CS896 Introduction to Web Science
Fall 2013
Report for Assignment 3

Corren G. McCoy

October 5, 2013

Contents

List of Figures

List of Tables

1 Question 1

1.1 Problem

Download the 1000 URIs from assignment 2 from the command line. Use a tool to remove (most) of the HTML markup.

1.2 Response

For this question, we used a Unix shell script, as shown in Appendix A to read the file (i.e., TweetFile1000.txt) containing the 1000 previously-harvested 'Twitter URIs. The script then uses Lynx to download the source HTML for each URI, then remove the HTML tags. As a file naming convention, we extracted only the top-level domain name from each URI (e.g., www.globalreport.org). This was done to avoid output errors which might occur when the full URI contained characters that could be interpreted by the operating system as a directive to access a subdirectory (e.g., `http://www.globalreport.org/usa/`).

2 Question 2

2.1 Problem

Choose a query term (e.g., "shadow") that is not a stop word and not HTML markup that matches at least 10 documents. Compute the TFIDF values for the term in each of the 10 documents. Create a table with the term frequency (TF), inverse document frequency (IDF) and TFIDF values as well as the corresponding URIs. The URIs will ranked in decreasing order by TFIDF values.

2.2 Response

As a query term, we chose *Syria*, which was one of the terms used to originally query Twitter to find our URIs. We used the Unix command shown below to locate the 10 documents with the highest frequency of this term.

```
grep -o -i syria *.processed | uniq -c | sort
```

As explained by Levine [3], in order to avoid favoring longer documents in which our query term is more likely to occur, the term frequency (TF) was normalized using the total number of words in each document. The normalized term frequency is shown in Table 1. To estimate the size of the web, we used the estimated size of Google's index, shown in Figure 1, which is approximately 43 billion pages. A query for our search term on Google indicated 413 million documents in the result set, Figure 2. Therefore, the IDF for *Syria* is the log base 2 (43000/413) or 6.7021. Finally, we calculated TFIDF[1] using the formula shown in Figure 3 to obtain our ranking, Table 2.

3 Question 3

3.1 Problem

Rank the same 10 URIs from question 2, but this time by their PageRank (PR) using any of the free PR estimators on the web. Normalize the returned values to be between 0 and 1.0. Create a table which ranks the URIs by decreasing page rank. Briefly compare and contrast the rankings produced in questions 2 and 3.

URI	TF	Total Words	Normalized TF
http://www.hhassan.com	47	3137	0.0149
http://www.nationaljournal.com	47	5044	0.0093
http://goglobalmedia.com	49	7700	0.0063
http://bigbluerock.wordpress.com	50	4360	0.0115
http://AmericanSyrians.com	47	2115	0.0222
http://beta.syriadeeply.org	54	1320	0.0409
http://brown-moses.blogspot.co.uk	62	6711	0.0092
http://www.islamicinvitationturkey.com	85	5111	0.0166
http://zamanalwsl.net	99	2565	0.0039
http://carlsonsperspective.tumblr.com	104	8301	0.0125

