

Exploring the .gov Domain from 2002 to 2005:
InDegree and OutDegree Analyses

Jacob Bank (jeb369)

May 7, 2009

Contents

0.1	Abstract	2
0.2	Introduction	3
0.3	Description of Influential Pages Across Crawls	4
0.4	InDegree and OutDegree Distributions	7
0.4.1	OutDegree Distributions with Outliers Removed	7
0.4.2	InDegree Distributions	11
0.4.3	Curve Fitting	14
0.5	Conclusions and Future Work	16
0.6	Acknowledgements	17
0.7	Appendix: Source Code	19
0.7.1	Power Law Exponent Calculation	19
0.7.2	InDegree and OutDegree Calculation	20

0.1 Abstract

The goal of the project as a whole was to explore the structure of the .gov domain and its evolution over time. The group worked with crawls provided by the Internet Archive[1] from the years 2002, 2003, 2004, and 2005. This specific part of the project analyzed the InDegree and OutDegree distributions of pages in the .gov domain in each of the four crawls.

This exploration had a few major goals. First, the group wanted to determine which pages in the .gov domain were the most influential in the web graph. InDegree provides a simple and easy to calculate measure of power in such a network. Second, by comparing data across the four crawls, it became possible to see how pages rise and fall in influence over time. In the context of the .gov domain, this type of study is particularly interesting because the structure of the web graph tends to reflect the importance of current events.

The distributions of the InDegree and OutDegree rank-frequency plots were also quite interesting to analyze. The group hypothesized that these distributions could be fit to either a power law or logarithmic curve. After exploring the graphs, the power law distribution did seem to be the best fit, and power law exponents were estimated for the distributions using the maximum likelihood exponent technique[2]. This analysis of the distributions also led to the discovery of a few interesting anomalies that led to conclusions about a certain type of page that is characterized by a huge OutDegree relative to its InDegree. The InDegree and OutDegree calculations, as well as the power law estimates were done using MapReduce[4] programming on the Cornell Center for Advanced Computing's Hadoop[3] Cluster.

0.2 Introduction

The purpose of the Web Laboratory is to provide data and computing tools for research about the Web and information on the Web. One of the teams this semester worked on analyzing the structure and evolution of the .gov domain. The team analyzed data taken from four crawls of the .gov domain, performed by the Internet Archive [1]. The crawls, titled DJ, DP, DV, and EB, came from the years 2002, 2003, 2004, and 2005 respectively. This document describes the findings of this exploration that focused on the InDegree and OutDegree distributions. There were two main goals to this segment of the project. The first was to use InDegree and OutDegree rankings to find the most influential pages in the .gov domain as a whole. The second was to analyze the shape of the InDegree and OutDegree distributions as a whole. InDegree for a page is defined as the number of hyperlinks that point to it, and OutDegree is similarly defined as the number of hyperlinks that point out from a page. The curve-fitting of the distributions to Power Law curves was done using the Maximum Likelihood Estimate of Exponents Method [2].

Due to the scale of the data, the Map Reduce programming paradigm running on the Web Lab's Hadoop cluster was used to perform most of the computations. Hadoop [3] is an open source framework that provides both reliability and data motion for applications. It implements the Map Reduce [4] model, in which an application's work load is divided into many smaller pieces and distributed over the a cluster's nodes. Hadoop also provides a reliable distributed file system (HDFS) that stores the data across the nodes. This in turn allows for extremely high aggregate bandwidth over the entire cluster. The Hadoop cluster allowed analyses on large datasets (millions of points) to be performed easily.

0.3 Description of Influential Pages Across Crawls

Overview

Pages that have high InDegree tend to be influential and often-visited nodes of web graphs. The following list shows the top fifteen pages in terms of InDegree for each of the four crawls. Pages not in the .gov domain and pages in subdomains belonging to specific states have been removed. Those not in the .gov domain have been removed because they are not relevant to this study of influential pages in the domain. Pages belonging to specific states have been removed because the structure of these sites change greatly over the four crawls as states began to establish their own domains. This inconsistency across crawls is the reason for the removal of these pages. These InDegree and OutDegree numbers were calculated using a Hadoop MapReduce program (source code in the appendix). The data itself was drawn from the Internet Archive's crawls, which is stored in the Web Lab's database.

DJ

Rank	InDegree	URL
1	36663	www.fcc.gov/fcc-bin/htimage/pub/www/pub/opa.map
2	36327	www.fcc.gov/mmb/asd/decdoc/intro.html
3	36324	www.fcc.gov/mmb/asd/welcomeALT.html
4	36321	www.fcc.gov/mmb/asd/main/am.html
5	35655	images.jsc.nasa.gov/feedback/
6	29411	www.usda.gov/
7	17471	www.ncbi.nlm.nih.gov/HomoloGene
8	17320	www.ncbi.nlm.nih.gov/UniGene/At.Home.html
9	17319	www.ncbi.nlm.nih.gov/UniGene/query.cgi
10	17319	www.ncbi.nlm.nih.gov/UniGene/Os.Home.html
11	17319	www.ncbi.nlm.nih.gov/UniGene/Dr.Home.html
12	17318	www.ncbi.nlm.nih.gov/genome/guide/human/index.html
13	16433	www.npwrc.usgs.gov/sitemap.htm
14	16016	www.cdc.gov/search.htm
15	14771	democrats.senate.gov/calendar/

DP

Rank	InDegree	URL
1	153187	www.noaa.gov/
2	131795	www.epa.gov/
3	125607	www.usgs.gov/
4	117738	www.dol.gov/
5	99929	www.firstgov.gov/
6	97529	www.nasa.gov/
7	90286	www.epa.gov/epafiles/usenotice.htm
8	89411	www.epa.gov/cgi-bin/epaprintonly.cgi
9	88723	www.nws.noaa.gov/disclaimer.html
10	86424	www.hhs.gov/
11	82340	www.nih.gov/
12	77997	www.epa.gov/epafiles/scripts/dateurl.js
13	77421	www.dol.gov/dol/privacynotice.htm
14	76598	www.nws.noaa.gov/
15	73549	www.ed.gov/

