

HW5 Report: Mining text from the web

Seong Hyeok Im, Inseong Joe, Dongyoung Kang

March 2, 2014

1 Project Overview

For this project, we have decided to analyze texts in 3 college websites, including Olin, Babson, and Wellesley. As we run some computational analysis on it, we aim to relate the each college's traits to texts that appear in its websites. For instance, Olin is an engineering school so, hopefully, Olin college website would contain texts related to engineering more than other college websites. The general approach of our project is dividing our task into two parts. Simply put, we will first extract texts and then analyze those extracted data. Result of our project will appear in form of list of words that appear most frequently and we will be able to determine the number of words with top frequency that we want to seek.

2 Implementation

We have divided the coding work into two parts. The first part is the code that extracts texts from webs and the second part is the code that analyzes those extracted texts.

2.1 Part I: Extracting text from webs

Python code "crawler.py" is that first part which crawls webpages from given url. To summarize what the script does, it fetches webpage data from given url and then extract link urls from fetched webpage data. Next, it standardize all urls and remove duplicates. Finally, it fetches webpage data from all standardized urls and save to a file. While this is brief overview of what "crawler.py" does, we specified the roles of each functions contained in this script to make it successfully perform its task. Basically, this codes can be divided into two group of functions. The first group of functions are URL functions. "fetch link urls" fetches all link urls from given url. Followingly, those urls are converted to the specifically defined form of url. Besides URL functions, main functions, as you can tell by how it is called, does main work of extracting texts from web. The function "do crawl" in this group of function is the main crawling function responsible for bringing data from url and saving those data into files.

2.2 Part II: Analyzing extracted texts

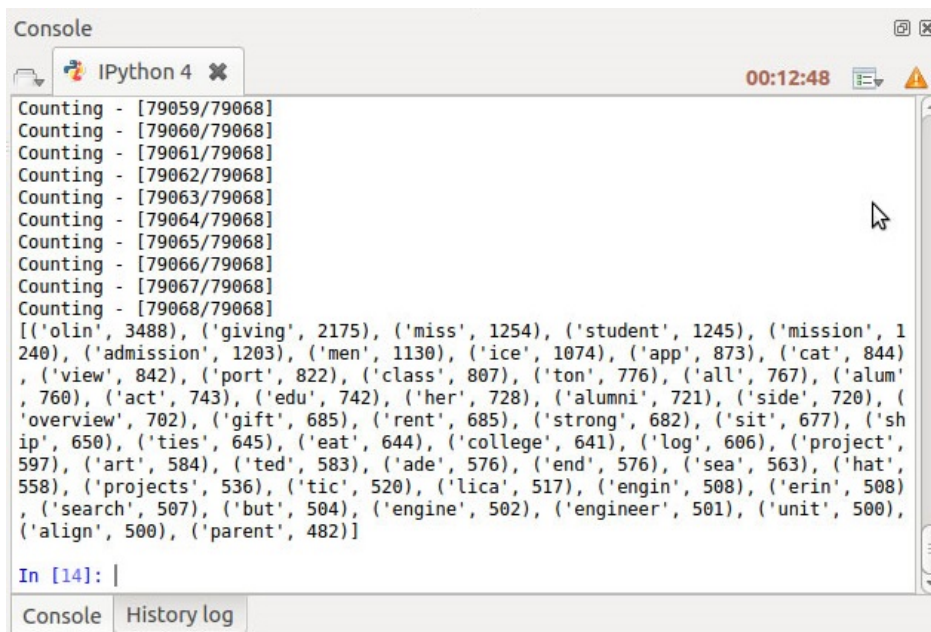
This part of our work is mainly, actually exclusively, concerned with our code "testhw5.py". Easily put, this code analyzes extracted texts in terms of the number of times they appear. In other words, it measures the frequency of each different words. In order to make this piece of code do what it should be doing, we had to define several functions that would filter out the words in more digestible format. For instance, we eliminated the difference between word containing the uppercase letter and word containing the lower case letter. For instance, consider the word "And"

and "and". Although they are the same, this code might treat them differently since they are "technically" different. Therefore, we devised the function "turn text to lower case" that converts all letters into lower case letters. Remember that our project is focused on analyzing texts in terms of the number of times, frequency, in certain url. To make our project all the more meaningful, we made sure that we ignore words that appear quite often but lack in significance. That is the reason why the list "ignoringwords" appears in the beginning of this code.

