|  |
| --- |
|  |
| CS432 Spring 2018 |
| Assignment 3 |
|  |
| **Jonathan Kruszewski** |
| **20/02/2018** |

|  |
| --- |
|  |

**Part 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.

Now use a tool to remove (most) of the HTML markup for all 1000 HTML documents.

Keep both files for each URI (i.e., raw HTML and processed).

Upload both sets of files to your github account.

To accomplish this task I used two python programs, “GetHTML.py” and “GetRawText.py”. First “GetHTML.py” was executed and ran requests.get(URL).text and printed the result to an individual file for each URL, which can be found in the folder “HTML Files”, any requests.get() execution that resulted in an error would simply print an empty html file.

After “GetHTML.py” completed I ran “GetRawText.py” which opened each html file for and reads the contents of the file into a variable which is fed into a boilerpipe Extractor as a string as raw\_html. The extracted content is then printed to an individual text file for each URL, these can be found in “Raw Text Files”

for x in range(len(urlTable)):

try:

uHTML = requests.get(urlTable[x][1]).text

urlID = urlTable[x][0]

#uHTML = requests.get("http://www.odu.edu/compsci").text

print(uHTML, file=open(urlID+"RawHtml.html", "w+"))

print(urlID, " Succeeded")

except:

print(genericErrorInfo(), file=open(urlID + "RawHtml.html", "w+"))

print(urlID, " Failed")

genericErrorInfo()

try:

htmlID = htmlTable[x][0]

rawHtml = open(htmlID+htmlFile, "r").read()

extractor = Extractor(extractor='ArticleExtractor', html=rawHtml)

print(extractor.getText(), file=open(htmlID+"RawText.txt", "w+"))

print(htmlID, " Success")

except:

htmlID = htmlTable[x][0]

print(genericErrorInfo(), file=open(htmlID+"RawText.txt", "w+"))

print(htmlID, " Failed")

genericErrorInfo()

**Part 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:

To accomplish this task I use an unsaved python program to print out the grep command ‘grep -i -c "Sport" [1000 URL Raw Text File] > grepOutput.txt’. I then used another unsaved python program to trim the “grepOutput.txt” file down and remove any with 0 instances of “Sport”, using this output file I selected 10 URIs select those with the most results and recorded the selected URIs and the count of “Sport” in the text file “AS3P2UrlID.txt”. I then opened each raw text file for the selected URIs in LibreOffice to check the number of words and calculated the TF for each URI.

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

|  |  |  |  |
| --- | --- | --- | --- |
| TFIDF | TF | IDF | URI |
| 0.79 | 0.030438312 | 25.88955559257228 | https://www.usatoday.com/story/sports/nfl/2018/02/11/eagles-lane-johnson-calls-patriots-fear-based-organization/327887002/?utm\_source=feedblitz&utm\_medium=FeedBlitzRss&utm\_campaign=usatodaycomsports-topstories |
| 0.46 | 0.017913593 | 25.88955559257228 | http://www.cbc.ca/news/thenational/sport-doping-scandal-ben-johnson-russia-ioc-steroids-1.4511130?cmp=rss&utm\_source=dlvr.it&utm\_medium=twitter |
| 0.42 | 0.01618929 | 25.88955559257228 | https://fitlivinglifestyle.com/the-best-sports-video-games-today/ |
| 0.36 | 0.014038657 | 25.88955559257228 | https://www.azcentral.com/picture-gallery/sports/high-school/2017/10/28/arizona-high-school-boys-and-girls-soccer-2017-18/107032128/ |
| 0.35 | 0.013645973 | 25.88955559257228 | https://www.commercialappeal.com/story/sports/columnists/geoff-calkins/2018/02/12/winter-olympics-2018-shaun-white-goes-deep/328303002/ |
| 0.29 | 0.011363636 | 25.88955559257228 | https://www.marketingweek.com/2018/02/08/brands-neglecting-womens-sports-sponsorship/#.WoA960xGt\_M.twitter%0A |
| 0.25 | 0.009546539 | 25.88955559257228 | https://www.vanityfair.com/style/2018/02/sports-illustrated-swimsuit-metoo-era?mbid=social\_twitter |
| 0.24 | 0.009120931 | 25.88955559257228 | https://www.argusleader.com/story/sports/winter-olympics-2018/2018/02/11/mirai-nagasu-lands-triple-axel-first-american-do-winter-olympics/328064002/ |
| 0.19 | 0.007411983 | 25.88955559257228 | https://www.golfdigest.com/story/the-winter-olympics-are-amazing-but-also-a-shining-beacon-of-sports-privilege?utm\_source=dlvr.it&utm\_medium=facebook |
| 0.07 | 0.002821759 | 25.88955559257228 | https://frntofficesport.com/why-the-nba-nhl-mls-and-nfl-are-in-on-esports/ |

**Part 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:

Using <http://pr.eyedomain.com/> I found the rank for each of the 10 URIs from part 2.

Table 2. 10 hits for the term "Sport", ranked by PageRank, using <http://pr.eyedomain.com/>.

|  |  |
| --- | --- |
| Page Rank | URI |
| 0.8 | https://fitlivinglifestyle.com/the-best-sports-video-games-today/ |
| 0.8 | https://www.azcentral.com/picture-gallery/sports/high-school/2017/10/28/arizona-high-school-boys-and-girls-soccer-2017-18/107032128/ |
| 0.8 | https://www.commercialappeal.com/story/sports/columnists/geoff-calkins/2018/02/12/winter-olympics-2018-shaun-white-goes-deep/328303002/ |
| 0.7 | https://www.marketingweek.com/2018/02/08/brands-neglecting-womens-sports-sponsorship/#.WoA960xGt\_M.twitter%0A |
| 0.6 | https://www.vanityfair.com/style/2018/02/sports-illustrated-swimsuit-metoo-era?mbid=social\_twitter |
| 0.6 | https://www.argusleader.com/story/sports/winter-olympics-2018/2018/02/11/mirai-nagasu-lands-triple-axel-first-american-do-winter-olympics/328064002/ |
| 0.5 | https://www.golfdigest.com/story/the-winter-olympics-are-amazing-but-also-a-shining-beacon-of-sports-privilege?utm\_source=dlvr.it&utm\_medium=facebook |
| 0.0 | https://frntofficesport.com/why-the-nba-nhl-mls-and-nfl-are-in-on-esports/ |
| Unable to rank | https://www.usatoday.com/story/sports/nfl/2018/02/11/eagles-lane-johnson-calls-patriots-fear-based-organization/327887002/?utm\_source=feedblitz&utm\_medium=FeedBlitzRss&utm\_campaign=usatodaycomsports-topstories |
| Unable to Rank | http://www.cbc.ca/news/thenational/sport-doping-scandal-ben-johnson-russia-ioc-steroids-1.4511130?cmp=rss&utm\_source=dlvr.it&utm\_medium=twitter |