2017

Assignment 3

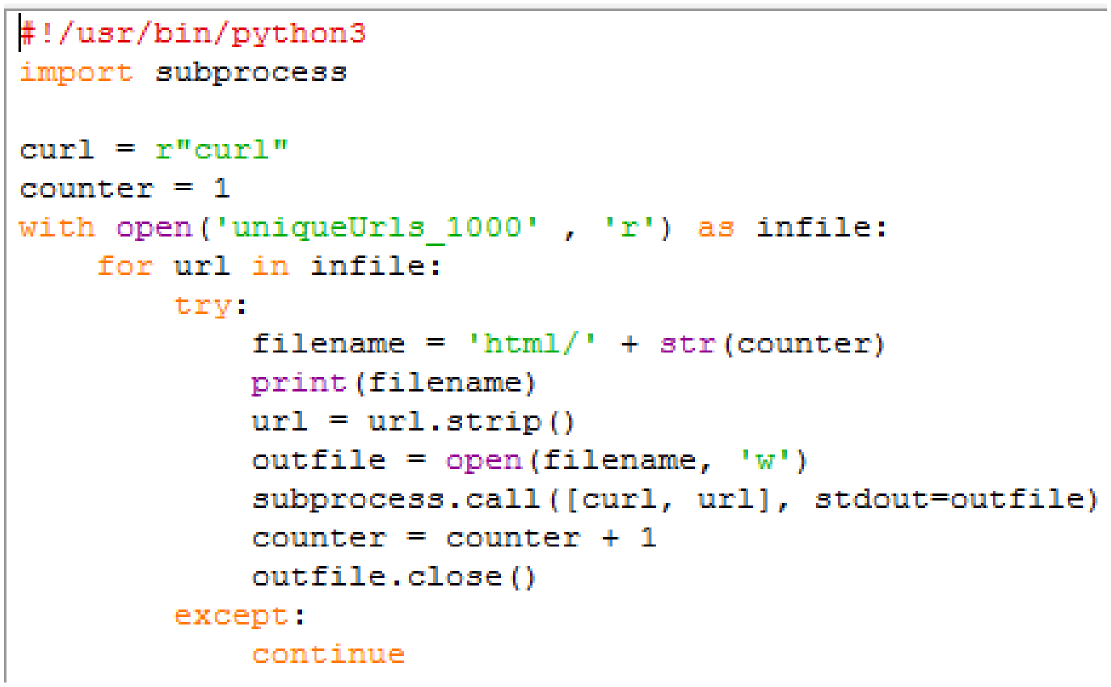
CS 432

MIRANDA SMITH

Question 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.