DV

Rank	InDegree	URL
1	286405	www.firstgov.gov/
2	217012	www.noaa.gov/
3	164538	www.nasa.gov/
4	145854	www.hhs.gov/
5	142619	www.epa.gov/
6	141710	www.usgs.gov/
7	133424	www.nws.noaa.gov/disclaimer.html
8	121047	www.dol.gov/
9	112790	www.nws.noaa.gov/
10	109717	www.epa.gov/epafiles/usenotice.htm
11	104466	www.whitehouse.gov/
12	100525	www.epa.gov/cgi-bin/epaprintonly.cgi
13	96457	www.doi.gov/
14	95493	www.usda.gov/
15	83603	www.dhs.gov/

EB

Rank	InDegree	URL
1	240823	www.ncbi.nlm.nih.gov/COG/new/release/COGhelp.html
2	164171	www.firstgov.gov/
3	99288	www.noaa.gov/
4	90627	www.hhs.gov/
5	82143	www.nih.gov/
6	73352	www.dol.gov/
7	68173	www.census.gov/
8	65755	www.census.gov/main/www/subjects.html
9	65752	www.census.gov/main/www/access.html
10	65747	www.census.gov/main/www/srchtool.html
11	65631	www.census.gov/main/www/contacts.html
12	64835	www.census.gov/index.html
13	63691	www.census.gov/main/www/cen2000.html
14	62938	www.census.gov/main/www/policies.html
15	62356	www.loc.gov/

