Facebook URL Frequency Analysis

Franton Lin

Project Overview

For this project, I used the Python Pattern web API to grab data from Facebook. I gave Pattern access to my Facebook account, and scanned my friends' timelines previous 100 posts for URLs, storing frequency and cumulative like data in a Python dictionary. The program stores this data via pickling using the Python cPickle API. Then the program parses the URLs using regular expressions, grouping the cumulative likes and post counter to common URL bases (i.e. youtube.com). I hoped to learn what URLs got the most likes and are posted the most often within my Facebook friend group.

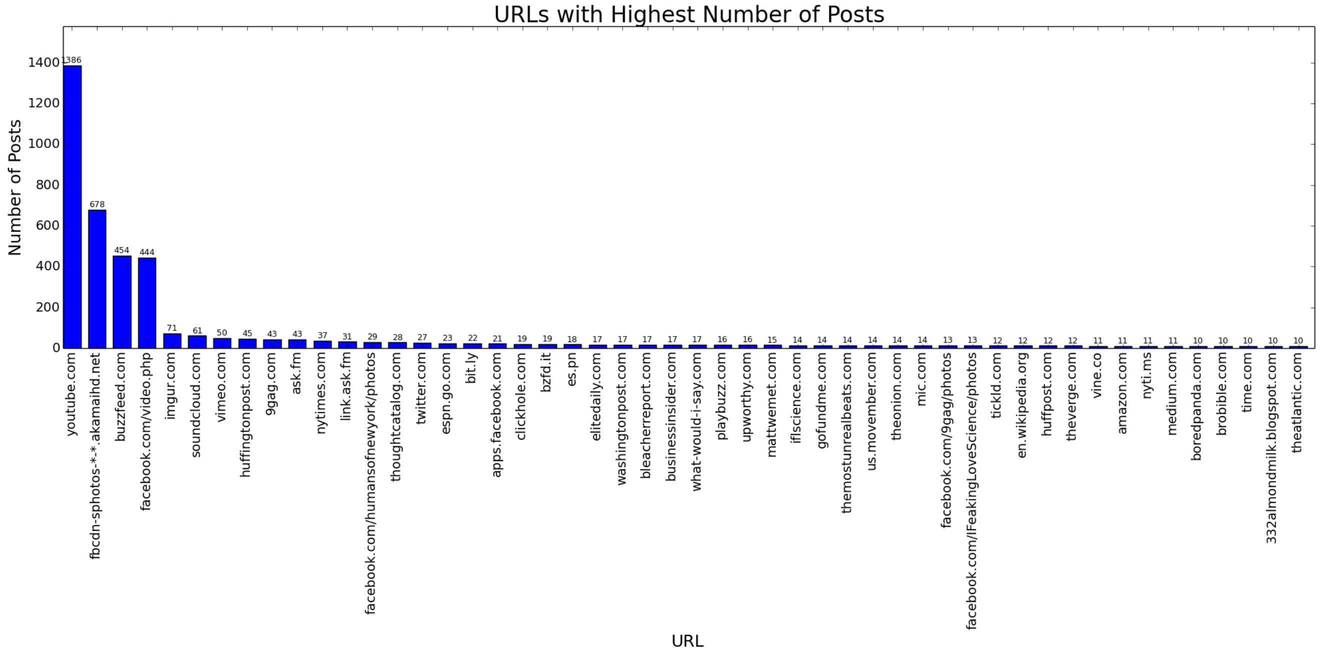
Implementation

To fetch the data, Pattern crawls up to 1000 of my Facebook friends' timelines, searching up to 100 previous posts for URLs. Data on posts with URLs are stored in a Python dictionary with the key as the URL and the value as a list containing a frequency counter and the cumulative likes for the given URL key. I chose to store the data in a dictionary because any information (such as the frequency) for each URL can be accessed and modified very easily by using the URL as the key. I also implemented pickling using the Python cPickle library so I would not have to crawl Facebook for data every time I wanted to analyze the data differently.

