CS595 Assignment 9

Jon Robison

November 24, 2013

Q1.
Create a blog-term matrix. Start by grabbing 100 blogs.

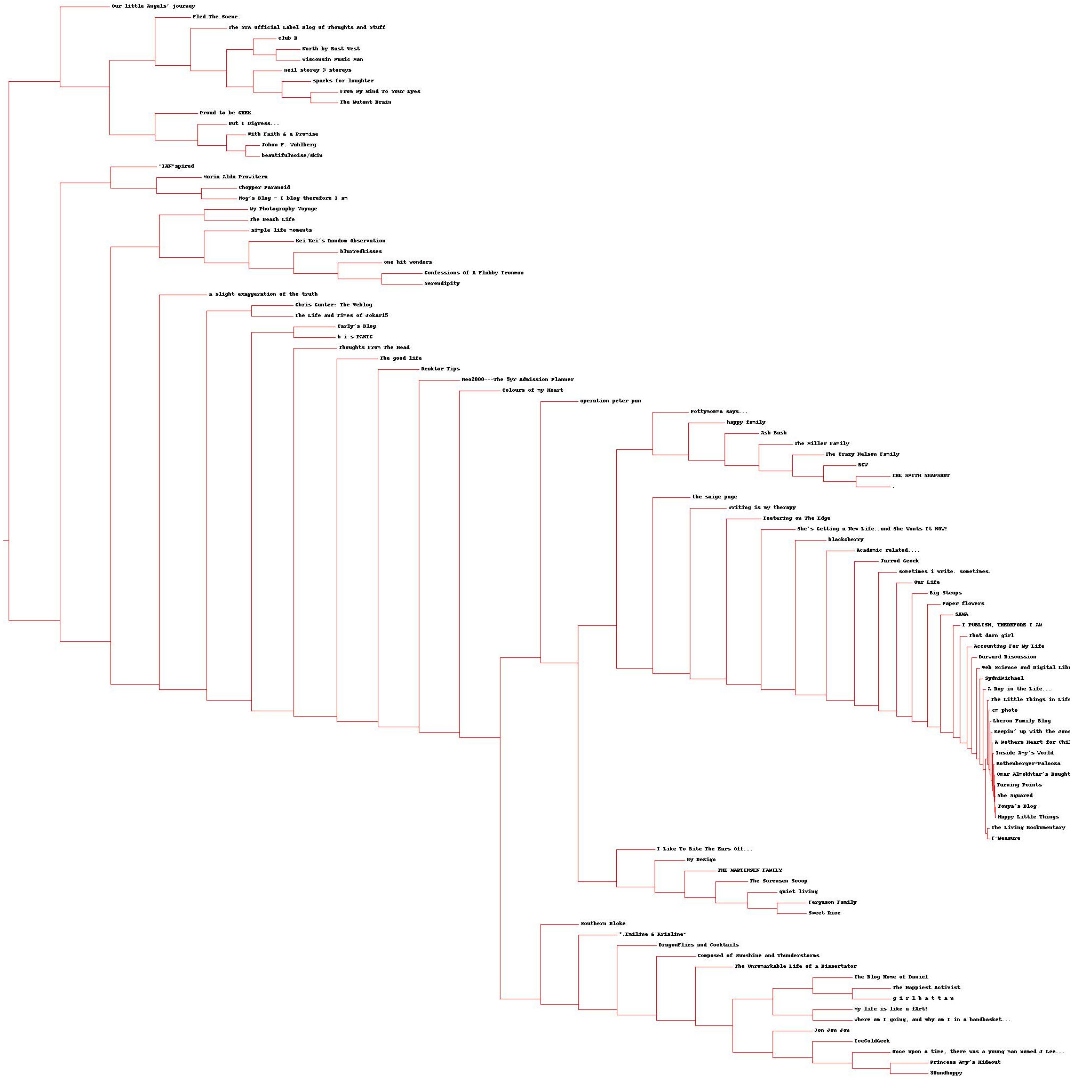
After writing a program to grab some given number of blogs, the given generatefeedvector provided the matrix. Another program, generateMatrix.py, was written to accomplish the same task, however, it was noted on the slides in a sneaky location that code was provided accomplishing the same thing. Much frustration at self ensued.
See Appendix A for program to generate bloglist, generateUrls.py

Q2.
Create an ASCII and JPEG dendrogram that clusters (i.e., HAC) the most similar blogs.

The first step is always making sure this is believable to the person assigning grades, and sure enough, F-Measure is beside The Living Rockumentary. My grade for this assignment can go up to 11, you know.

Similar blogs are grouped together, most notably, technology blogs and meta-type blogs. Meta type blogs as used here are blogs about either blogging, commentary (on life, ideals and culture), the meaninglessness of commentary, or the meaninglessness of blogging. I found these to be most interesting, as often the bloggers portray themselves as outsiders looking in to a culture they appreciate, don't understand, don't connect with, are mad at, however, their proximity in the dendrogram indicates they are at least similar to each other. This implies there is a large subculture of people thinking they are unique as an inherently positive attribute. Notable that it is (almost) the majority group of this dataset, ultra large sample size that it is (/hyperbole).