3 Results

As our script takes "n", the number of words as part of its input, we can derive list of words containing "n" words according to frequency. Following are the results and descriptions of results that we attained using our code. We returned the top 50 frequently words used in each website. Note that due to the difficulties of perfectly extracting only visible text and eliminating all html coding, the text file we extracted was not perfect. Therefore, some words were cut off and some words resembled html language. So, we focused more on the overall trend of the words not the actual rankings.

3.1 Olin



```

Console
IPython 4
00:12:48

Counting - [79059/79068]
Counting - [79060/79068]
Counting - [79061/79068]
Counting - [79062/79068]
Counting - [79063/79068]
Counting - [79064/79068]
Counting - [79065/79068]
Counting - [79066/79068]
Counting - [79067/79068]
Counting - [79068/79068]
[('olin', 3488), ('giving', 2175), ('miss', 1254), ('student', 1245), ('mission', 1
240), ('admission', 1203), ('men', 1130), ('ice', 1074), ('app', 873), ('cat', 844)
, ('view', 842), ('port', 822), ('class', 807), ('ton', 776), ('all', 767), ('alum
', 760), ('act', 743), ('edu', 742), ('her', 728), ('alumni', 721), ('side', 720), ('
overview', 702), ('gift', 685), ('rent', 685), ('strong', 682), ('sit', 677), ('sh
ip', 650), ('ties', 645), ('eat', 644), ('college', 641), ('log', 606), ('project'
, 597), ('art', 584), ('ted', 583), ('ade', 576), ('end', 576), ('sea', 563), ('hat'
, 558), ('projects', 536), ('tic', 520), ('lica', 517), ('engin', 508), ('erin', 508)
, ('search', 507), ('but', 504), ('engine', 502), ('engineer', 501), ('unit', 500)
, ('align', 500), ('parent', 482)]

In [14]: |

Console History log

```

Figure 1: Result for top 50 frequently used words in olin.edu

You might as well have expected but you can see that the word "Olin" appears most in our website. This result is not surprising at all nor is it very meaningful but we can see that our code is working properly. Also, there can be seen many words like "engine", "engineering", "engineer", "erin" which we assume that all come from the word engineering as some may have been cut off. Also, words like "miss" or "mission" all add to "admission". The word "project" also occurs frequently, indicating that our college is heavily related to engineering. The interesting thing is that the word "giving" appears most frequently only next to the word "Olin". We also assume that words like "ice", "ties", "ted", "ade" and so on are all parts of other words.