Table 1: Normalized Term Frequency

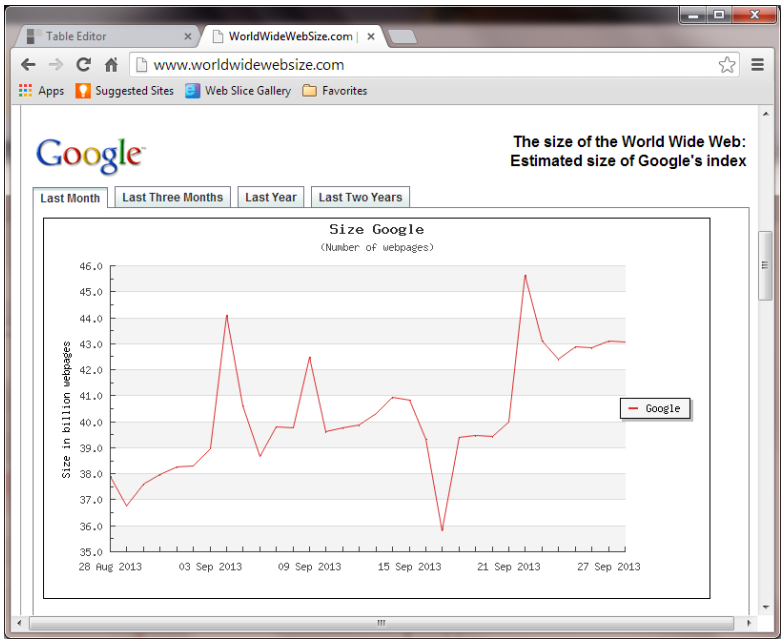

Figure 1: Estimated Size of Google's Index

TFIDF	TF	IDF	URI
0.2741	0.0409	6.7021	http://beta.syriadeeply.org
0.1488	0.0222	6.7021	http://AmericanSyrians.com
0.1112	0.0166	6.7021	http://www.islamicinvitationturkey.com
0.0999	0.0149	6.7021	http://www.hhassan.com
0.0838	0.0125	6.7021	http://carlsonsperspective.tumblr.com
0.0771	0.0115	6.7021	http://bigbluerock.wordpress.com
0.0623	0.0093	6.7021	http://www.nationaljournal.com/reporters/bio/15
0.0617	0.0092	6.7021	http://brown-moses.blogspot.co.uk
0.0422	0.0063	6.7021	http://goglobalmedia.com
0.0261	0.0039	6.7021	http://zamanalwsl.net/en

Table 2: 10 Hits for the term "Syria", ranked by TFIDF

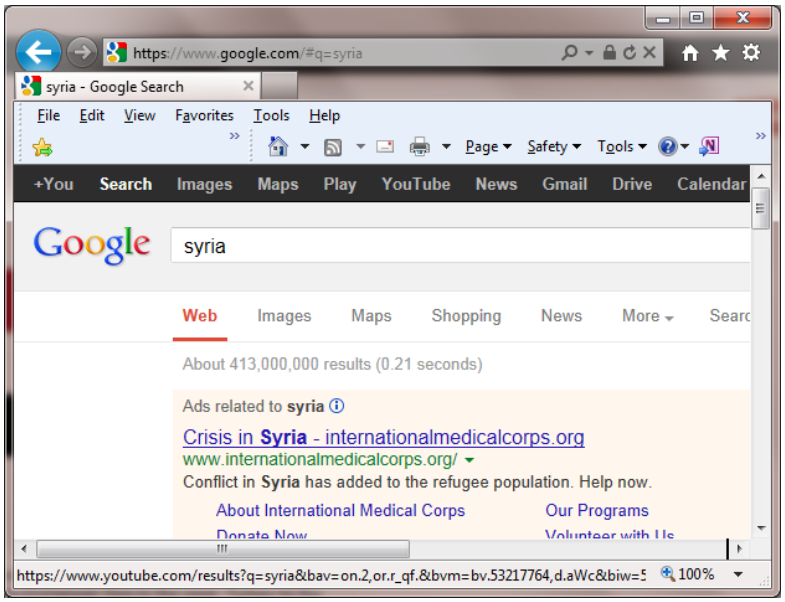

Figure 2: Query Hits for Syria

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Figure 3: TFIDF Equation

3.2 Response

We submitted each of the URIs to the Google PageRank tool (`http://www.checkpagerank.net/`). According to the site, "Google PageRank (Google PR) is one of the methods Google uses to determine a page's relevance or importance. Important pages receive a higher PageRank and are more likely to appear at the top of the search results." The page rank is reported on a scale of 0 to 10 (e.g., 5/10) which we have converted to decimal so the values are between 0 and 1. Google PR was unable to obtain a ranking for one of the URIs, NA as opposed to 0, which could indicate the site had never been indexed. A representation of the tool is shown in Figure 4. The final rankings are shown in Table 3. A comparison of the TFIDF and PageRank outcomes reveals that only one URI, `http://goglobalmedia.com`, retained the same position in both lists. This may be more coincidental than a result of any other factors. It is difficult to compare the two methodologies because they are based on different criteria. As stated by Croft [2], TFIDF "reflects the importance of a term in the collection of documents." On the other hand, page rank rates popularity based on the number and quality of the links to a particular site.