Figure 1: Dendrogram



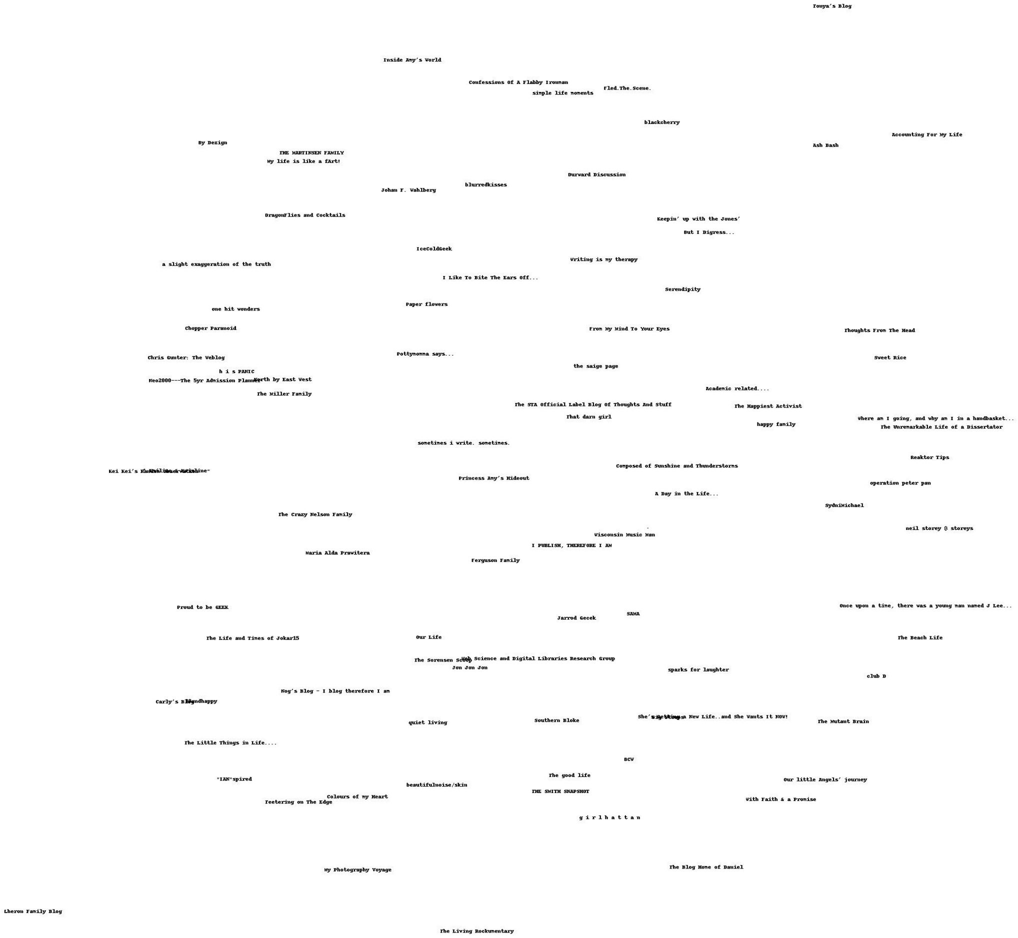
Q3.
Cluster the blogs using K-Means, using k=5,10,20. (see slide 18).
How many interations were required for each value of k?

K	Iterations
5	5
10	7
20	7

Q4.
Use MDS to create a JPEG of the blogs similar to slide 29.
How many iterations were required?

Only one iteration before error started increasing by a factor of ten. TODO This is odd and I mean to go back to see why, haven't yet.

Figure 2: Clustered Blogs



Appendix A

```
#!/usr/bin/python3
import sys
from bs4 import BeautifulSoup
import urllib.request
from urllib.parse import urlparse

DEFAULT_COUNT=98
DEFAULT_SEED_URL='http://www.blogger.com/next-blog?navBar=true&blogID=347163
if len(sys.argv) != 3:
    print('Pass the blog count, defaulting to ' + str(DEFAULT_COUNT))
    print('Pass the seed URL, defaulting to ' + DEFAULT_SEED_URL)
    count=DEFAULT_COUNT
    url=DEFAULT_SEED_URL
else:
    count=sys.argv[1]
    url=sys.argv[2]

def parse(link):
    response = urllib.request.urlopen(link)
    soup = BeautifulSoup(response.read())
    response.close()
    return soup

def addNext(url,s):
    try:
        soup=parse(url)
        for atom in soup.findAll('link',rel='alternate',type='application/at
            atomHref=str(atom['href']).strip()
            s.add(atomHref)
            print('Added atom href: ' + atomHref)
    except:
        print('Exception parsing URL, skipping to next')
        pass

s=set()
while len(s) < count:
    addNext(url,s)
with open('feedlist.txt', 'w') as f:
    for atom in s:
        f.write(atom + '\n')
```