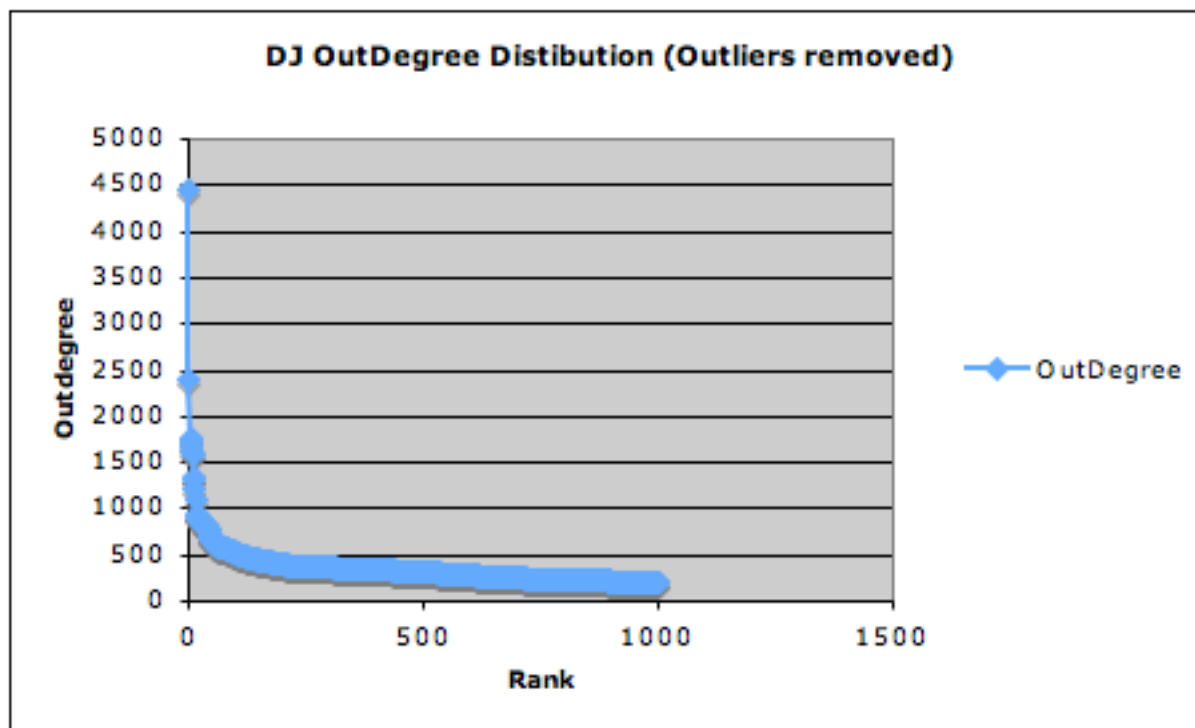
Observations

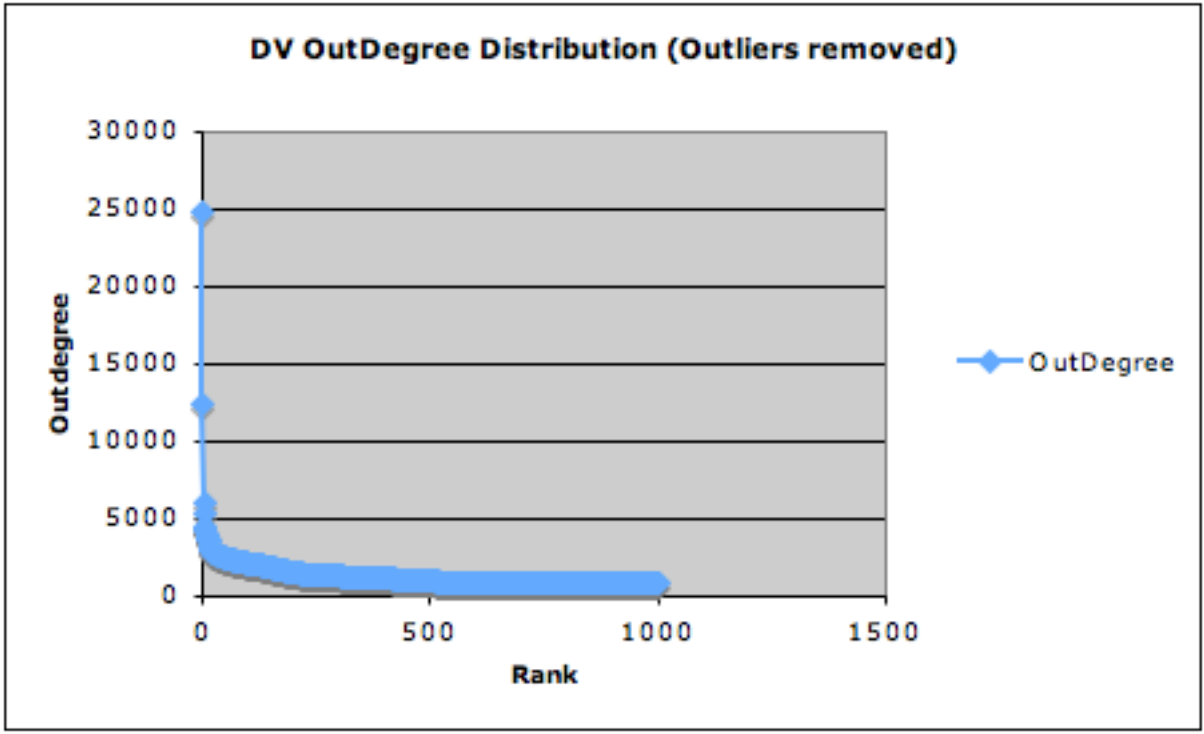
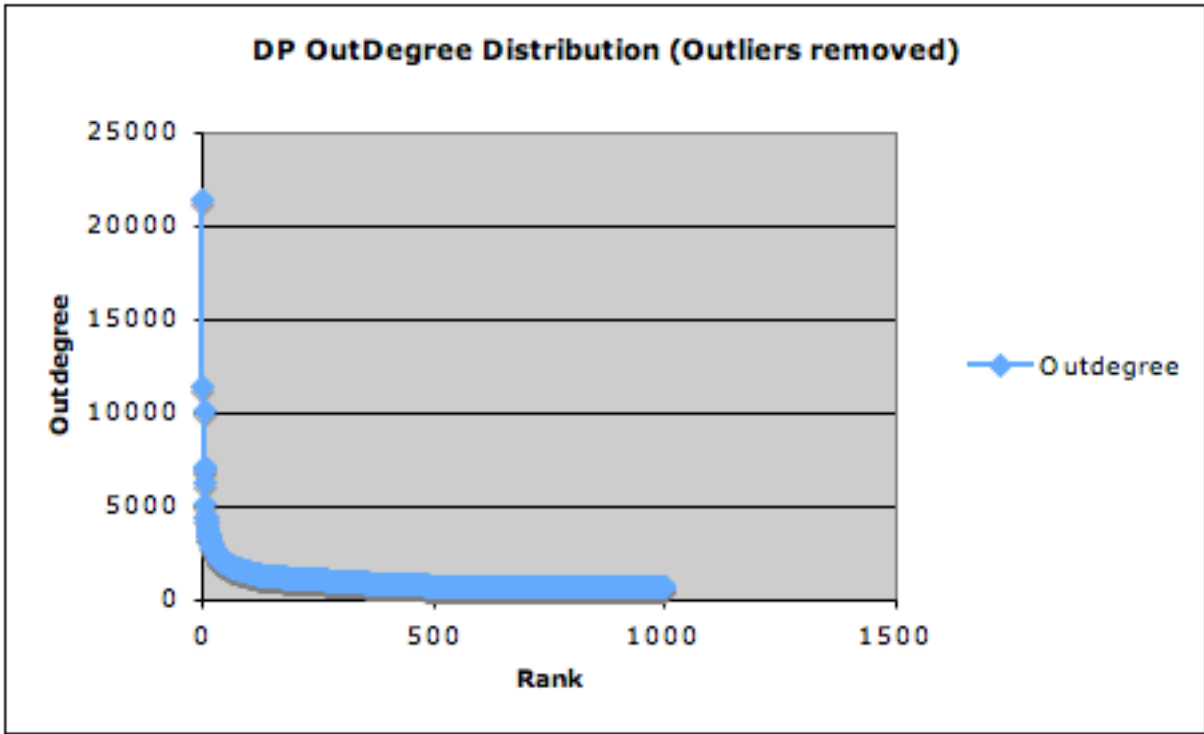
The top 15 InDegrees in the DJ crawl are dominated by pages from the fcc and nih subdomains. Beginning in DP, then through DV and EB, we see the home pages of the largest subdomains beginning to dominate the InDegree counts. The largest subdomain across all four crawls was NOAA, so it is no surprise to see www.noaa.gov ranked first in DP, second in DV, and third in EB. Similarly, NASA, the second largest subdomain, has www.nasa.gov ranked sixth in DP and third in DV. Other results are a bit more interesting. www.firstgov.gov is ranked fifth in DP, first in DV, and second in EB despite [firstgov.gov](http://www.firstgov.gov/) being only the ninety-eighth largest subdomain. In EB, we see pages from [census.gov](http://www.census.gov/) in ranks seven through 15, which may indicate an increased interest in census data. These rankings also can lead to interesting conclusions about how current events change the structure of the web domain. For example, whitehouse.gov only appears in one of the above rankings, 2004, which happened to be the year of a presidential election. Sites in the epa.gov subdomain featured prominently in the rankings of DP and DV, but not in the rankings of DJ and EB. By performing large scale analyses of this sort, one can observe how issues become more and less important to the government based on current events.

