CS 432/532 Web Science: Assignment #3

Alexander C. Nwala

Bethany DeMerchant
28 February 2019

Contents

Problem 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. Now use a tool to remove (most) of the HTML markup for all 1000 HTML documents. "python-boilerpipe" will do a fair job. Keep both files for each URI (i.e., raw HTML and processed). Upload both sets of files to your GitHub account.

Solution 1:

Pages were downloaded using `curl [URI]`. A Python script was used to run the command, capture the result, and output it to a file for each URI. `Python-boilerpipe`[1] was then used to extract text from the pages. The original files are located in `A3/results/curl` and the processed files are located in `A3/results/text`.

Problem 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. 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.

Solution 2:

The word "charge" was used as a query term for this exercise. A python script was used to execute `grep -li charge [filename]` on each of the processed HTML files. This returned a list of 13 pages, 10 of which were selected for further analysis. IDF was calculated using Bing as a search engine, returning 39,300,000 results, and assuming 20 billion indexed pages as outlined in the week 5 slides[2]. Calculations are available in `A3/wordcount.xlsx`.

TFIDF	TF	IDF	URI
0.272	0.0303	8.99	https://www.newsbreakapp.com/police-two-women-arrested-for-drugs-one-for-felony-gun-possession-in-nashville-rental?id=0LIYXUrl&s=a2&pd=42740242%0A
0.071	0.0079	8.99	https://www.ebay.com/itm/SUNDANCE-Womens-Leather-Asymm-Skirt-Size-4-Lined-Brown-Diagonal-Zip-Up-9-Slit/264204640641?hash=item3d83d2e181:g:iaQAAOSwgX9ccErS%0A
0.045	0.0050	8.99	https://www.foxnews.com/us/2-arrests-officer-on-desk-duty-after-wild-nyc-highway-chase?utm_source=dlvr.it&utm_medium=twitter%0A
0.036	0.0040	8.99	https://www.nytimes.com/2019/03/01/opinion/trump-michael-cohen-crime.html
0.034	0.0038	8.99	https://theheartysoul.com/neutralize-radiation-with-himalayan-salt-lamp/?t=scom&W=Viral%0A
0.032	0.0035	8.99	https://www.hugedomains.com/domain_profile.cfm?d=dainashi&e=com#PBOrIjF.twitter_tweet_ninja_l%0A
0.025	0.0028	8.99	https://www.nbcbayarea.com/news/local/BAY-ONLYPolice-Arrest-Suspect-in-Attack-on-Conservative-Activist-at-UC-Berkeley-506570131.html?akmobile=o%0A
0.013	0.0014	8.99	https://te4.org/characters/231996/tome/57b169b8-ec49-413a-9c58-0334f151d729%0A
0.003	0.0004	8.99	https://www.theatlantic.com/health/archive/2016/07/trump-and-sociopathy/491966/
0.003	0.0003	8.99	https://s2.washingtonpost.com/camp-rw/?e=andpbmViYW5rc0BnbWFpbC5jb20%3D&s=5c78ca8bfe1ff6099d6f6476%0A

[1] https://github.com/slaveofcode/boilerpipe3
[2] https://anwala.github.io/lectures/cs532-s19/

Problem 3:

Now rank the same 10 URIs from question #2, but this time by their PageRank. Use any of the free PR estimators on the web. 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). Briefly compare and contrast the rankings produced in questions 2 and 3.

Solution 3:

PageRank estimation was completed using `checkpagerank.net`. As specific pages are more difficult to estimate, estimation was done using the top domain only.

PR	URI
0.9	www.nytimes.com
0.9	www.ebay.com
0.9	www.hugedomains.com
0.8	www.foxnews.com
0.8	www.theatlantic.com
0.7	www.nbcbayarea.com
0.7	s2.washingtonpost.com
0.6	theheartysoul.com
0.5	te4.org
0.4	www.newsbreakapp.com

One interesting result of comparing the TFIDF and PR results is the fact that despite having the highest TFIDF the PageRank for `www.newsbreakapp.com` is the lowest of the sample. This is likely caused in part by the low wordcount of the specific page. Although calculating TF as the percentage of words in the document that represent the search term, a short document still needs few instances of the search term to reach a high TF. `www.hugedomains.com` has a very high PageRank but a low TFIDF because PageRank is a measure of inlinks rather than relevance. `www.ebay.com` is in the top rankings for both PR and TFIDF. The results show that the number of links to a page has limited relation to the relevance of the page to a given search term.