CS 535: Assignment #3

Due on 11:59pm February 20, 2018

Alexander C. Nwala 16:20

David Sinclair

February 18, 2018

Contents

1 Problem 1

1.1 Question 1

1. Download the 1000 URIs from assignment #2. ”curl”, ”wget”, or ”lynx” are all good candidate programs to use. We want just the raw HTML, not the images, stylesheets, etc.

from the command line:

% curl http://www.cnn.com/ ¿ www.cnn.com

%wget -O www.cnn.com http://www.cnn.com/

% lynx -source http://www.cnn.com/ ¿ www.cnn.com

”www.cnn.com” is just an example output file name, keep in mind that the shell will not like some of the characters that can occur in URIs (e.g.,). You might want to hash the URIs to associate them with their respective filename, like:

echo -n ”http://www.cs.odu.edu/show_features.shtml?72” — md5 41d5f125d13b4bb554e6e31b6b591eeb

(”md5sum” on some machines; note the ”-n” in echo – this removes the trailing newline.)

Now use a tool to remove (most) of the HTML markup for all 1000 HTML documents. ”python-boilerpipe” will do a fair job see
(http://ws-dl.blogspot.com/2017/03/2017-03-20-survey-of-5-boilerplate.html):

```
from boilerpipe.extract import Extractor
extractor = Extractor(extractor='ArticleExtractor', html=html)
extractor.getText()
```

Keep both files for each URI (i.e., raw HTML and processed).
Upload both sets of files to your github account.

1.2 Answer 1

I created a program called 01_gethtml.sh which used bash scripting and curl to get the links file which was a total of 499 records without the twitter.com links. from assignment 2. The program took the links hashed the number to create the file name then put all the files into a directory called html. The total number of files created was 499. I archived the directory and added to the main directory called html.tar.

I then created another program using the recommend python library to processes the html documents and remove the html code and leave only text. The library ran into a couple of issues. Some of the html files had no text in them, and some of the html files had code the library was unable to identify. Out of the 499 files in the html directory, 472 files were processed with no issues. A key file was created that linked the uri to a md5 file called htmlmd5.key.

Below is an example of the output of the file.

https://www.buzzfeed.com/albertonardelli/thegovernmentsownbrexitanalysissaystheukwillbe?utm_term=.jcbRaqNkz3#.jcbRaq
048d96bf6b3ab3a7f855e600ec0fa1dd.html
https://www.instagram.com/p/BennV8RHV4p/ 72f66bc4d6e9ba35875f52526ebc151e.html
https://mashable.com/2018/01/31/apple-responds-us-government-iphone-throttling-probe/ 1111953b36d7632584b9607977a5b51
http://www.ParliamentToday.com/free/viewnews.html?id=99247 4399a4fe928063c2d16fc5094701b1ed.html
http://zeenews.india.com/hindi/india/modi-government-will-abolish-posts-vacant-from-past-5-years/369486 fc22ab56e514be6052

12 files had no text a sample of the list is provided but the entire list can be seen in the file called "NotextProcessing.txt".

No Text. b8ebc6df67a08cd31cd94bc1fa3c547c.html
No Text. b30ee796fc9b349f27ee551d29f933da.html
No Text. d0f33467d1dfd4353b8d9114d79f18cd.html
No Text. c5542b9fca268da7a319fc24bb8fe238.html
No Text. 2fdfa52642586bb5774f0f37880ef2ad.html

Another 16 files had code the library was unable to identify and a sample of the list is provided but the entire list can be seen in the file called "ErrorProcessing.txt".

UnicodeDecodeError. 1ed4459af9a5ed696690864be613702f.html
UnicodeDecodeError. 82527b6708b8b5da93fa240c0f3731dc.html
UnicodeDecodeError. cfac5dc77582a98b10d2ba6b4cf49b64.html
UnicodeDecodeError. 14ae8829ac6414af9b21b7ad632d9695.html

2 Problem 2

2.1 Question 2

2. Choose a query term (e.g., "shadow") that is not a stop word (see week 5 slides) and not HTML markup from step 1 (e.g., "http") that matches at least 10 documents (hint: use "grep" on the processed files). If the term is present in more than 10 documents, choose any 10 from your list. (If you do not end up with a list of 10 URIs, you've done something wrong).

As per the example in the week 5 slides, compute TFIDF values for the term in each of the 10 documents and create a table with the TF, IDF, and TFIDF values, as well as the corresponding URIs. The URIs will be ranked in decreasing order by TFIDF values. For example:

Table 1. 10 Hits for the term "shadow", ranked by TFIDF.

TFIDF TF IDF URI
— – — —
0.150 0.014 10.680 http://foo.com/
0.044 0.008 10.680 http://bar.com/

You can use Google or Bing for the DF estimation. To count the number of words in the processed document (i.e., the deonminator for TF), you can use "wc":

% wc -w www.cnn.com.processed
2370 www.cnn.com.processed

It won't be completely accurate, but it will be probably be consistently inaccurate across all files. You can use more accurate methods if you'd like, just explain how you did it.