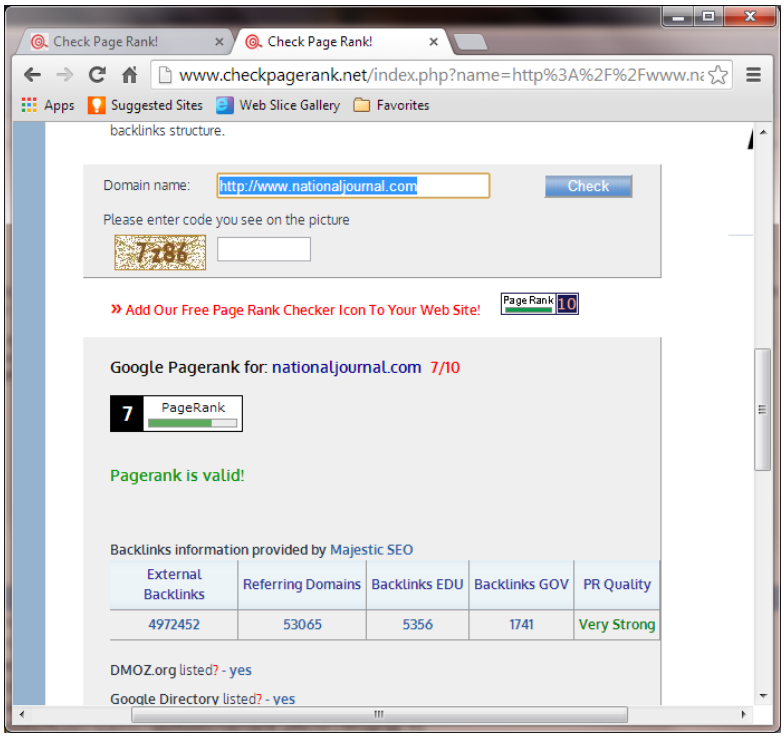

Figure 4: Google PR Tool

4 Question 4 Extra Credit

4.1 Problem

Compute the Kendall Tau_b score for both lists. Report both the Tau value and the "p" value.

4.2 Response

We used online statistics software to calculate Kendall Tau_b. The software developed by Wessa[1] implements the R Kendall library. For the first input vector, we used the numerical ranking of URIs, by

Page Rank	URI
0.07	http://www.nationaljournal.com/reporters/bio/153
0.05	http://beta.syriadeeply.org
0.05	http://AmericanSyrians.com
0.04	http://zamanalwsl.net/en
0.04	http://brown-moses.blogspot.co.uk
0.03	http://www.islamicinvitationturkey.com
0.03	http://carlsonsperspective.tumblr.com
0.01	http://bigbluerock.wordpress.com
0.00	http://goglobalmedia.com
NA	http://www.hhassan.com

Table 3: 10 Hits for the term "Syria", ranked by PageRank

TFIDF	Page Rank	URI
1	5	http://zamanalwsl.net/en
2	1	http://goglobalmedia.com
3	5	http://brown-moses.blogspot.co.uk
4	9	http://http://www.nationaljournal.com/reporters/bio/153
5	2	http://bigbluerock.wordpress.com
6	3	http://carlsonsperspective.tumblr.com
7	NA	http://www.hhassan.com
8	3	http://www.islamicinvitationturkey.com
9	7	http://AmericanSyrians.com
10	7	http://beta.syriadeeply.org

