BB8899V2.NET/'),
 Row(source='http//0-gate.com/category/anime-catagory/anime-currently-covering/noragami/', target='/WWW.937607.COM/'),
```

Table 2 Data Example for original URLs

3.3.2 Extract domain URLs from general URLs

After extracting URLs and Hyperlinks of web pages, we found that the original URLs dataset are messy and inconsistent. Therefore, we decided to extract domain URLs from original urls using regular expressions in Python. Finally, aftering deleting Null values or values without suitable format, our whole dataset includes 20, 000 nodes and edges. Below is the example of graph dataset with URLs links and graph node value in our page rank algorithm. Below is the intermediate data example of domain URLs.



```
+-----------------+-------------+
|           target|       source|
+-----------------+-------------+
|    WWW.998141.COM|   0-gate.com|
|     WWW.861AM.COM|   0-gate.com|
|    WWW.YH6273.COM|   0-gate.com|
|   WWW.7766524.COM|   0-gate.com|
|    WWW.YH4635.COM|   0-gate.com|
|    WWW.BM3118.COM|   0-gate.com|
|              null|   0-gate.com|
|     i.ssbet160.cn|   0-gate.com|
|   WWW.8777567.COM|   0-gate.com|
|              null|   0-gate.com|
|              null|   0-gate.com|
|    WWW.800346.COM|   0-gate.com|
|    WWW.158028.COM|   0-gate.com|
|   WWW.1499138.COM|   0-gate.com|
|              null|0.ssbet070.cn|
|     WWW.2355E.COM|0.ssbet138.cn|
|nlmkj.ssbet163.cn|0.ssbet163.cn|
|     0.ssbet163.cn|0.ssbet163.cn|
|    WWW.HG1074.COM|0.ssbet163.cn|
|    WWW.1148RR.COM|0.ssbet163.cn|
+-----------------+-------------+
```

Table 3 Data Example for domain URLs

3.4 Hypothesis

The classic hypothesis of implementing page rank in our data is 'Power-law hypothesis' which refers in a scale-free network the PageRank scores follow a power law with the same exponent as indegress. In the Power-law behavior of PageRank, it is obtained in the directed configuration model with independent in and out degrees. This result follows crucially from the coupling of the graph with a branching tree. However, formalizing and proving the Power-law hypothesis mathematically turns out to be challenging for many Mathematicians. Therefore, we decided not to prove this hypothesis in our project. Also, we hypothesized that there is an equal weight for each hyperlink appearing in our web page data which is the fundamental assumption for our page rank algorithm.

3.5 Model

PageRank algorithm works by counting the number of links to a page to determine an estimation of how important the website is. The assumption is that more important websites are likely to receive more links from other websites. To calculate page rank of our dataset, we implement it in two steps, mapping and reducing. In the mapper, we calculate the page rank for each edge/link. In the reducer, we aggregate the page rank of each edge by the same website and generate a list of websites ordering by total page rank value, which indicates the importance of each website. Mapper and Reducer is an efficient tool for PageRank calculation, it breaks down the whole calculation into steps and makes the process easy to implement and interpret.

Mapper

We see each website as a node in PageRank. In the mapper, we first assign the initial value to each distinct node regardless of the relationship among them. We assume that the total initial value of all distinct nodes is 1. Each distinct node should have an equal initial value, which is 1 divided by the number of distinct nodes. The new column called initialValue is added to the dataset, each record has the same initial value here, and it is used to calculate PageRank in the following steps.

After that, we sort the dataset by source node and calculate the PageRank for each single edge or each record. In the dataset, each record represents one source node links to another target node. One source node may have multiple edges linking to different target nodes, within the same source node, the PageRank is equal to the initial value of the source node divided by the number of target nodes. For example, source node 401 has initial value 1/3, it has 4 target nodes, then the page rank of each target node is 1/12. We add a new column pageRank to the dataset to record the page rank of each single edge.

