Assignment 3

Part 1

Wrote script using hashlib and os to download html from URL list, saved in .txt files

```
import hashlib
import sys
import os

input_file = open(sys.argv[1],'r')

for line in input_file:
    site = line.strip()

    #make md5 hash name
     m= hashlib.md5()
    name = site
    #os.system(' echo -n ' + name)
    name2 = name.encode('utf-8')
    #os.system(' echo -n ' + str(name2))
     m.update(name2)
    hname = m.hexdigest()
    #os.system(' echo -n ' + hname)
    #make file to be imported to
    print(hname +'.html')
    #run os command to get web page, put into made file
    os.system('curl '+site+' > ' + hname +'.html ')
```

After failing to get goose, justext, or python-boiler to function, used code provided by instructor in slack with a loop to iterate through HTML files and remove the majority of boilerplate, saving the remaining text as the name of the original file with .txt added at the end.

```
for file in os.listdir(directory):
    filename = file.strip()
    if filename.endswith(b'.html'):
        try:
            openfile = open(file, 'r')
            rawtext = openfile.read()
            #text = clean_html(rawtext)

            # First we remove inline JavaScript/CSS:
            cleaned = re.sub(r"(?is)<(script|style).*?>.*?(</\1>)", "", rawtext.strip())
            # Then we remove html comments. This has to be done before removing
regular
            #tags since comments can contain '>' characters.
            cleaned = re.sub(r"(?s)<!--(.*?)-->[\n]?", "", cleaned)
            # Next we can remove the remaining tags:
            cleaned = re.sub(r"(?s)<.*?>", " ", cleaned)
            # Finally, we deal with whitespace
            cleaned = re.sub(r"&nbsp;", " ", cleaned)
            cleaned = re.sub(r"  ", " ", cleaned)
            cleaned = re.sub(r"  ", " ", cleaned)

            #my addition to remove blank lines
            cleaned = re.sub("\n\s*\n*", "\n", cleaned)



            #os.system(str(cleaned) + ' > ~/ass3/rawHTML/textFiles/' + str(filename) +
'.txt')
            openfile.close()

            with open(str(filename.decode("utf-8"))+'.txt', 'w') as f:
                print(cleaned, file=f)
        except:
            continue
```

Part 2

Patched together a set of code to return the occurrence of a word in a file and total word count of files that had a least one occurrence, saved result in a text file. Word chosen was “Toronto.”

```
import os
import sys

directory = os.fsencode(sys.argv[1])
checkedpos = 0
checked = 0
files = 0
for file in os.listdir(directory):
    files = files + 1
    filename = file.strip()
    #if str(filename).endswith('html.txt'):
    try:
        count = 0
        word = 'Toronto'
        #print ( word)
        with open(filename, 'r') as f:
            checked = checked + 1
            for line in f:
                words = line.split()
                for i in words:
                    #print(i)
                    if(i == word):
                        #print('true')
                        count=count+1

        if count > 0:
            checkedpos = checkedpos +1
            num_words = 0
            #print('b4 wordcoutn loop')

            with open(filename, 'r') as f:
                #print('file opened for wordcount')

                for line in f:
                    words = line.split()
                    num_words += len(words)
                    #print(str(num_words))
            print(str(count)+' '+str(word)+' occurance out of '+str(num_words)+' in file
'+str(filename))

    except:
        continue
print(str(checked) + ' files of ' + str(files) + ' files checked, '+str(checkedpos)+' files
with matches')
```

Modified script to get HTML to output a list of URLs and their hash.

Manually added 10 URLs and the wordcount stats to a text file tenSites.txt. Used data in this file and wordcount.txt to fill out following table.

TFIDF	TF	IDF	URL
.096	0.012	7.964	https://wayback.archive.org/web/20180210163606/https://www.thestar.com/news/gta/2017/10/06/toronto-writers-tweet-on-her-sexual-harassment-story-spurs-others-to-share-experiences.html
.215	.027	7.964	https://wayback.archive.org/web/20180208163805/https://www.thestar.com/news/gta/2018/02/08/one-dead-after-mississauga-house-fire.html
.080	.010	7.964	https://www.thestar.com/news/gta/2018/02/08/ontario-campuses-see-increase-in-precarious-jobs-study-shows.html
.016	.026	7.964	https://wayback.archive.org/web/20180210164249/https://www.thestar.com/news/gta/2017/09/30/man-in-serious-condition-after-shooting-in-etobicoke.html
.119	.015	7.964	http://www.cbc.ca/news/canada/toronto/snow-storm-gta-weekend-1.4527926
.078	.006	7.964	https://www.thestar.com/amp/news/gta/2017/02/08/honest-eds-sign-will-move-to-ed-mirvish-theatre.html
.127	.016	7.964	https://www.thestar.com/news/gta/2018/02/10/brrrave-souls-take-polar-plunge-in-fundraiser-for-ontario-special-olympics.html
.088	.011	7.964	https://www.thestar.com/news/gta/2017/12/31/city-says-shelters-still-had-space-but-people-at-moss-park-site-left-scrambling.html
.183	.023	7.964	https://wayback.archive.org/web/20180210171201/https://www.thestar.com/news/gta/2017/09/02/eb-express-lanes-closed-on-hwy-401-in-mississauga-after-collision.html
.088	.011	7.964	https://www.thestar.com/news/gta/2018/02/02/gay-village-stalked-by-a-serial-killera-second-time.html

Part 3

Pages sorted by pagerank:

8 http://www.cbc.ca/ (1 page)

6 https://wayback.archive.org/ (4 pages)

0 https://www.thestar.com/ (5 pages)

I'm not really noticing any correlation between pagerank and TFIDF, though this is probably due to my sample being bad. f I had time to spare, I would go get another set that at least consisted of unique domains, but as is I am out of time.