CS 432: Homework Number Seven

Due on April 8, 2019 at 4:20 PM

Alexander Nwala

Tim Bruce

Problem 1

Create a blog-term matrix. Start by grabbing 100 blogs hosted on https://www.blogger.com. Include: http://f-measure.blogspot.com/ http://ws-dl.blogspot.com/
The method described in class no longer works. So use your discretion to grab 98 more blogs. Describe how you generated the blog data - manually or automatically. Note that each student will separately do this process, so it is unlikely that these 98 blogs will be shared among students. In other words, no sharing of blog data. Upload to github your code (if you used one) for grabbing the blogs and provide a list of blog URIs, both in the report and in github.
Use the blog title as the identifier for each blog (and row of the matrix). Use the terms from every item/title (RSS) or entry/title (Atom) for the columns of the matrix. The values are the frequency of occurrence. Essentially you are replicating the format of the "blogdata.txt" file included with the PCI book code. Limit the number of terms to the most "popular" (i.e., frequent) 1000 terms, this is *after* the criteria on p. 32 (chapter 3 PCI book) (slide 8 - Week 11) has been satisfied. Remember that blogs are paginated (slide 46).

Solution
My main discovery for this question was that if you type "blogspot" into google with another term following it, you'll get a related blog from blogspot. With this knowlege, I was able to get enough unique blogs for this assignment. The biggest problem was when google would ban me for a couple of hours when they realized that I was being uncool. Because of this, I limited my calls to one per 60 seconds.

Listing 1: Getting random blogs.

```
finalResults = []
    for query in getNewWords():
        print(query)
        queryResults = []
        query = query + " .blogspot.com"
        time.sleep(60)
        for j in search(query, num=10, stop=20, pause=2):
            if ".blogspot.com" in j:
                queryResults.append(j)
                f =open("blogs.txt", "a")
                f.write(j)
                f.write('\n')
                f.close()
        finalResults.append(queryResults)
        print(queryResults[0])
    print(finalResults)
```

Unfortunately, this returned a lot of odd blogs, so characterization at the end may be a challenge. I also suspect that this will lead to fairly unrelated blogs as well.
I was then able to get the rss feed from the pages using a handy algorithm from user alexmill on github in their project 'python3feedfinder', which basically takes the uri as input and outputs a link to the rss feed.
From this point on, a lot of the work is from the book "Programming Collective Intelligence" book by Toby Segaran. I was able to modify the code from the book for use with Python 3.5 which I am using. I was able to take the most popular 1000 terms and organize them as a blog-term matrix.