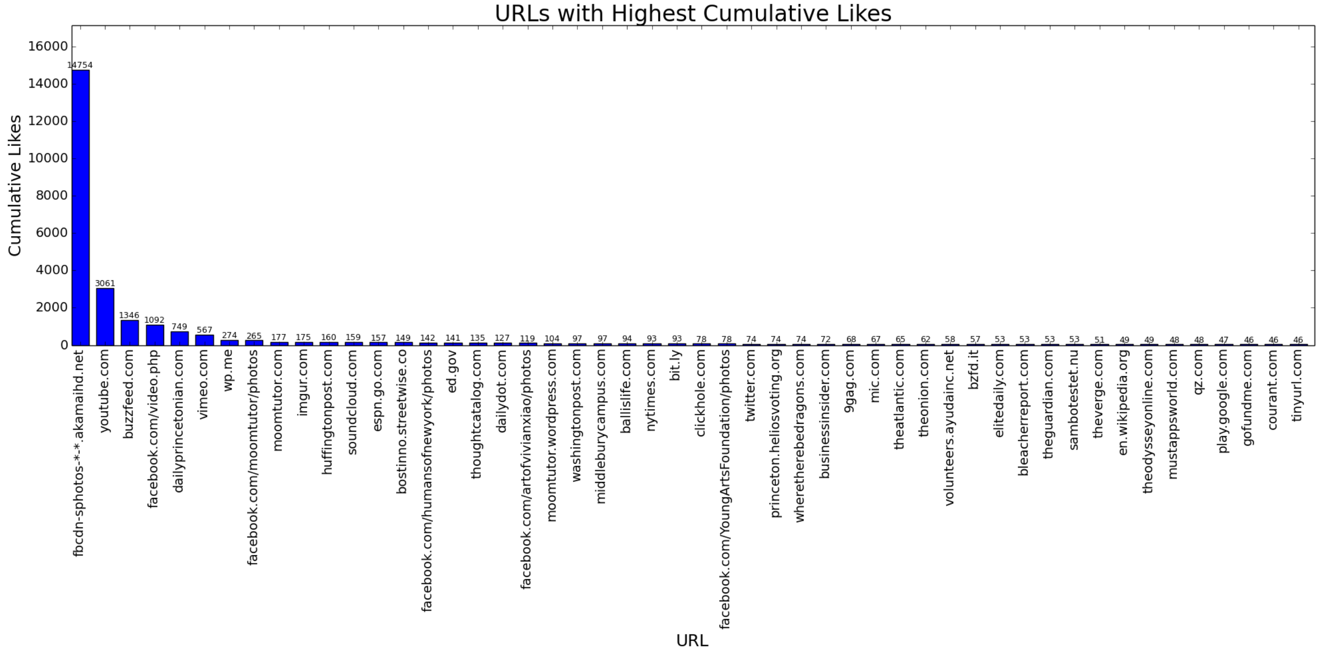
To process the data, I went through the pickled data and grouped URLs together if they were associated with the same base website (i.e. youtube.com), combining their frequency counters and cumulative like values. The processed data is stored in another dictionary, for the same reasons as above. The processing contains a lot of if statements to deal the the many different cases of URLs. I also used regular expressions to parse the URLs. I then copy the processed data to a list of tuples, which I sorted in various ways and plotted using the matplotlib API.

