# Assignment 3

Ryan Dill
CS 432
February 18, 2018

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. For this assignment I created a bash script, assignment3_1.sh, that hashed the URI from twitgone.txt from the last assignment. From there I ran a python script, assignment3_2.py, that extracted the text from the html. These files, along with the html.key file which stores the hash table, are in the html folder.

Here are some example links:
http://www.forgotten5.com/2018/02/13/conference-usa-basketball-power-rankings-feb-13/ , ff68305e81bacc865be648f9bd58fedc.html

https://www.scacchoops.com/boston-college-at-pittsburgh-basketball-live-stats-02132018 , 31c93820b343ecf628f756681c3280c0.html
http://www.espn.com/mens-college-basketball/story/_/id/22402389 , e257af757c1eca5737ece8876557cb4a.html
http://www.southernminn.com/waseca_county_news/sports/article_38b53f34-3ada-56d6-b804-d3cd363f36fa.html , d9da56d677de24cfce8af19c769617f9.html
http://www.orlandosentinel.com/sports/florida-state-seminoles/chopping-block/os-sp-fsu-basketball-0214-story.html , 97fe4eeeaf8944b8f250676e96b76cf8.html
https://dukereport.com/duke-basketball/duke-vs-virginia-tech-game-notes-dis/ , d192a8f861e8139dae8b7e4ac5220e09.html

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

For this assignment I chose the word “NCAA” and sorted the content of mincount.txt and got the following.

| File Name | Count | Words |
|---|---|---|
| 349ec66442a410163ddd3130f75e5c08.txt | 16 | 4102 |
| 2334985fd799ef1ca23751bb8756d6f8.txt | 6 | 874 |
| 37b448a3130861a07bc81b3f77072b07.txt | 4 | 504 |
| d9aff423f6e55e87f26af8b426cbcc6c.txt | 2 | 399 |
| b7512b6982abbcd3385b59d9e0ea2e94.txt | 1 | 45 |
| d1cc32376d1f774ccf05767a31fafc66.txt | 1 | 180 |
| 3ba1c07f0ff1c05ad0e3d2d469009614.txt | 1 | 19 |
| aabeda93399488822f80a811e7b1f781.txt | 1 | 200 |
| 32bc42a50ba3917e76c041c9ec40bbb1.txt | 1 | 85 |
| 13bb96482834bc8d6b365a08ef05661d.txt | 1 | 12 |

According to Google, they have 47,000,000,000 websites in the corpus, and Google has 115,000,000 websites that hit for NCAA.  This gives me an IDF of 8.6749.

Calculating each page for this IDF gives me the following table.

| File Name | TF | TFIDF |
|---|---|---|
| 349ec66442a410163ddd3130f75e5c08.txt | 0.0039 | 0.0338 |
| 2334985fd799ef1ca23751bb8756d6f8.txt | 0.0069 | 0.0599 |
| 37b448a3130861a07bc81b3f77072b07.txt | 0.0079 | 0.0688 |
| d9aff423f6e55e87f26af8b426cbcc6c.txt | 0.0050 | 0.0435 |
| b7512b6982abbcd3385b59d9e0ea2e94.txt | 0.0222 | 0.1928 |
| d1cc32376d1f774ccf05767a31fafc66.txt | 0.0555 | 0.4814 |
| 3ba1c07f0ff1c05ad0e3d2d469009614.txt | 0.0526 | 0.4563 |
| aabeda93399488822f80a811e7b1f781.txt | 0.0050 | 0.0434 |
| 32bc42a50ba3917e76c041c9ec40bbb1.txt | 0.0118 | 0.1021 |
| 13bb96482834bc8d6b365a08ef05661d.txt | 0.0833 | 0.7229 |

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.

For this question I used http://pr.eyedomain.com and I received the following websites(Note: there was a bit of overlap in websites so there are less than 10 total domains):

| Website | Webrank |
|---|---|
| ESPN.com | 8 |
| Reuters.com | 8 |
| Foxsports.com | 7 |
| CBSsports.com | 8 |
| Bleacherreport.com | 6 |
| Basketballinsider.com | 7 |

I would say most of these TFIDF scores don’t necessarily match up to theses websites webrank.  I believe this is due to the fact that many of these web pages are small, since it was most likely a link to a stats page posted on twitter, or even a small caption on a video.  Most of the websites I used are well-known sports sites and have a high Webrank.