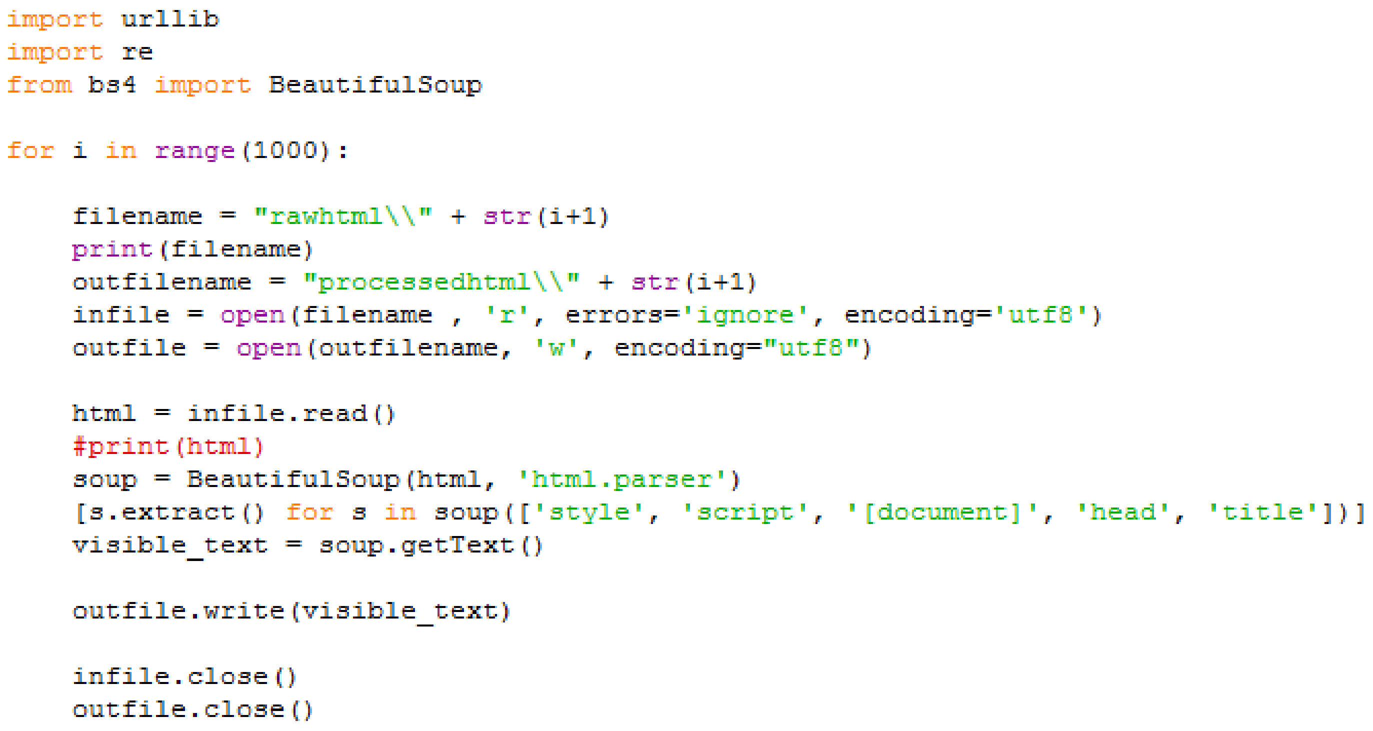
I downloaded all of the html to the folder 'rawhtml' with the python program 'gatherHTML.py'.



```
#!/usr/bin/python3
import subprocess

curl = r"curl"
counter = 1
with open('uniqueUrls_1000' , 'r') as infile:
    for url in infile:
        try:
            filename = 'html/' + str(counter)
            print(filename)
            url = url.strip()
            outfile = open(filename, 'w')
            subprocess.call([curl, url], stdout=outfile)
            counter = counter + 1
            outfile.close()
        except:
            continue
```

If there was an issue with the html, i.e. the webpage doesn't exist anymore, it skips it and moves on. Instead of using a hash to name each file I used the line number it is stored in the file as. Every file has been kept in order and blank lines have been inserted when data wasn't available, i.e. no creation date for assignment 2.

I used the program ‘processHTML.py’ to clean up each file and show only visible text. I used a combination of different techniques shown here: http://stackoverflow.com/questions/1936466/beautifulsoup-grab-visible-webpage-text.



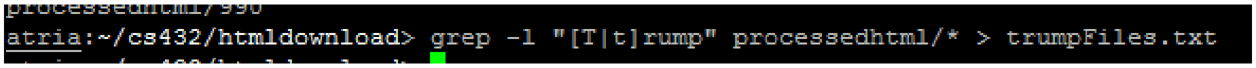
```python
import urllib
import re
from bs4 import BeautifulSoup

for i in range(1000):

    filename = "rawhtml\\" + str(i+1)
    print(filename)
    outfilename = "processedhtml\\" + str(i+1)
    infile = open(filename , 'r', errors='ignore', encoding='utf8')
    outfile = open(outfilename, 'w', encoding="utf8")

    html = infile.read()
    #print(html)
    soup = BeautifulSoup(html, 'html.parser')
    [s.extract() for s in soup(['style', 'script', '[document]', 'head', 'title'])]
    visible_text = soup.getText()

    outfile.write(visible_text)

    infile.close()
    outfile.close()
```

For each file I cleaned up the text to show only visible text and placed it in the folder ‘processedhtml’. To clean up each file I used Beautiful Soup 4. I stripped all the lines with a style, script, document, head, and title tags and used Beautiful Soup’s getText function to grab what would be on the page otherwise.