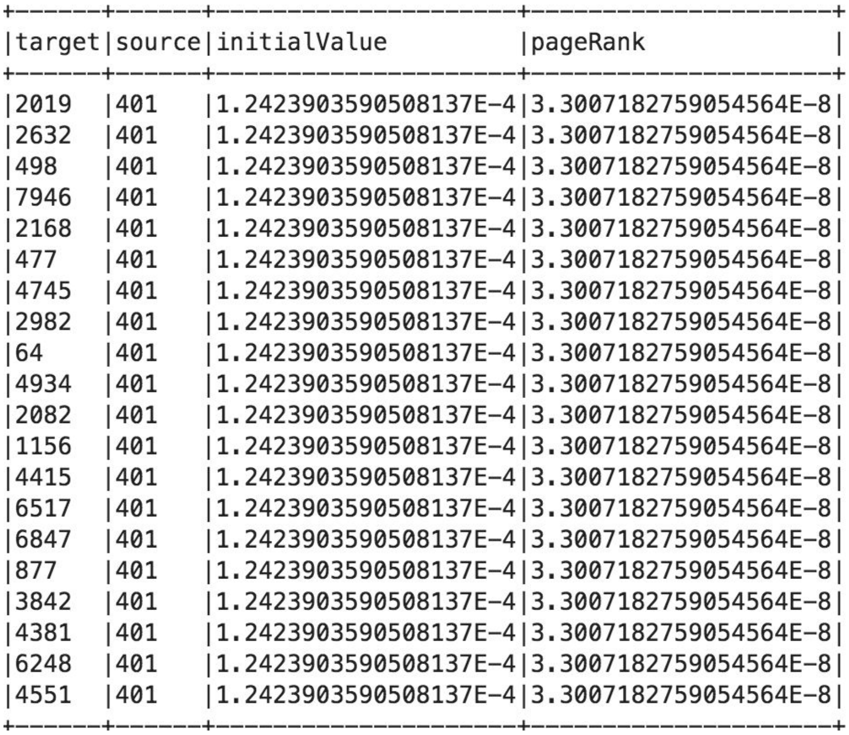
Reducer

In the reducer, we sort the dataset by target nodes and aggregate pageRank by the same target node as pageRankTotal, as a new column added to the dataset. Each target node has one pageRankTotal value, and we sort it in descending order. As a result, we get a list of websites ordering by the page rank total value. The higher page rank total is, the more important the website is.

4 Result

4.1 Mapper

In the mapper, the sample of results is shown as the table underneath. In the dataset, there are 8049 distinct nodes, the initial value of each node is about 0.000124. Next, we calculate the page rank for each edge, using the equation mentioned in the prior section, the result is shown in the pageRank column. In this table, it shows the pageRank of each target node linking from source node 401, the result is about 3.3E-8. Target nodes linked from other source nodes may have different values of page rank, depending on the number of target nodes link from the same source node.

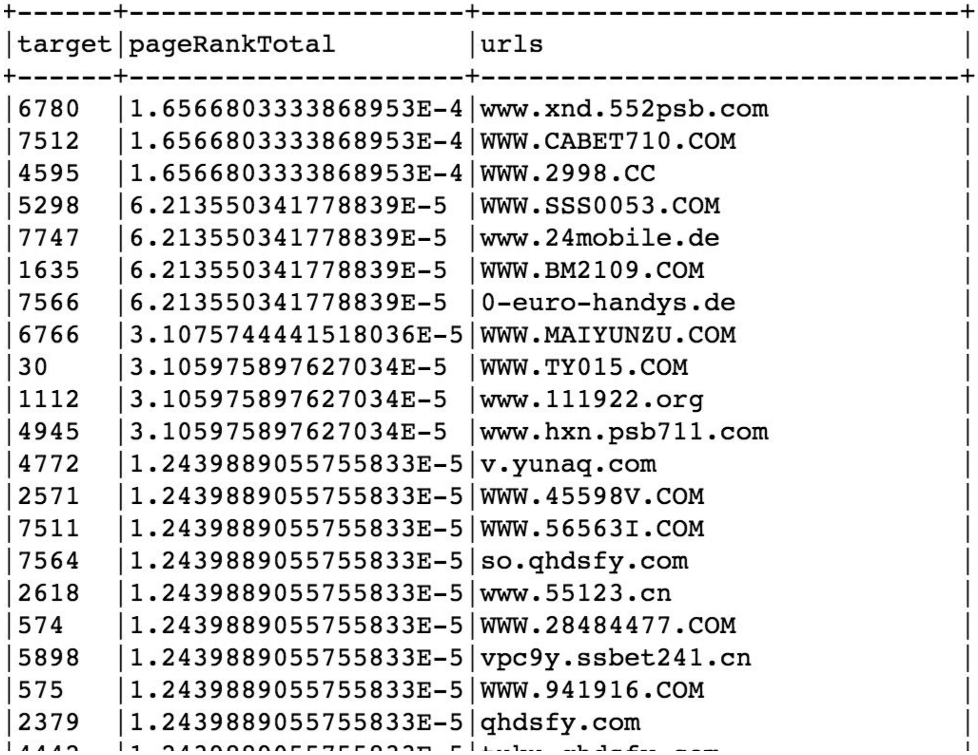


```
+------+------+---------------------+---------------------+
|target|source|initialValue         |pageRank             |
+------+------+---------------------+---------------------+
|2019  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|2632  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|498   |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|7946  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|2168  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|477   |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|4745  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|2982  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|64    |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|4934  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|2082  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|1156  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|4415  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|6517  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|6847  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|877   |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|3842  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|4381  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|6248  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
|4551  |401   |1.2423903590508137E-4|3.3007182759054564E-8|
+------+------+---------------------+---------------------+
```

Table 4 Data Example for Mapper Results

4.2 Reducer

In the reducer, after aggregating page rank of single edge, we finally get the result of a list of websites sorting by page rank total value in descending order, as it is shown in the table below. The largest page rank total is about 0.0000166. There are three websites with the same largest page rank total rank, and these three websites are the most important websites in the dataset.