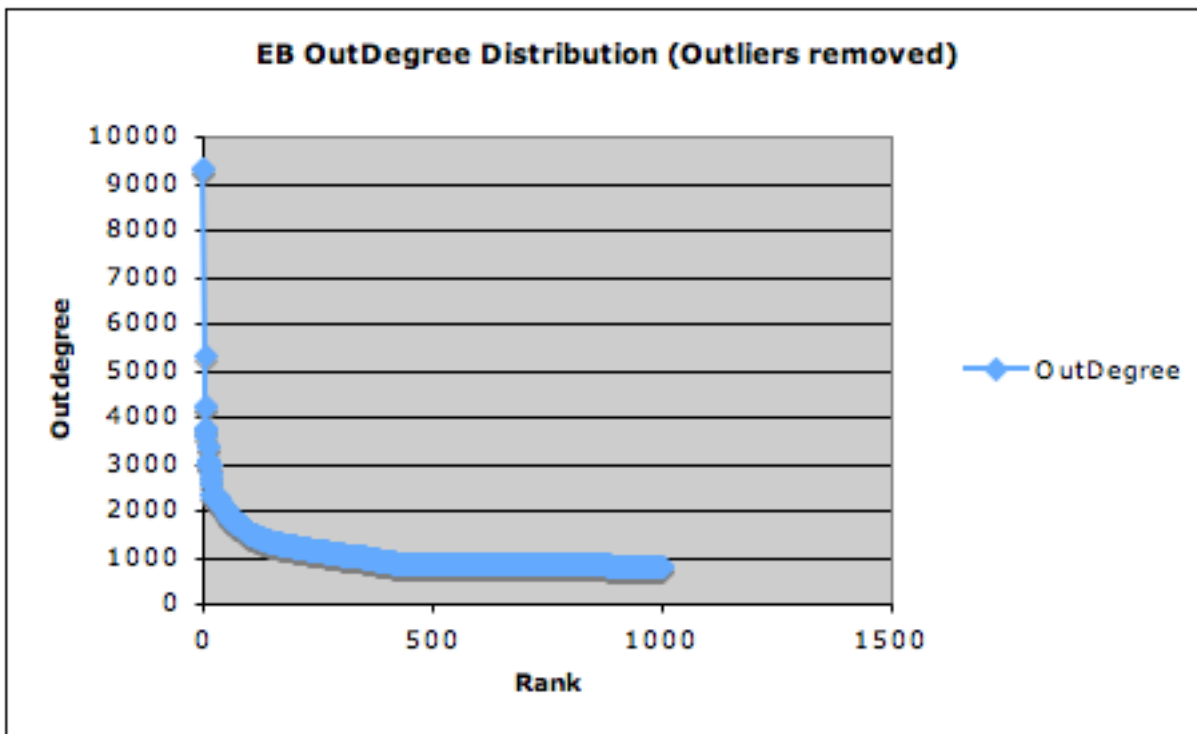
0.4 InDegree and OutDegree Distributions

0.4.1 OutDegree Distributions with Outliers Removed

The following graphs are the OutDegree distributions of the top 1,000 pages in terms of OutDegree in each of the four crawls. The plots below are rank-frequency plots, with the rank from 1 to 1,000 appearing on the x-axis, and the frequency (OutDegree) appearing on the y-axis. Outliers have been removed by a method to be explained later in this section.







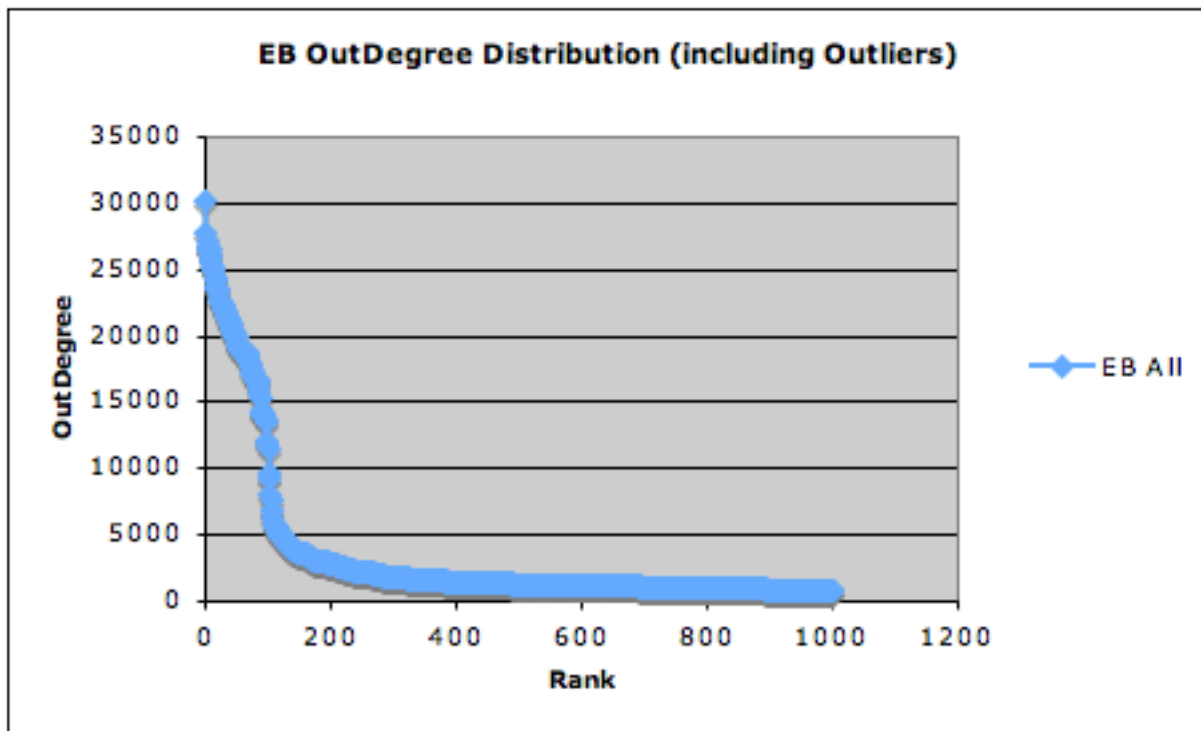
Observations

The first interesting point to note about these distributions is the fluctuation of the highest OutDegree number. From DJ to DP, there is a big jump in OutDegree of the highest ranked page, from about 4,500 to over 20,000. Then from DP to DV, there is a small jump up to about 25,000. Then, from DV to EB, the highest OutDegree actually drops back down to about 9,000. The huge jump from DJ to DP is expected, due to the much larger size of the DP crawl. The small increase from DP to DV was expected as well. The drop to EB, however, is quite surprising given the fact that the .gov domain did grow every year from 2002 to 2005.

The next noteworthy feature of the graphs is the OutDegree at which each distribution levels off. These figures are hard to determine from looking at the graph, but from the raw data we can see that DJ levels off at around 400, DP at around 800, DV at around 900, and EB at around 900 as well. Though the highest ranked page in EB had a much smaller InDegree than that of DV, the distributions actually began to level off at about the same point. This seems more intuitively correct, but it is still quite interesting that EB does not level off at a higher point than DV. This deviation from the pattern of DJ, DP, and DV could either mean that the EB crawl was conducted differently, or that the structure of the .gov domain did change significantly from 2004 to 2005.