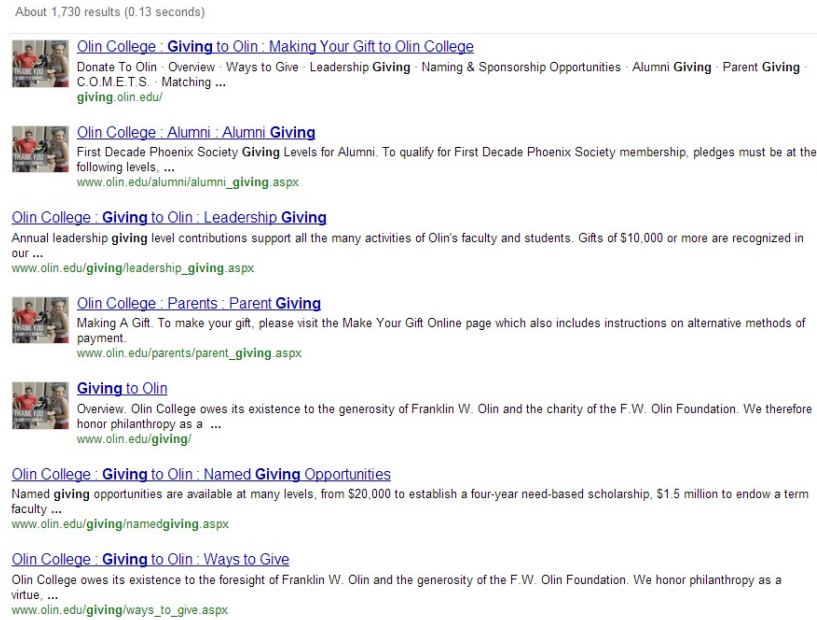


Figure 2: Searching one of the top words "giving" from olin.edu in google

We were curious in which context this word "giving" appears so often. As is turned out, "giving" appeared most in the "giving to olin" section. This might be a little bit of a stretch but we were able to assume that our college either really necessitates donation or appreciated donation from others.

3.2 Wellesley

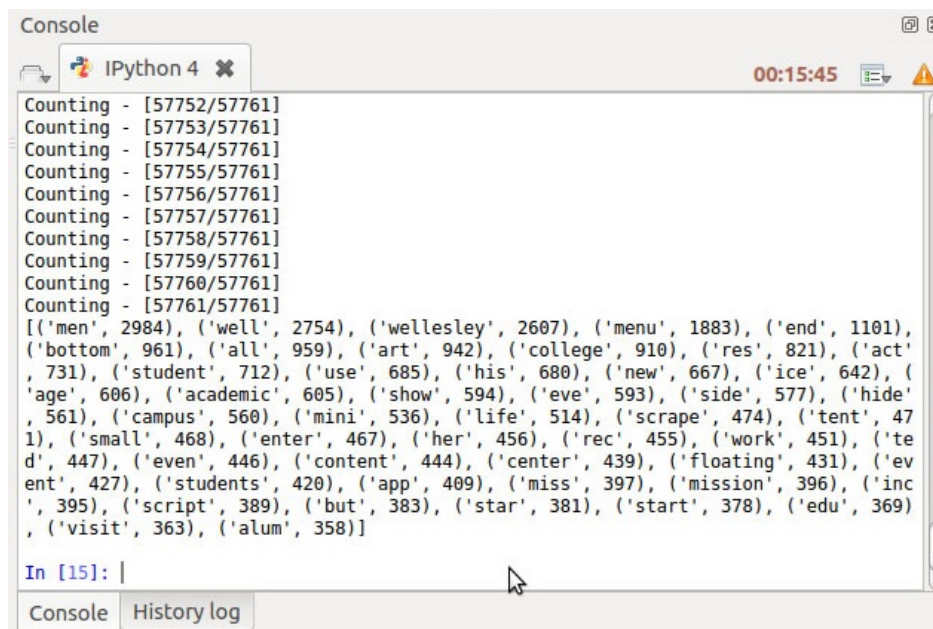
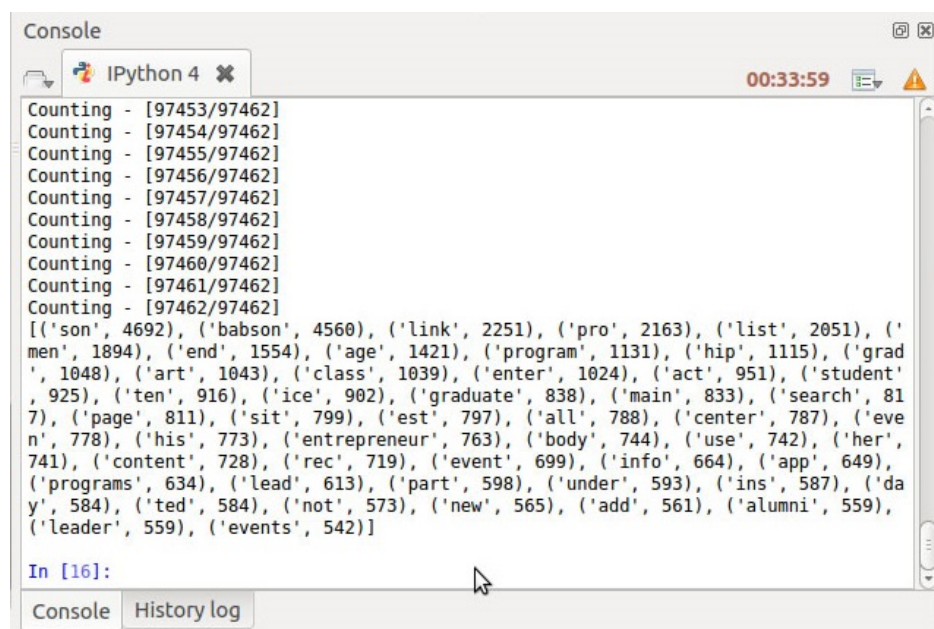