Don't forget the log base 2 for IDF, and mind your significant digits!

https://en.wikipedia.org/wiki/Significant_figures#Rounding_and_decimal_places

2.2 Answer 2

The term that I looked for was "minister". Ouc of the 472 files it was found in 76 files. They ranged from max of 16 in one document to 1 in many other documents. The list of files and the number of times minister was in each file is in a file called "minwordcount.txt".

I sorted the minwordcount.txt file and selected the following 10 files to rank with the corrisponding minister count.

Table 1. 10 Hits for the term "minister" by Count.

File Name	Count	Line Count	Word Count	Char Count
f81f0a8863d96799127430d352b8570f.txt	16	181	3862	24029
fea2be6a9c57ddd7026b56029291b080.txt	6	92	1899	11247
ea2dc97574e60ecb10171ec08afabeb7.txt	6	26	1031	6375
91f0ccb56b3daa6095ac19def910cefb.txt	5	33	733	4387
c95f9367f4aaf444cb1d33c3f5f86ea5.txt	4	8	279	1738
2e124b0f8e405b73483fa6364732a868.txt	4	19	419	2409
75d0df4f758c79db48f58b50ebe100d5.txt	4	17	361	2109
936767979808b7b3348a18366158559e.txt	4	21	472	2742
7e36f66513f6cd8cae045e3e814bedd0.txt	3	21	532	3250
af9ca261c6e9e3937080a10e9b86b92e.txt	3	28	563	3321

To determine IDF the number according to bing was 220,000,000 for the Document for Coupe from http://www.worldwidewebsize.com and the Minister search using Bing pulled 40,700,000 documents. This created the IDF number of .73283.

Table 2. 10 Hits for the term "minister", ranked by TFIDF.

TFIDF	TF	IDF	File Name	URI
.01051	.01434	.73283	c95f9367f4aaf444cb1d33c3f5f86ea5.txt	http://www.app.com.pk/imran-not-blame-federal-gover
.00812	.01108	.73283	75d0df4f758c79db48f58b50ebe100d5.txt	https://www.ndtv.com/india-news/in-embarrassment-f
.00700	.00955	.73283	2e124b0f8e405b73483fa6364732a868.txt	http://www.abplive.in/india-news/up-minister-embarra
.00621	.00847	.73283	936767979808b7b3348a18366158559e.txt	https://www.ndtv.com/indianews/inembarrassmentfor
.00500	.00682	.73283	91f0ccb56b3daa6095ac19def910cefb.txt	https://www.politicshome.com/news/uk/political-parti
.00427	.00582	.73283	ea2dc97574e60ecb10171ec08afabeb7.txt	https://economictimes.indiatimes.com/news/politicsan
.00413	.00564	.73283	7e36f66513f6cd8cae045e3e814bedd0.txt	https://www.reuters.com/article/uk-britain-eu-may-sp
.00391	.00533	.73283	af9ca261c6e9e3937080a10e9b86b92e.txt	http://www.bbc.co.uk/news/ukpolitics42884610
.00303	.00414	.73283	f81f0a8863d96799127430d352b8570f.txt	http://www.abc.net.au/news/2018-01-31/cabinet-files-
.00232	.00316	.73283	fea2be6a9c57ddd7026b56029291b080.txt	http://www.independent.co.uk/news/uk/politics/there

3 Problem 3

3.1 Question 3

3. Now rank the same 10 URIs from question #2, but this time by their PageRank. Use any of the free PR estimaters on the web, such as:

http://pr.eyedomain.com/
http://www.prchecker.info/check_page_rank.php
http://www.seocentro.com/tools/search-engines/pagerank.html
http://www.checkpagerank.net/

If you use these tools, you'll have to do so by hand (they have anti-bot captchas), but there are only 10 to do. Normalize the values they give you to be from 0 to 1.0. Use the same tool on all 10 (again, consistency is more important than accuracy). Also note that these tools typically report on the domain rather than the page, so it's not entirely accurate.

Create a table similar to Table 1:

Table 2. 10 hits for the term "shadow", ranked by PageRank.

PageRank URI
——— —
0.9 http://bar.com/
0.5 http://foo.com/

Briefly compare and contrast the rankings produced in questions 2 and 3.

3.2 Answer 3

I used http://pr.eyedomain.com for PageRank

Table 3. 10 Hits for the term "minister", ranked by PageRank.

Page Rank	URI
9	http://www.bbc.co.uk/
8	https://www.reuters.com
8	http://www.abc.net.au
8	http://www.independent.co.uk
7	https://economictimes.indiatimes.com
6	https://www.ndtv.com/
6	https://www.ndtv.com/
5	http://www.app.com.pk/
N/A	http://www.abplive.in/
Can't Find	https://www.politicshome.com

Briefly compare and contrast the rankings produced in questions 2 and 3.