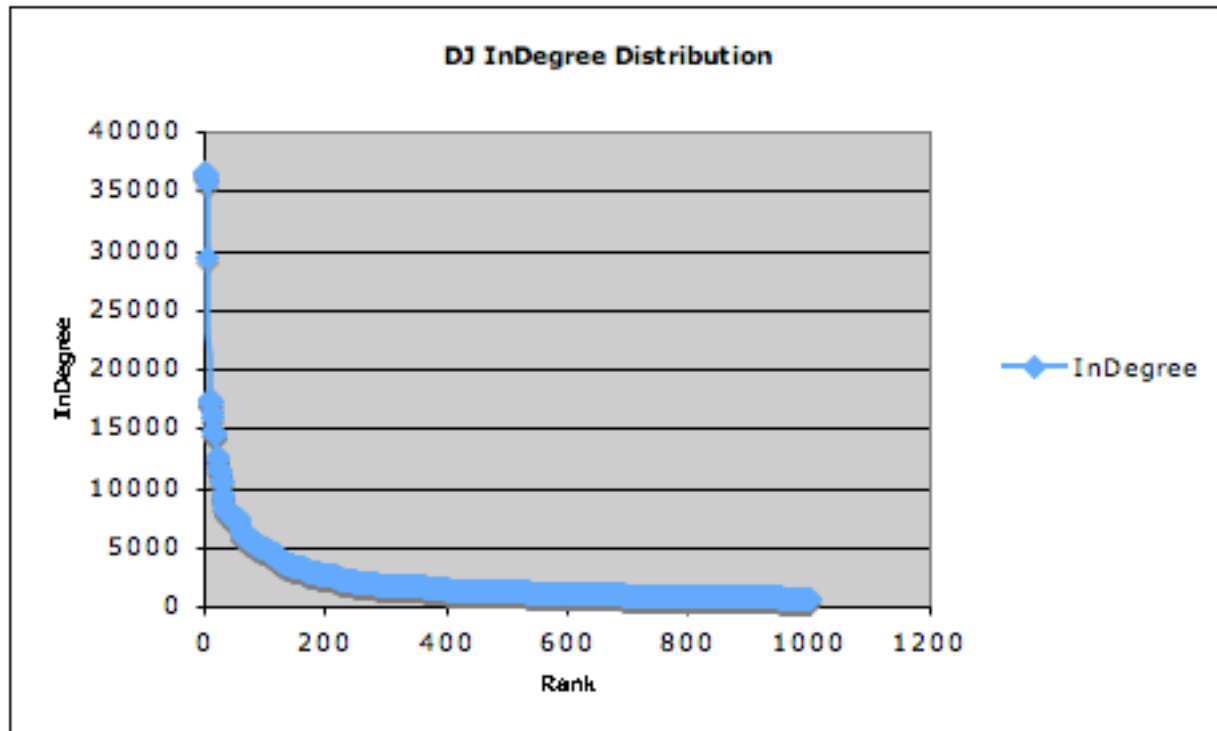
Outlier Removal

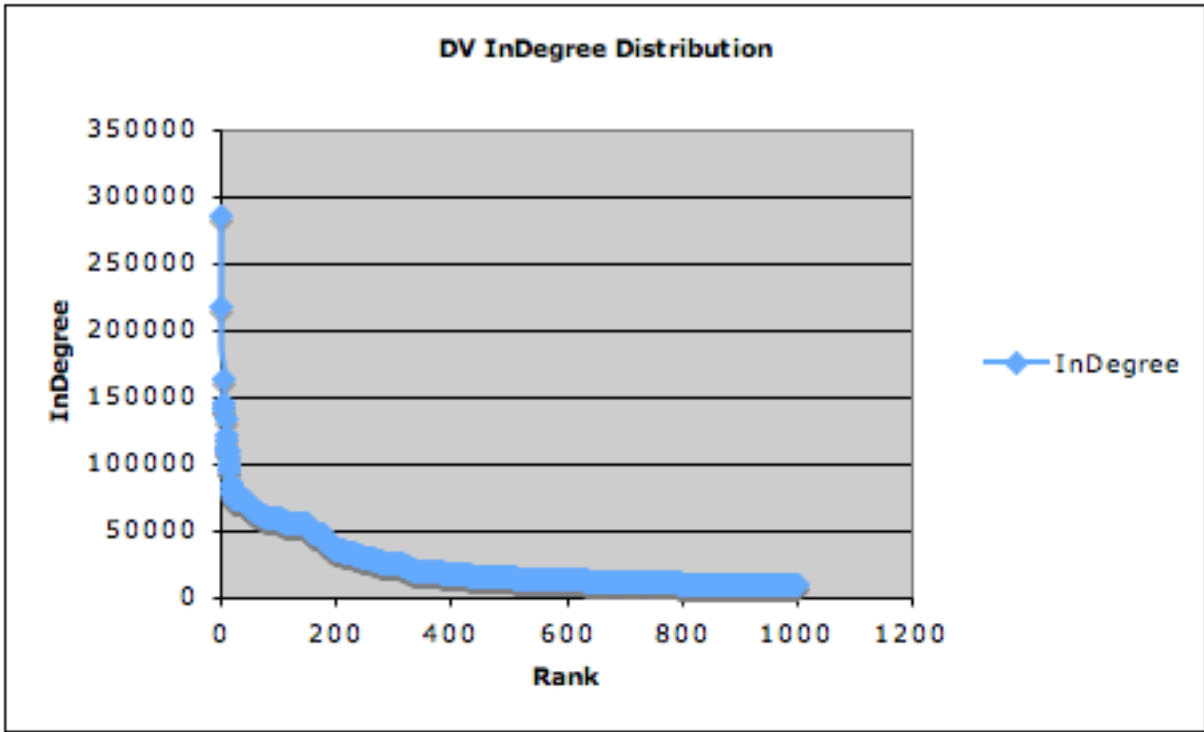
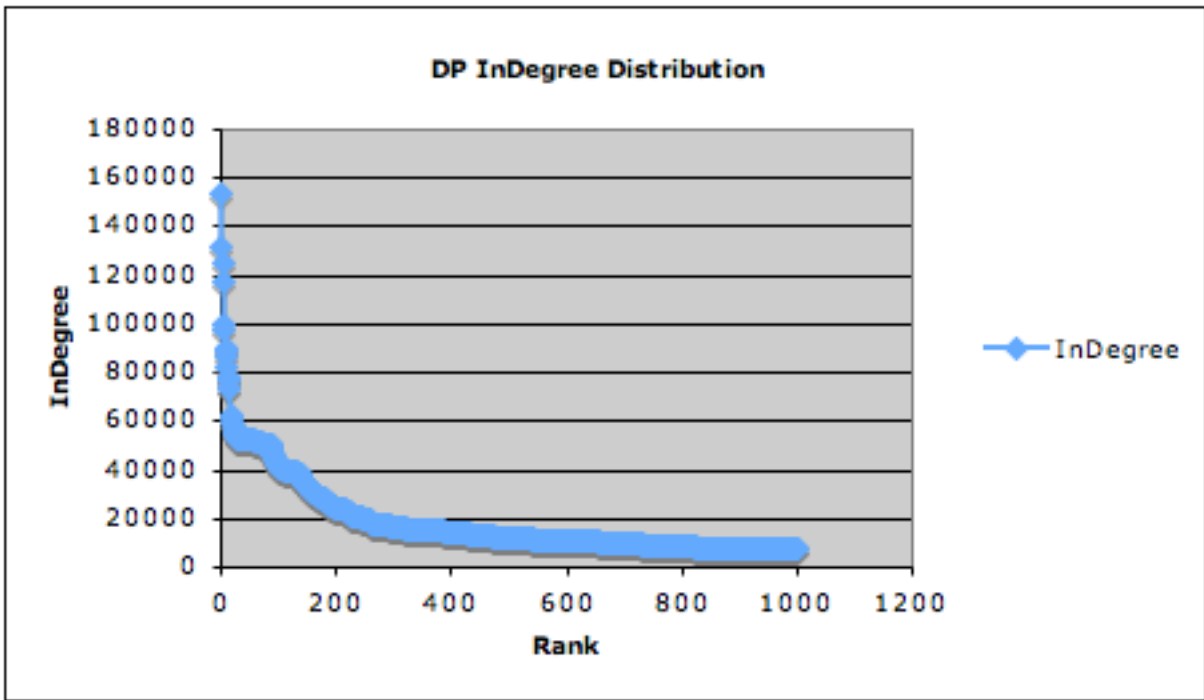
In the OutDegree distributions, large kinks appeared between the ranks of about 50 and 200 (as seen below in the graph of EB's OutDegree distribution before the removal of outliers). Upon further inspection, it was found that these kinks were caused by internal web statistics pages that were never supposed to be seen by users. This conclusion was drawn from the fact that they had huge OutDegree numbers, but very small InDegree numbers (generally fewer than 2 links in). Based on this observation, a method was devised to eliminate this type of outlier. All pages with OutDegree greater than 500 times the InDegree were removed from the data set before graphing. This removal gave the smooth curves of the graphs seen above. The risk of this type of removal was the elimination of relevant pages, but observation showed that basically only pages in the category of internal statistics were removed. Discarding these outliers eliminated the kinks in all of the OutDegree graphs.