```
+------+---------------------+------------------------------+
|target|pageRankTotal        |urls                          |
+------+---------------------+------------------------------+
|6780  |1.6566803333868953E-4|www.xnd.552psb.com            |
|7512  |1.6566803333868953E-4|WWW.CABET710.COM              |
|4595  |1.6566803333868953E-4|WWW.2998.CC                   |
|5298  |6.213550341778839E-5 |WWW.SSS0053.COM               |
|7747  |6.213550341778839E-5 |www.24mobile.de               |
|1635  |6.213550341778839E-5 |WWW.BM2109.COM                |
|7566  |6.213550341778839E-5 |0-euro-handys.de              |
|6766  |3.1075744441518036E-5|WWW.MAIYUNZU.COM              |
|30    |3.105975897627034E-5 |WWW.TY015.COM                 |
|1112  |3.105975897627034E-5 |www.111922.org                |
|4945  |3.105975897627034E-5 |www.hxn.psb711.com            |
|4772  |1.2439889055755833E-5|v.yunaq.com                   |
|2571  |1.2439889055755833E-5|WWW.45598V.COM                |
|7511  |1.2439889055755833E-5|WWW.56563I.COM                |
|7564  |1.2439889055755833E-5|so.qhdsfy.com                 |
|2618  |1.2439889055755833E-5|www.55123.cn                  |
|574   |1.2439889055755833E-5|WWW.28484477.COM              |
|5898  |1.2439889055755833E-5|vpc9y.ssbet241.cn             |
|575   |1.2439889055755833E-5|WWW.941916.COM                |
|2379  |1.2439889055755833E-5|qhdsfy.com                    |
|4442  |1.2439889055755833E-5|tuku.qhdsfy.com               |
```

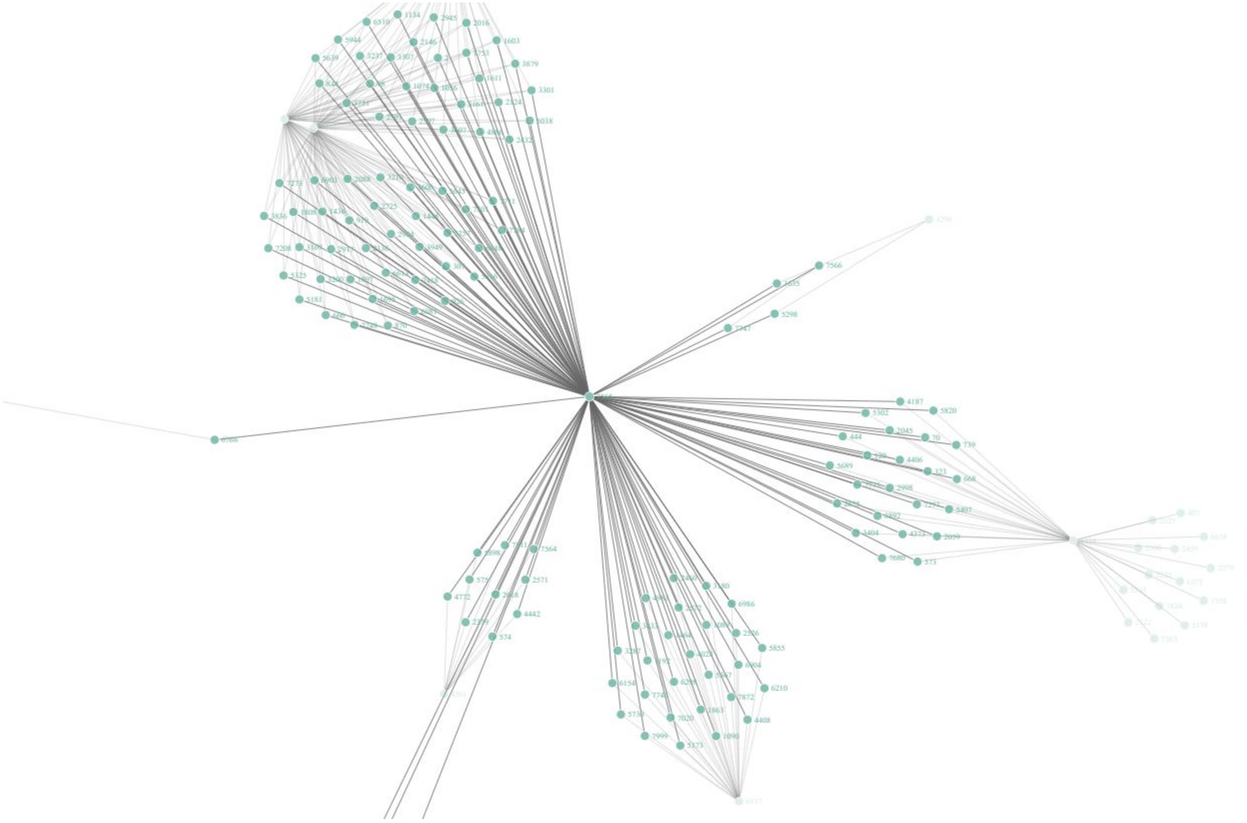
Table 5 Data Example for Reducer Results

There are four websites with the second largest page rank total. As the page rank total value decreases, the number of websites have the same rank increasing. In this way, less important websites compete with tons of other websites with the same rank, and it is hard to be clicked and reviewed by users.

There is another interesting finding in the result. By checking some websites ranking in the top of the results, certain of them are small online games, like the game ads you see when you check your Facebook. We can have an assumption that the Web Ads is an efficient way to attract users' attention and increase the website clicks.

4.3 Visualization