Problem 2

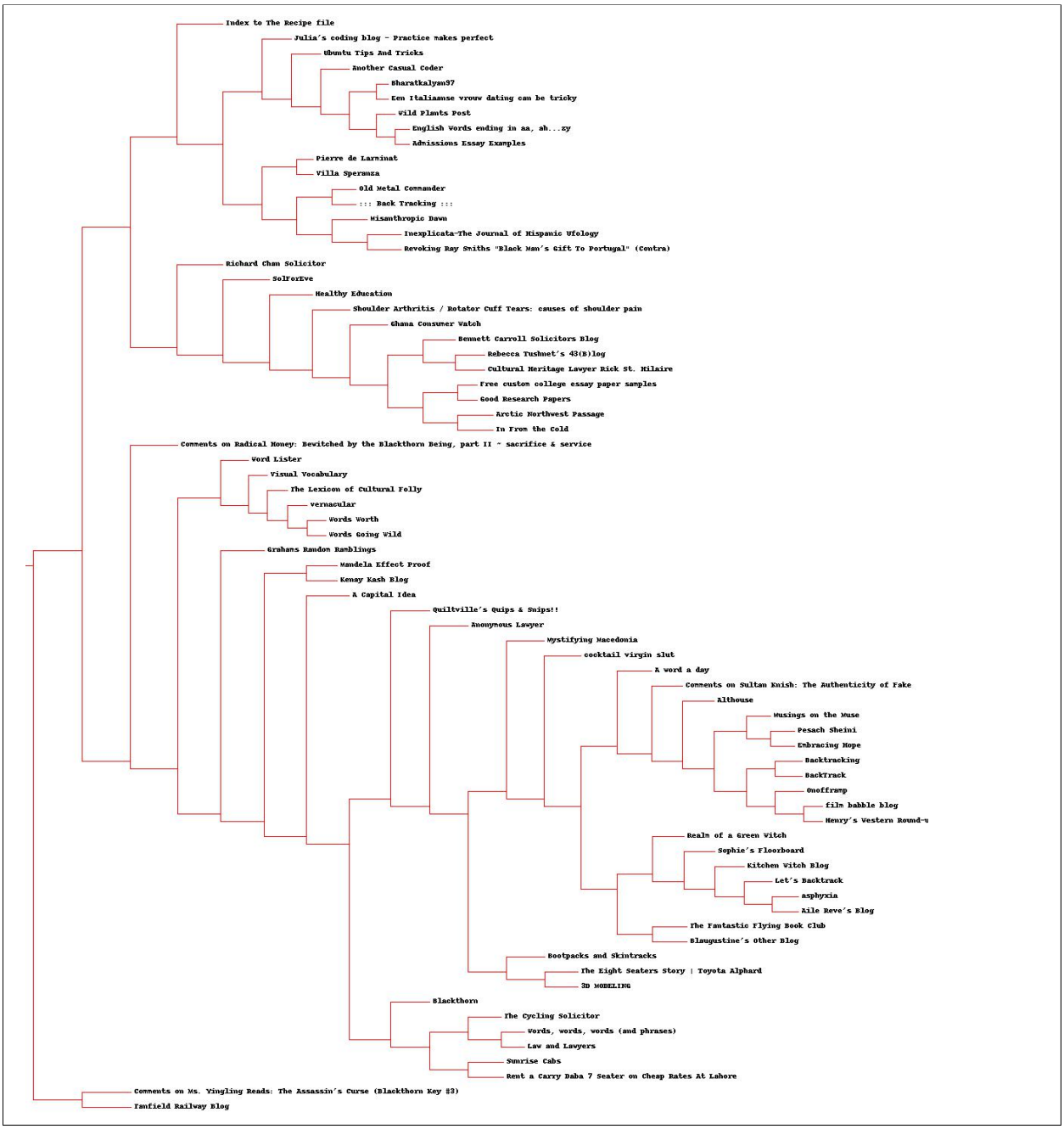
Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs (see slides 13 & 14). Include the JPEG in your report and upload the ascii file to github (it will be too unwieldy for inclusion

in the report).

Solution

Once again the PCI book has come in very handy. I was able to use its dendrogram method to create a very good ASCII dendrogram which is included in the code, as this is a bad format for the full ASCII dendrogram. However, for part two of this problem, the JPEG can be shown, but it is still unreadable.

Image 1: Dendrogram of similar blogs.



As you can see, the blogs have been clustered together.

Problem 3

Cluster the blogs using K-Means, using k=5,10,20. (see slide 25). Print the values in each centroid, for each value of k. How many iterations were required for each value of k?

Solution
Once again, I have used the PCI book code to solve the problem. This time I converted their K-Means code into Python 3 and ran it three times, for k=5, k=10 and k=20. k=5 and k=20 took five iterations, and k=10 took four iterations, however as the k number increased, the computer took longer to process each iteration. The values in each centroid were too large to fit into this form factor, but can be found in the too-big-for-latex.txt file in this directory.