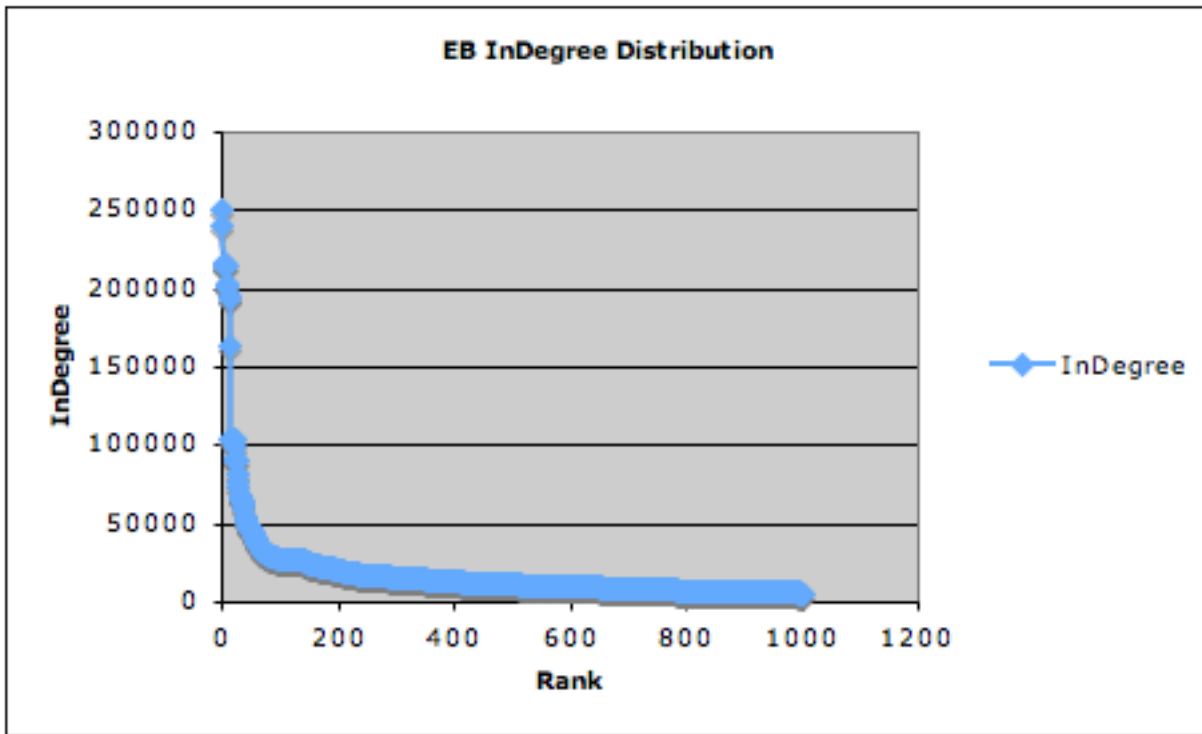


0.4.2 InDegree Distributions

The following graphs are the InDegree distributions of the top 1,000 pages in terms of InDegree in each of the four crawls. The plots below are rank-frequency plots, with the rank from 1 to 1,000 appearing on the x-axis, and the frequency (InDegree) appearing on the y-axis.