Results

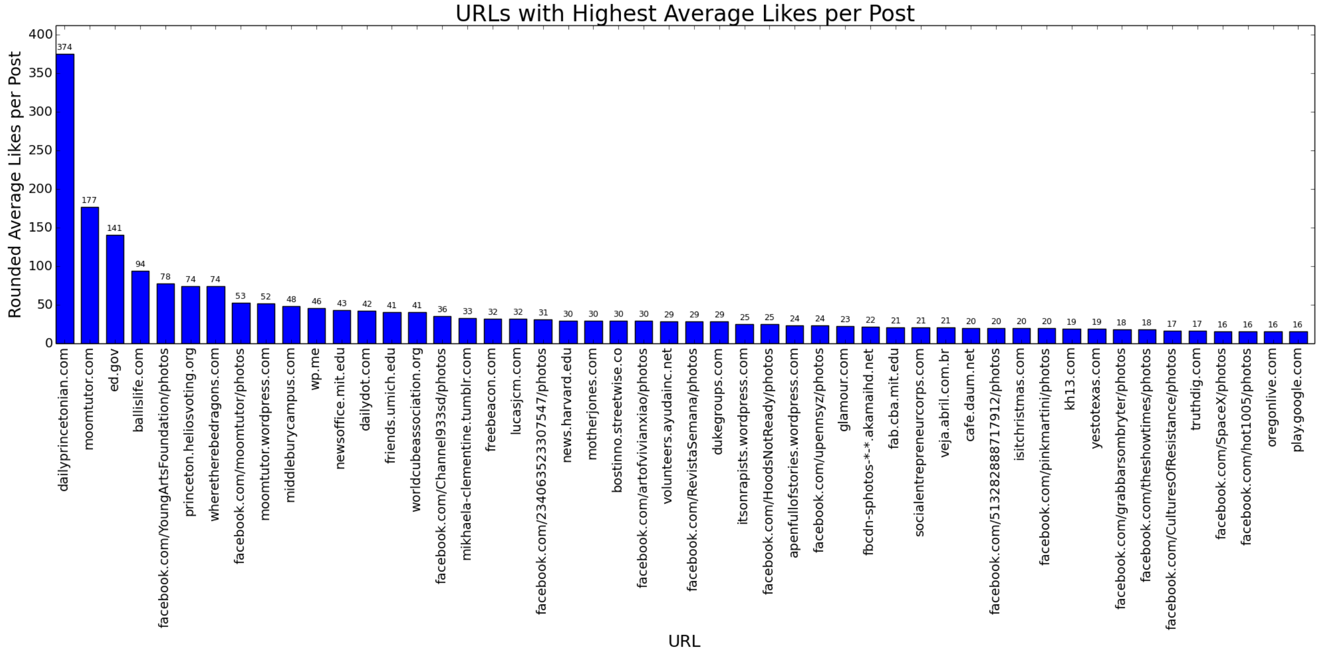
To visualize the results, I plotted bar charts of the URLs with the top number of posts, highest cumulative likes, and highest average likes per post. Pattern looked at a total of 11342 posts, 6692 of which contained links (59%). The URL fbcdn-sphotos-*-*.akamaihd.net is a generalized URL representing all of people's own Facebook-hosted photos that are on their timelines (cover photos, added photos to album, etc.). The URL facebook.com/video.php represents people's own Facebook-hosted videos that are on their timelines.



We can see that youtube.com is the most commonly posted URL by a very large margin.



Facebook-hosted photos have the highest number of cumulative likes by an extrodinarilly large margin.



A few random websites that aren't posted very much but get lots of likes are at the top of this list. It appears that groups'/pages' photos very often fall into this category. Most of the URLs appear to be related to companies or colleges. However, the Facebook-hosted generalized user photo URL still appears in the top 50, as well as being very near the top of the other two lists. If you want Facebook likes, post some good photos.

Results

Figuring out how to use the Pattern web API with Facebook was a little bit difficult and time consuming, given the lack of easy to find documentation. Learning cPickle, regular expressions, and matplotlib also took a decent amount of time. It turns out that even Pattern does not accurately return information about Facebook posts, and dealing with all of the different URL cases was frustrating at times. Overall, the project was a bit large considering the time constraint. I wish I had known about the difficulties of dealing with Facebook's complexity and the vast range of URL formats beforehand and not combined them in one project. I definitely will use what I learned about matplotlib and regular expressions in future projects.