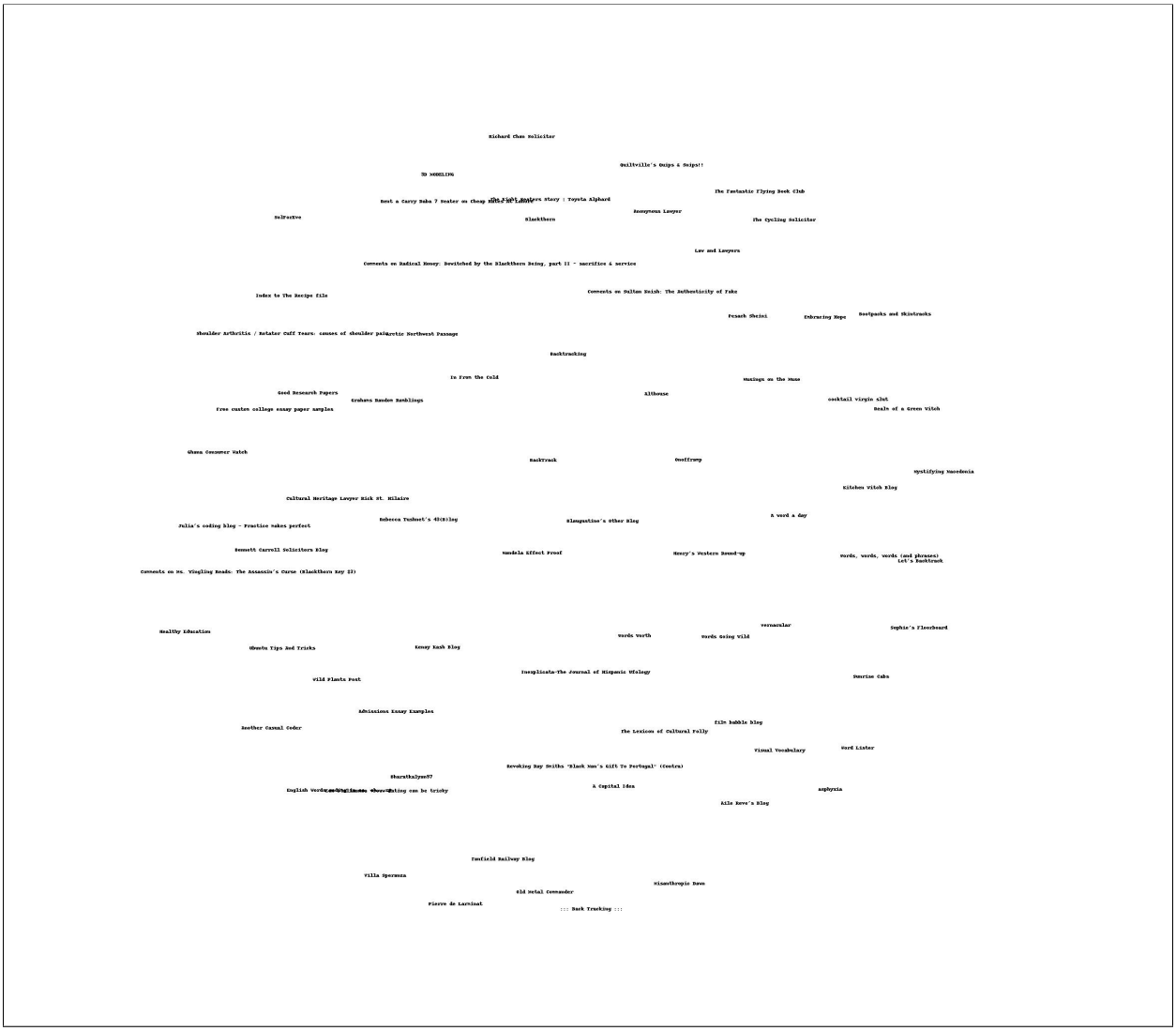
Problem 4

Use MDS to create a JPEG of the blogs similar to slide 29 of the week 11 lecture. How many iterations were required?

Solution
This is once again from the PCI book. It took 335 iterations to my understanding to complete the scale down, and produced a beautiful result.

Image 2: Image of similar blogs arranged by distance.

Problem 4 continued on next page...



Problem 5

Not attempted.

Problem 6

Re-run questions 1-4, but this time instead of using the 98 "random" blogs, use 98 blogs that should be "similar" to:
http://f-measure.blogspot.com/
http://ws-dl.blogspot.com/
Choose approximately equal numbers for both blog sets (it doesn't have to be a perfect 49-49 split, but it should be close). Explain in detail your strategy for locating these blogs.
Compare and contrast the results from the 98 "random" blogs and the 98 "targeted" blogs.