After the PageRank algorithm was applied, the visualization of the network graph was displayed with R codes, showing the network's inner structure. (please see the attached html file for more details)



Our findings:

- Each local network community has high centrality
- These local network communities are connected by some important dots (web sites) with high betweenness
- Some local communities have few connections to the other network communities

5 Future Work

In this project, we mainly focus on the domain of the website as the source nodes and target nodes. In the future project, we could expand the dataset by including the subdomain of the websites, since the same domain with different subdomains are considered as different websites.

As we construct more detailed and sophisticated page rank models, we could extend the study field to the important features of high page rank websites, using the Machine Learning techniques to study the key features of high page rank websites. As the result of study, we can provide customers suggestions on the way of improving the quality and related content of websites and increasing website clicks.

6 References

1. The PageRank Citation Ranking: Bringing Order to the Web, Stanford University, *http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf*
2. Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Ecient crawling through url ordering. In To Appear: Proceedings of the Seventh International Web Conference, (WWW 98), 1998.
3. Power-Law Hypothesis for PageRank, Nelly Litvak, University of Twente and Eindhoven University of Technology, Netherlands
4. Geeks for Geeks *https://www.geeksforgeeks.org/page-rank-algorithm-implementation/*
5. Graph Tutorial Pyspark *https://pysparktutorial.blogspot.com/2017/10/graphframes-pyspark.html*

Code Files

GitHub Repo: *https://github.com/JianingSun-js4770/Kungfu-4--pagerank.git*

Data Preprocess Files:

- DataScraper.ipynb (Data Collection)

- BuildGraph.ipynb (Build Graph and Edges for URLs)
- ExtractDomain.ipynb (Extract Domain URLs from Original Query URLs)

Mapper and Reducer File:

- MapperAndReducer.ipynb

Other Files:

- README.md
- instance-metadata.json
- network_viz.html

Division of Work

Wen Li and Yi Xiang: Worked on Data Collection and Data Preprocessing
Yiran Liu: Worked on mapper and reducer tasks
Jianing Sun: Worked on result visualization and interpretation