Table 4: Kendall Tau_b Input Vectors

TFIDF, sorted in ascending order. For the second vector, we used the numerical ranking corresponding to the URIs associated page rank. The data source is shown in Table 4. While the TFIDF rankings were unique, we encountered three ties among the page rank data. The output from the R calculations, Figure 5, shows a Kendall value of 0.2611 and a p-value of 0.3964. This indicates a 26 percent probability that the two ranking methods will produce the same ordering for the given set of URIs. It's more likely for the rankings to diverge. The p-value represents the confidence interval or statistical significance of tau. Our calculated value is significantly higher than the generally accepted level of 0.05 which seems to strongly suggest that we should not expect similar rankings from TFIDF and Page Rank. These results are consistent with what is evidenced in Tables 2 and 3.

[1] http://chalow.net/2005-10-12-1.html

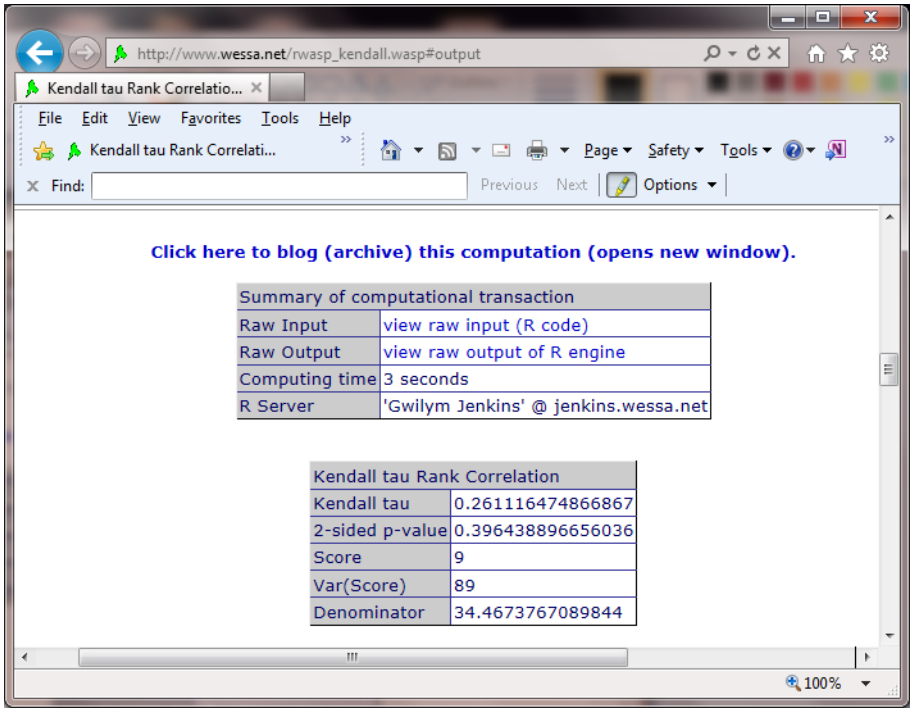

Figure 5: Kendall Rank Correlation

Bibliography

[1] Kendall tau rank correlation (v1.0.11) in free statistics software (v1.1.23-r7). `http://www.wessa.net/rwasp_kendall.wasp/`. Accessed: 2013-09-30.

[2] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice.* Addison-Wesley Reading, 2010.

[3] M. Levene. *An introduction to search engines and web navigation.* Wiley. com, 2011.

Appendix A

Source getHTMLSource Script

```
#!/bin/bash
while read line
do
uri=$line
output=$line

#Extract just the top level domain name for the output file name
output=‘echo "$output" | awk -F/ ’{print $3}’ ‘
echo $output
#Download the source for the URI
lynx -source $uri > ‘echo $output ‘

# Remove the HTML markup
lynx -dump -force_html $output > ‘echo $output.processed‘
# input the 1000 Twitter URIs
done < tweetFile1000.txt
```