Solution
Starting from the top, problem 1.

Problem 1

I was able to find 100 targeted blogs with some difficulty. They range from music to concert blogs in one half, and computer science department to environmental science blogs on the other.

Problem 2

This file is called sblogclust.jpeg in the code, and stoo-big-for-latex.txt

Image 3: Dendrogram of similar blogs for the targeted dataset.

NOTE: The above image causes LaTeX to crash every time, but it can be found at sblogclust.jpg. Sorry for the inconvenience.

As you can see, the clusters are in two groups. This means that the clustering algorithm sees inhearent differences between the two types of blog that I was searching for. This means that my data set was at least partly correct, and that this is a method that can at least spot overarching differences in data.

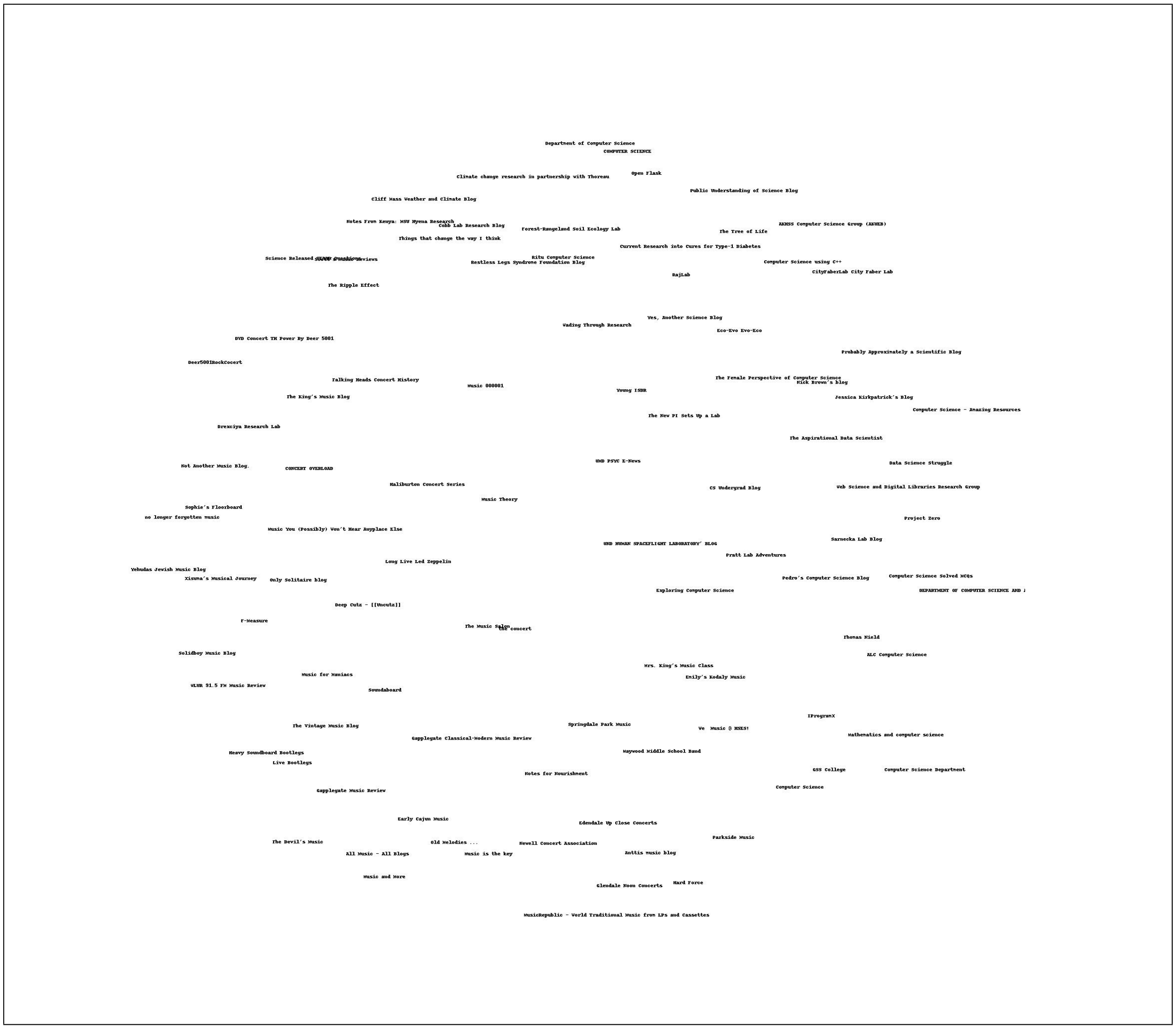
Problem 3

Once again, this can be found in stoo-big-for-latex.txt, as it is too big for this document. K=5 took four iterations, k=10 took 6 iterations, and k=20 took six iterations as well. The same slowdown occurred as in the real problem 3, where increasing values of K caused iterations to take longer.

Problem 4

This time, it only took 126 iterations to group the values. This is probably due to the targeted nature of the dataset producing less, or less complicated, clusters.

Image 4: Image of similar blogs arranged by distance for the targeted data.


Department of Computer Science
COMPUTER SCIENCE
Climate change research in partnership with Thoreau
Open Flask
Public Understanding of Science Blog
Cliff Mass Weather and Climate Blog
Notes From Kenya: MSU Hyena Research
Cobb Lab Research Blog
Forest-Rangeland Soil Ecology Lab
The Tree of Life
AKHSS Computer Science Group (AKUEB)
Things that change the way I think
Current Research into Cures for Type-1 Diabetes
Ritu Computer Science
Restless Legs Syndrome Foundation Blog