In looking at question 2 and question 3. I was expecting to see that the document that had the highest number of hits with the key word be the one in both questions. But I was wrong. The TFIDF for this document was actually on the lower end because it actually and a lot of words in the document. This document, f81f0a8863d96799127430d352b8570f.txt, which had the word Minister appear 16 times. actually ranked at 9 with a score of .00303. The document, c95f9367f4aaf444cb1d33c3f5f86ea5.txt, that actually ranked highest in the TFIDF had the least number of words, but had the word Minister more times. Document c95f9367f4aaf444cb1d33c3f5f86ea5.txt

website was http://www.app.com.pk and document f81f0a8863d96799127430d352b8570f.txt website was http://www.abc.net.au. So in looking at the page ranking by the website http://www.abc.net.au was ranked 8 and https://www.app.com.pk was ranked 5. The http://pr.eyedomain.com used a Google page rank from 0-9 with 0 being the worst and 9 being the best. So in looking at the two tables they almost reverse order.

4 Problem 4

4.1 Question 4

==========
=======Question 4 is for 3 points extra credit========
==========

4. Compute the Kendall Tau_b score for both lists (use "b" because there will likely be tie values in the rankings). Report both the Tau value and the "p" value.

See:
http://stackoverflow.com/questions/2557863/measures-of-association-in-r-kendalls-tau-b-and-tau-c
http://en.wikipedia.org/wiki/Kendall_tau_rank_correlation_coefficient#Taub
http://en.wikipedia.org/wiki/Correlation_and_dependence

4.2 Answer 4

5 Problem 5

5.1 Question 5

==========
=======Question 5 is for 3 points extra credit========
==========

5. Compute a ranking for the 10 URIs from Q2 using Alexa information (see week 4 slides). Compute the correlation (as per Q4) for all pairs of combinations for TFIDF, PR, and Alexa.

5.2 Answer 5

6 Problem 6

6.1 Question 6

==========
======Question 6 is for 2 points extra credit=======
==========

6. Give an in-depth analysis, complete with examples, graphs, and all other pertinent argumentation for Kristen Stewart's (of "Twilight" fame) Erdos-Bacon number.

6.2 Answer 6

Kristen Stewart co-wrote an academic paper about artificial intelligence. The paper was called "Bringing impressionism to life with neural style transfer in Come Swim".
http://www.businessinsider.com/twilights-kristen-stewart-co-authored-a-paper-on-artificial-intelligence-2017-1

Using this paper and counting the citations in it. I came up with an EROS number of 5.

https://mathscinet-ams-org.proxy.lib.odu.edu/mathscinet/MRAuthorID/189017

Table 6: Paul Erdos to Kristen Stewart work

Name	relation	name	Work Number
Paul Erdos	coauthored	Persi W. Diaconis	MR2126886
Persi W. Diaconis	coauthored	Bernd Sturmfels	MR1608156
Bernd Sturmfels	coauthored	Jean Ponce	MR3641809
Jean Ponce	coauthored	Andrew Zisserman	MR2912359
Andrew Zisserman	cited	Kristen Stewart	Not list in MathSci

Then to determine Kristen Stewart Bacon Number, I used the website "The Oracle of Bacon" According to the website:

Kristen Stewart has a Bacon number of 2

Table 6: Kevin Bacon to Kristen Stewart work

Name	Movie	Name
Kevin Bacon	Criminal Law	David Gow
David Gow	On the Road	Kristen Stewart

From this information I add the two numbers together to get Kristen Stewart Erdos-Bacon number which is 7.

7 Problem 7

7.1 Question 7

==========
======Question 7 is for 2 points extra credit=======
==========

7. Build a simple (i.e., no positional information) inverted file (in ASCII) for all the words from your 1000 URIs. Upload the entire file to github and discuss an interesting portion of the file in your report.

7.2 Answer 7

I created this. The file is of interest because it took a while to create. The files are located in 3 files total. The first is called quest7.txt. This is the raw text of all the files in processed. Then I removed foreign languages, and punctuation. I then went online and used a alphabetizer online to put all the words in order.

https://www.textfixer.com/tools/alphabeticalorder.php

The output was then copied to quest7a.txt. I removed additional foreign languages and additional punctuation and did it again. The results are listed in quest7b.txt.

I then modified my program 04_word.py and created anther program called 07_word.py and did a search of the following words: Trump, congress, government, justice, minister, the.

I thought these words would be interesting. The word I was must interested in was government. This is the word that I used to do my initial search in twitter for. I expected to see government in most of the documents so roughly about 472 governments but in reality. Out government was used 1415 times way more than I was expecting. Trump was used 238 and congress was 4. Making me think that alot of the articles dealt with more that just the United States government. Justice was used 38 times, minister was used 193 times. Minister showed that quite a few articles dealt with governments outside the United States. I also wanted to see the number of times a common word was used. I chose 'the' which was used 12910 times in all 472 documents.