Question 2

Choose a query term that is not a stop word and not HTML markup from question 1 that matches at least 10 documents. 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.

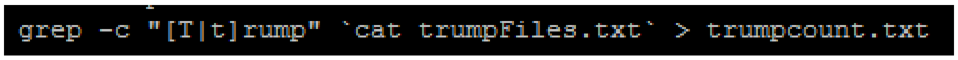
I used Trump\trump as my key word because I know it was a popular topic on twitter when I got my data. I used grep to find all documents containing the key word and placed them in a file.



```
processedhtml/990
atria:~/cs432/htmldownload> grep -l "[T|t]rump" processedhtml/* > trumpFiles.txt
```

I got 107 files. I narrowed it down to 10 by personal choice, based off randomness and peeking at the number of occurrences to give a good variety.

I got the count for each chosen document with another grep only outputting the count for each listed file.



```
grep -c "[T|t]rump" `cat trumpFiles.txt` > trumpcount.txt
```

I used Excel to calculate the TF-IDF. To calculate TF I did $\frac{Term\ Occurance\ in\ Document}{Word\ Count\ of\ Document}$, which was gathered above. To calculate IDF I did $Log_2 \frac{Size\ of\ Internet}{Total\ Docs\ with\ term}$. To get the size of the internet I used http://www.worldwidewebsize.com/. To get the total number of documents with the term 'Trump' I used Bing to search the term and got its estimate for total results. For TF-IDF, the TF and IDF were just multiplied.

TFIDF	TF	IDF	URI
0.185	0.023	8.2	http://www.independent.co.uk/news/world/americas/donald-trump-barack-obama-i-can-feel-he-likes-me-us-president-white-house-opinions-a7567151.html
0.096	0.012	8.2	https://www.nytimes.com/
0.082	0.010	8.2	https://actualidad.rt.com/actualidad/230369-eeuu-nino-matar-hermana-herir-hermano-pistola
0.053	0.006	8.2	https://www.ft.com/content/a8f3af30-ec63-11e6-930f-061b01e23655?ftcamp=published_links/rss/world/feed//product
0.053	0.006	8.2	https://www.ft.com/content/43c3f688-ec54-11e6-ba01-119a44939bb6?ftcamp=published_links%2Frss%2Fcompanies%2Ffeed%2F%2Fproduct
0.024	0.003	8.2	http://www.timeslive.co.za/thetimes/2017/02/07/Alleged-paedophile-free-to-visit-public-spots1?utm_source=dlvr.it&utm_medium=twitter
0.011	0.001	8.2	http://www.bfmtv.com/international/israel-le-parlement-adopte-une-loi-controversee-sur-les-colonies-1097850.html
0.011	0.001	8.2	http://fxvnpro.com/forex-fundamental-analysis.php?id=4010
0.008	0.001	8.2	https://actions.sumofus.org/a/amazon-stop-investing-in-hate
0.008	0.001	8.2	https://leftfootforward.org/2017/02/can-we-exit-brexit-greens-are-going-to-court-to-find-out/

Question 3

Rank the same 10 URIs from question #2, but this time by their PageRank.

I used http://pr.eyedomain.com/ and manually entered each URI to get their PageRank. Some of the pages gave a PankRank of N/A which I replaced with 0. The range of values I was given was between 0 and 9. To properly convert that to a 0.0 to 1.0 range I used the formula $X_{new} = {X_{old} - \min(x)} / {\max(x) - \min(x)}$.

PageRank	URI
0.9	http://www.independent.co.uk/news/world/americas/donald-trump-barack-obama-i-can-feel-he-likes-me-us-president-white-house-opinions-a7567151.html
0.9	https://www.ft.com/content/a8f3af30-ec63-11e6-930f-061b01e23655?ftcamp=published_links/rss/world/feed//product
0.9	https://www.ft.com/content/43c3f688-ec54-11e6-ba01-119a44939bb6?ftcamp=published_links%2Frss%2Fcompanies%2Ffeed%2F%2Fproduct
0.8	https://actualidad.rt.com/actualidad/230369-eeuu-nino-matar-hermana-herir-hermano-pistola
0.8	http://www.bfmtv.com/international/israel-le-parlement-adopte-une-loi-controversee-sur-les-colonies-1097850.html
0.7	http://www.timeslive.co.za/thetimes/2017/02/07/Alleged-paedophile-free-to-visit-public-spots1?utm_source=dlvr.it&utm_medium=twitter
0.6	https://actions.sumofus.org/a/amazon-stop-investing-in-hate
0.6	https://leftfootforward.org/2017/02/can-we-exit-brexit-greens-are-going-to-court-to-find-out/
0.0	https://www.nytimes.com/
0.0	http://fxvnpro.com/forex-fundamental-analysis.php?id=4010