Computer Science using C++
CityFaberLab City Faber Lab
RajLab
The Ripple Effect
Yes, Another Science Blog
Wading Through Research
Eco-Evo Evo-Eco
DVD Concert TH Power By Deer 5001
Probably Approximately a Scientific Blog
Deer5001RockCocert
Talking Heads Concert History
Music 000001
The Female Perspective of Computer Science
Nick Brown's blog
Young ISDR
The King's Music Blog
Jessica Kirkpatrick's Blog
Computer Science - Amazing Resources
The New PI Sets Up a Lab
Drexciya Research Lab
The Aspirational Data Scientist
UMD PSYC E-News
Data Science Struggle
Not Another Music Blog.
CONCERT OVERLOAD
Haliburton Concert Series
CS Undergrad Blog
Web Science and Digital Libraries Research Group
Music Theory
Sophie's Floorboard
no longer forgotten music
Project Zero
Music You (Possibly) Won't Hear Anyplace Else
Sarnecka Lab Blog
UND HUMAN SPACEFLIGHT LABORATORY' BLOG
Pratt Lab Adventures
Long Live Led Zeppelin
Yehudas Jewish Music Blog
Xisuma's Musical Journey
Only Solitaire blog
Pedro's Computer Science Blog
Computer Science Solved MCQs
Exploring Computer Science
DEPARTMENT OF COMPUTER SCIENCE AND
Deep Cutz - [[Uncutz]]
F-Measure
The Music Salon
The concert
Thomas Nield
Solidboy Music Blog
ALC Computer Science
Mrs. King's Music Class
Music for Maniacs
Emily's Kodaly Music
WLUR 91.5 FM Music Review
Soundaboard
IProgramX
The Vintage Music Blog
Springdale Park Music
We Music @ HSES!
Mathematics and computer science
Gapplegate Classical-Modern Music Review
Heavy Soundboard Bootlegs
Maywood Middle School Band
Live Bootlegs
GSS College
Computer Science Department
Notes for Nourishment
Computer Science
Gapplegate Music Review
Early Cajun Music
Edendale Up Close Concerts
Parkside Music
The Devil's Music
Old Melodies ...
Newell Concert Association
All Music - All Blogs
Music is the key
Anttis music blog
Music and More
Glendale Noon Concerts
Hard Force
MusicRepublic - World Traditional Music from LPs and Cassettes