Observations

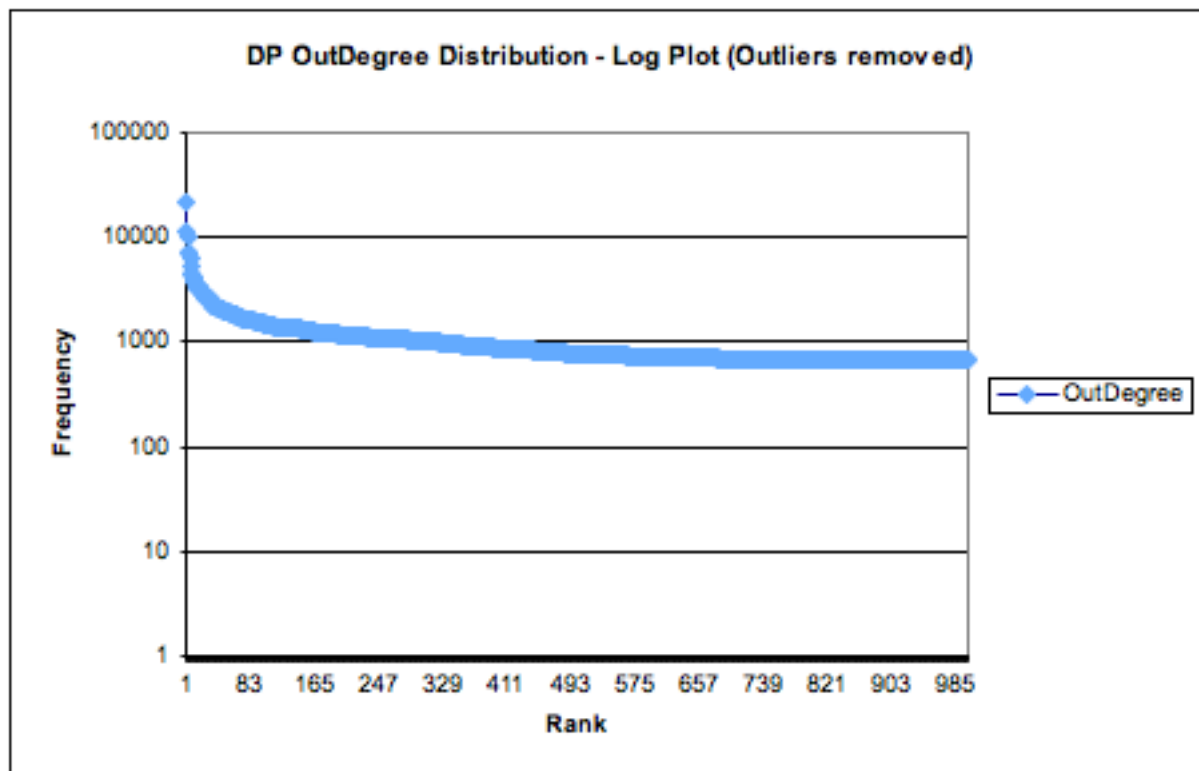
As with the OutDegree distributions, there is a major jump in the InDegree of the top-ranked page from DJ to DP. But, after that, the pattern changes. From DP to DV, the top-ranked InDegree almost doubles, from around 150,000 to around 300,000. Then, instead of significantly dropping for EB, the number stays at around 250,000. This fluctuation is significantly different from the one found in the OutDegree distributions.

Another notable feature of the graphs is that the InDegree numbers are much higher than the OutDegree numbers. This makes sense because it is quite rare to have a page with thousands upon thousands of hyperlinks leaving it, but it is quite reasonable for a page to have many thousands of other pages that point to it. The surprising fact is that the highest OutDegree totals are actually within about a factor of 10 of the highest InDegree totals, which is quite a bit closer than was expected.

0.4.3 Curve Fitting

Initial Approach - Logarithmic Fitting

The first curve fitting strategy was to fit the above distributions to a logarithmic curve of some sort. To do this log plots of all distributions were plotted. Below is a representative example, coming from the OutDegree distribution of DP.



As the above picture shows, the logarithmic curve fitting does not work properly for this distribution. The fact that the log plot does not show a linear pattern indicates that a power law curve fitting would be a better approach.

Next Approach - Power Law Fitting

The next approach was to fit the distributions to power law curves. Power law exponents were determined using the maximum likelihood estimate of exponents, which is given by the following equation:

$$\alpha = 1 + n[\sum_i \ln(\frac{x_i}{x_{min}})]^{-1} \quad (1)$$

where α is the exponent.

Then, the error is determined as follows:

$$\sigma = \frac{\alpha - 1}{\sqrt{n}} \quad (2)$$

where σ is the error.

Below are the results from running this calculation on the InDegree and OutDegree distributions from all four crawls. Unlike the graphs, which showed only the top thousand data points, these calculations were run on the entire distributions. The source code for the calculation of the exponent and the error is provided in the appendix.

Power Law Exponents

Crawl	Power Law Exponent	Error
DJ InDegree	2.5089838222688496	0.0010552633732642766
DP InDegree	2.357359366948505	.0004740268120498708
DV InDegree :	2.3869205552231136	.0004522088718363933
EB InDegree	2.231559800663152	.0004603645027594848
DJ OutDegree	1.743699745960439	.0006682445513451116
DP OutDegree	1.4409070458767528	.00021622936212034027
DV OutDegree	1.4207154713833794	.0001932240383081391
EB OutDegree	1.4055639052230275	.0002131674292500402

A few interesting trends appear in the power law exponents and error numbers. First, the exponents become smaller from DJ InDegree to DP InDegree, then hold relatively stable, then drop again down to EB. Over time, the power law exponent of the InDegree distributions is decreasing, signifying that the frequency of in links is becoming more evenly distributed. This means that as the .gov domain is growing, it is also becoming more connected. Another interesting thing to note in the InDegree distributions is the error numbers. The errors for DP, DV, and EB are all very similar, and about half of the error of DJ. This seems to indicate that the InDegree rank-frequency plots of the three later crawls better follow a power law distribution.