Comparing this table from the TF-IDF table, the first entry retained its top rank while the others got rearranged. For example, the nytimes.com URI got placed all the way at the bottom of the PageRank table, while it was 2nd in the TF-IDF table. The two ft.com sites also knocked the actualidad.rt.com site from its 3rd rank in the TF-IDF table to take 2nd and 3rd in the PageRank table. While the ft.com sites had just over half the TF-IDF scores of the nytimes.com and the actualidad.rt.com sites the page rank of that domain was high enough that its relevance would probably place those sites higher in a search engine.

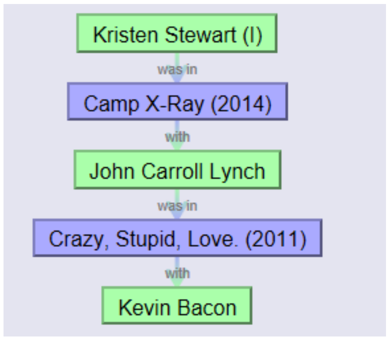
Question 6

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

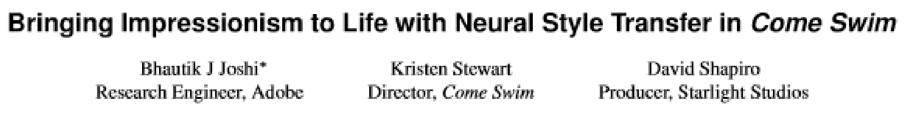
Calculating Bacon number:

https://oracleofbacon.org/

Bacon number of **2**



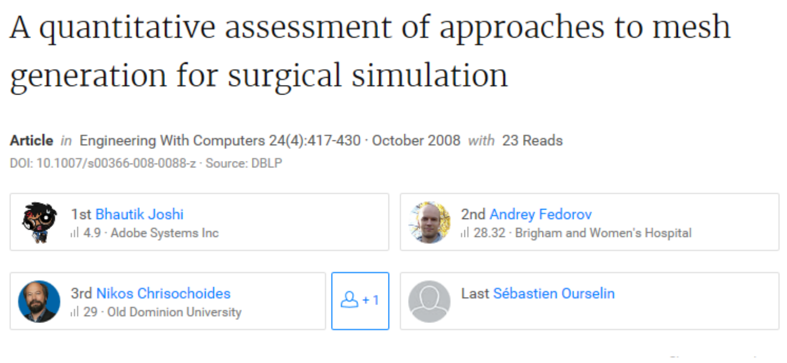
Calculating Erdos number:



Bringing Impressionism to Life with Neural Style Transfer in *Come Swim*

Bhautik J Joshi* — Research Engineer, Adobe
Kristen Stewart — Director, *Come Swim*
David Shapiro — Producer, Starlight Studios

1 https://arxiv.org/pdf/1701.04928v1.pdf



A quantitative assessment of approaches to mesh generation for surgical simulation

Article *in* Engineering With Computers 24(4):417-430 · October 2008 *with* 23 Reads
DOI: 10.1007/s00366-008-0088-z · Source: DBLP