Figure 3: Result for top 50 frequently used words in wellesley.edu

At first when we saw the output for wellesley.edu, we were shocked because the most frequently used word was "men". But then, after debugging our code and looking out for some limitations, we found out that this "men" would most likely be part of the word "women" as it was somehow separated into "wo" and "men" in the process of turning html into plain text. We also see that the word "well" should be part of "wellesley". Other than these errors, we see that the overall trend is women, art, and factual information like campus, academic, admission and etc.

3.3 Babson



```

Console
IPython 4
00:33:59

Counting - [97453/97462]
Counting - [97454/97462]
Counting - [97455/97462]
Counting - [97456/97462]
Counting - [97457/97462]
Counting - [97458/97462]
Counting - [97459/97462]
Counting - [97460/97462]
Counting - [97461/97462]
Counting - [97462/97462]
[('son', 4692), ('babson', 4560), ('link', 2251), ('pro', 2163), ('list', 2051), ('men', 1894), ('end', 1554), ('age', 1421), ('program', 1131), ('hip', 1115), ('grad', 1048), ('art', 1043), ('class', 1039), ('enter', 1024), ('act', 951), ('student', 925), ('ten', 916), ('ice', 902), ('graduate', 838), ('main', 833), ('search', 817), ('page', 811), ('sit', 799), ('est', 797), ('all', 788), ('center', 787), ('even', 778), ('his', 773), ('entrepreneur', 763), ('body', 744), ('use', 742), ('her', 741), ('content', 728), ('rec', 719), ('event', 699), ('info', 664), ('app', 649), ('programs', 634), ('lead', 613), ('part', 598), ('under', 593), ('ins', 587), ('day', 584), ('ted', 584), ('not', 573), ('new', 565), ('add', 561), ('alumni', 559), ('leader', 559), ('events', 542)]

In [16]:
Console History log

```

Figure 4: Result for top 50 frequently used words in babson.edu

Last but not least, the result for babson college also seemed to represent some key traits of this college. We were able to figure this out by looking at the word "entrepreneurship". While this also showed the same errors of words getting split up, we could still see that the overall trend of words showed "entrepreneurship". Like Olin and Wellesley, the most frequently used word was "Babson" and some other factual information leading words like "class", "student" and so on.

Overall, as a result we found that like we had expected each school showed their distinct characteristics in their websites. However, because for this homework we only extracted the links that were reachable from the main page, most words showed factual information of the college.

4 Reflection

4.1 what went well

In retrospect, we were happy about the appropriate scope of our project. We were not completely sure about whether we would be able to handle our project when we first established our goal of analyzing texts in 3 college websites. As it turned out, however, our coding was still doable in

a sense that it was not like a hot potato that makes us inhibited from even getting our hands on it and at the same time, it was not too simple a task for us to finish. Additionally, we were also happy about our strategy towards approaching this project: works were clearly divided in reasonable portion and those works were done according to initially desired timeline.

4.2 rooms for improvement

What we have done so far is creating a script that explores texts and output the most frequently appearing words. There are many ways to improve our work but to mention some specific rooms for improvement, they are as follows:

- extract domain automatically
- enhance url standardization method
- visit websites recursively by setting depth of visit and checking already visited webpage
- crawl pages using multi thread for better performance