In the OutDegree distributions, we notice similar trends. The power law exponent decreases from crawl to crawl chronologically. Also, in the error numbers, DP, DV, and EB all have similar error numbers, which are about one third of DJ's error.

0.5 Conclusions and Future Work

This project represents an initial, exploratory step into the task of profiling the structure of the .gov domain and its development over time. InDegree and OutDegree distributions provided a simple and easy way to find influential pages within the domain as a whole and also give insight into the structure of the domain through curve fitting on the distributions.

This types of analyses can provide useful information in a couple different contexts. First, the rankings of influential pages give us some idea of the nature of current events. The pages that score highly generally reflect the issues that the government feels are most important (As whitehouse.gov's high ranking in the 2004 crawl demonstrated).

Furthermore, the curve fitting of the distributions also gave interesting insight into how the .gov domain is developing, with power law exponents following a decreasing trend in both InDegree and OutDegree distributions. This type of analysis of distributions gives information about the structure of the domain as a whole. As power law exponents decrease, the disparity in degree of pages shrinks, which is an interesting trend in the domain. This indicates that as the domain grows, it becomes more connected as well.

As far as future work goes, much remains to be done on this project. The types of analyses done in this project should be extended to more crawls after 2005. Particularly interesting would be to observe the changes from 2008 to 2009 as a new presidential administration took office. Also, the analysis should go beyond simple InDegree and OutDegree calculations. Other measures of influence in networks like PageRank and Hub and Authority scores would give new data to analyze and perhaps reinforce the conclusions made here, or lead to new insights. Overall, there are many new interesting directions for this project to take in the future.

0.6 Acknowledgements

I would like to thank my advisor Professor William Arms for his guidance throughout the project. I would also like to thank Lucia Walle of the Cornell Center for Advanced Computing for her support in maintaining the Hadoop cluster. Finally, I would like to thank Manuel Calimlim who generated the datasets from the Web Lab's database and made them available for use on the Hadoop cluster.

The Cornell Web Lab is funded in part by National Science Foundation grants CNS-0403340, SES-0537606, IIS-0634677, IIS-0705774 and IIS 0917666. The Web Lab is an NSF Next Generation Cyberinfrastructure project.

Bibliography

- [1] Internet Archive <http://www.archive.org/index.php>
- [2] Power Laws, Pareto Distributions and Zipfs Law by M. E. J. Newman.
found at: <http://arxiv.org/abs/cond-mat/0412004v3>
- [3] Hadoop. <http://hadoop.apache.org/>
- [4] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters.
In *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation*,
pages 137-149, 2004.

0.7 Appendix: Source Code

0.7.1 Power Law Exponent Calculation

```
import java.util.*;
import java.io.*;

/*
 * This program calculates the maximum likelihood power law exponent
 * of a rank frequency distribution
 */
public class PowerLaw
{

    /**
     * Running instructions: java PowerLaw file_name number_of_pages min_frequency
     * Requires: a file with lines of the following format: page_id frequency
     * Requires: the number of pages in the file
     * Requires: the minimum frequency of any page in the file
     * Outputs: Prints the exponent and error to the console.
     */
    public static void main(String [] args) throws Exception
    {
        String fileName = args[0];
        File file = new File(fileName);
        double length = Double.parseDouble(args[1].toString());
        double min = Double.parseDouble(args[2].toString());
        double sum = 0.0;
        Scanner sc = new Scanner(file);
        while(sc.hasNextLine())
        {
            String line = sc.nextLine();
            StringTokenizer st = new StringTokenizer(line);
            st.nextToken();
            double frequency = Double.parseDouble(st.nextToken());
            double term = Math.log(frequency/min);
            sum += term;
        }
        sum = (1/sum) * length + 1;
        System.out.println("The exponent is: " + sum);
        System.out.println("The error is: " + ((sum - 1)/Math.sqrt(length)));
    }
}
```

0.7.2 InDegree and OutDegree Calculation

```
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

/*
 * Class to calculate the InDegree and OutDegree of every page, given
 * input with lines of the form: fromURL toURL.
 * Each line represents a single hyperlink.
 */
public class BothDegree {

    /*
     * Mapper Class
     * This map class outputs two key-value pairs for each {fromURL, toURL} pair
     * The first has the toURL as the key and the fromURL as the value
     * (with the modifier "From" prepended)
     * The second has the fromURL as the key and the toURL as the
     * value (with the modifier "To" prepended)
     */
    public static class Map extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, Text> {

        public void map(LongWritable key, Text value,
            OutputCollector<Text, Text> output, Reporter reporter)
            throws IOException {

            String s = value.toString();
            StringTokenizer st = new StringTokenizer(s);
            String from = st.nextToken().trim();
            String to = st.nextToken().trim();
            output.collect(new Text(to), new Text("From" + from));
            output.collect(new Text(from), new Text("To" + to));
        }
    }

    /*
     * Reducer Class
     * Sums all of the "From" links and "To" links associated with each page.
     * Outputs: page InDegree OutDegree
     * for each page in the input with either in links or out links or both.
     */
}
```

```

public static class Reduce extends MapReduceBase
    implements Reducer<Text, Text, Text, Text> {

    public void reduce(Text key, Iterator<Text> values,
        OutputCollector<Text, Text> output, Reporter reporter)
        throws IOException {

        TreeSet<Text> outlinks = new TreeSet<Text>();
        TreeSet<Text> inlinks = new TreeSet<Text>();
        while (values.hasNext()) {
            String s = values.next().toString();
            if(s.startsWith("To")) outlinks.add(new Text(s));
            else inlinks.add(new Text(s));
        }
        if (Double.parseDouble(outlinks.size() + "")
            / Double.parseDouble(inlinks.size() + "") < 500)
            output.collect(key, new Text(inlinks.size() + " "
                + outlinks.size()));
    }
}

/*
 * Main method for running the MapReduce job.
 */
public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(BothDegree.class);
    conf.setJobName("BothDegree");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(Text.class);
    conf.setMapperClass(Map.class);
    conf.setReducerClass(Reduce.class);
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);
    FileInputFormat.setInputPaths(conf, new Path("input"));
    FileOutputFormat.setOutputPath(conf, new Path("output"));
    JobClient.runJob(conf);
}
}

```