1st Bhautik Joshi
4.9 · Adobe Systems Inc

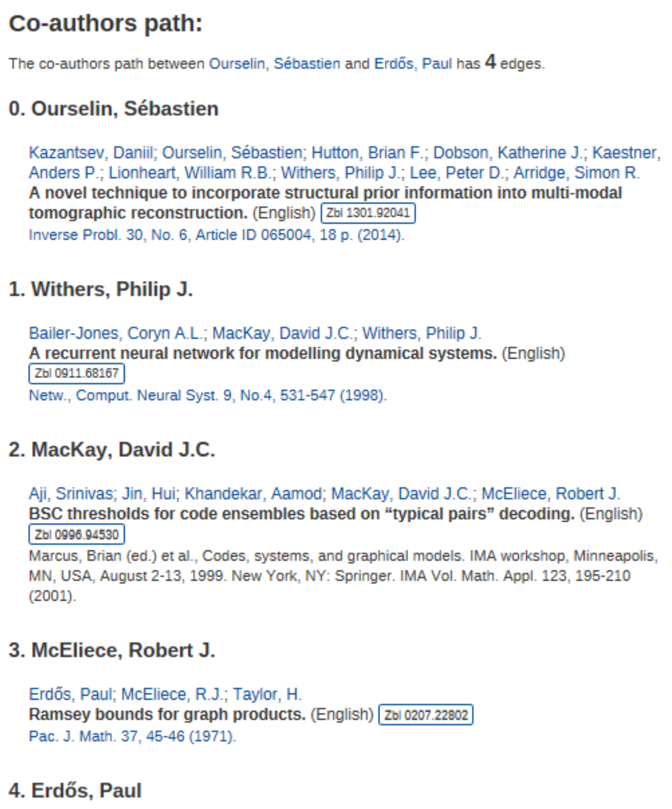
2nd Andrey Fedorov
28.32 · Brigham and Women's Hospital

3rd Nikos Chrisochoides
29 · Old Dominion University

+1

Last Sébastien Ourselin

2 https://www.researchgate.net/publication/220677921_A_quantitative_assessment_of_approaches_to_mesh_generation_for_surgical_simulation



Co-authors path:

The co-authors path between Ourselin, Sébastien and Erdős, Paul has **4** edges.

0. Ourselin, Sébastien

Kazantsev, Daniil; Ourselin, Sébastien; Hutton, Brian F.; Dobson, Katherine J.; Kaestner, Anders P.; Lionheart, William R.B.; Withers, Philip J.; Lee, Peter D.; Arridge, Simon R.
A novel technique to incorporate structural prior information into multi-modal tomographic reconstruction. (English) Zbl 1301.92041
Inverse Probl. 30, No. 6, Article ID 065004, 18 p. (2014).

1. Withers, Philip J.

Bailer-Jones, Coryn A.L.; MacKay, David J.C.; Withers, Philip J.
A recurrent neural network for modelling dynamical systems. (English) Zbl 0911.68167
Netw., Comput. Neural Syst. 9, No.4, 531-547 (1998).

2. MacKay, David J.C.

Aji, Srinivas; Jin, Hui; Khandekar, Aamod; MacKay, David J.C.; McEliece, Robert J.
BSC thresholds for code ensembles based on "typical pairs" decoding. (English) Zbl 0996.94530
Marcus, Brian (ed.) et al., Codes, systems, and graphical models. IMA workshop, Minneapolis, MN, USA, August 2-13, 1999. New York, NY: Springer. IMA Vol. Math. Appl. 123, 195-210 (2001).

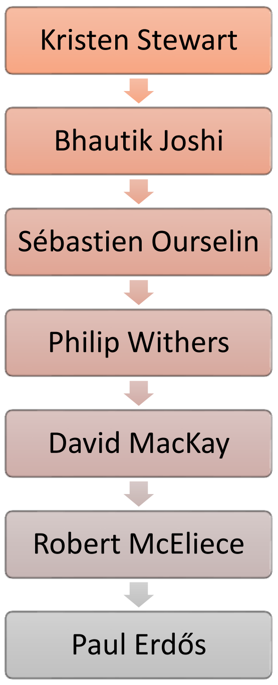
3. McEliece, Robert J.

Erdős, Paul; McEliece, R.J.; Taylor, H.
Ramsey bounds for graph products. (English) Zbl 0207.22802
Pac. J. Math. 37, 45-46 (1971).

4. Erdős, Paul

3 https://zbmath.org/collaboration-distance/?a=ourselin.sebastien&b=erdos.paul

In Summary:



Erdos number of 6

Erdos-Bacon number = 8