Ian Paul

Project overview:

For this project I focus on analyzing data from GitHub. Primarily I looked at the comments on pull requests. My dataset consisted of one gigabyte of raw data from GitHub, containing the pull requests and other information. I analyzed this data by looking at word frequencies in the text of the comments. This analysis led to a graph of word frequencies and a list of the most common words. Initially I hoped to create a program to do Markov text synthesis, but do to the size of my data and time constraints I just analyzed word frequencies.

My primary source for this project was GitHub. I began by querying GitHub's API but in order to get a more extensive set of data I used GHTorrent's database. I tried querying their web databases for parts of the data, but I ended up downloading their complete set of GitHub transactions. The most accessible part was their pull request comments, which I've analyzed for this project. I primarily looked at word frequencies for analysis because of the size of this project. I hoped to learn something about what people

Implementation:

In order to get my dataset I began by exploring the GitHub API. From this I learned the type of data I could get out of GitHub, but in order to get a more extensive set of data I used GHTorrent to download a partial set of their archive of GitHub public events. This set of data was incomplete (the full set of data is 4 TB of compressed JSON data), so I used the most complete set I had: pull request comments.

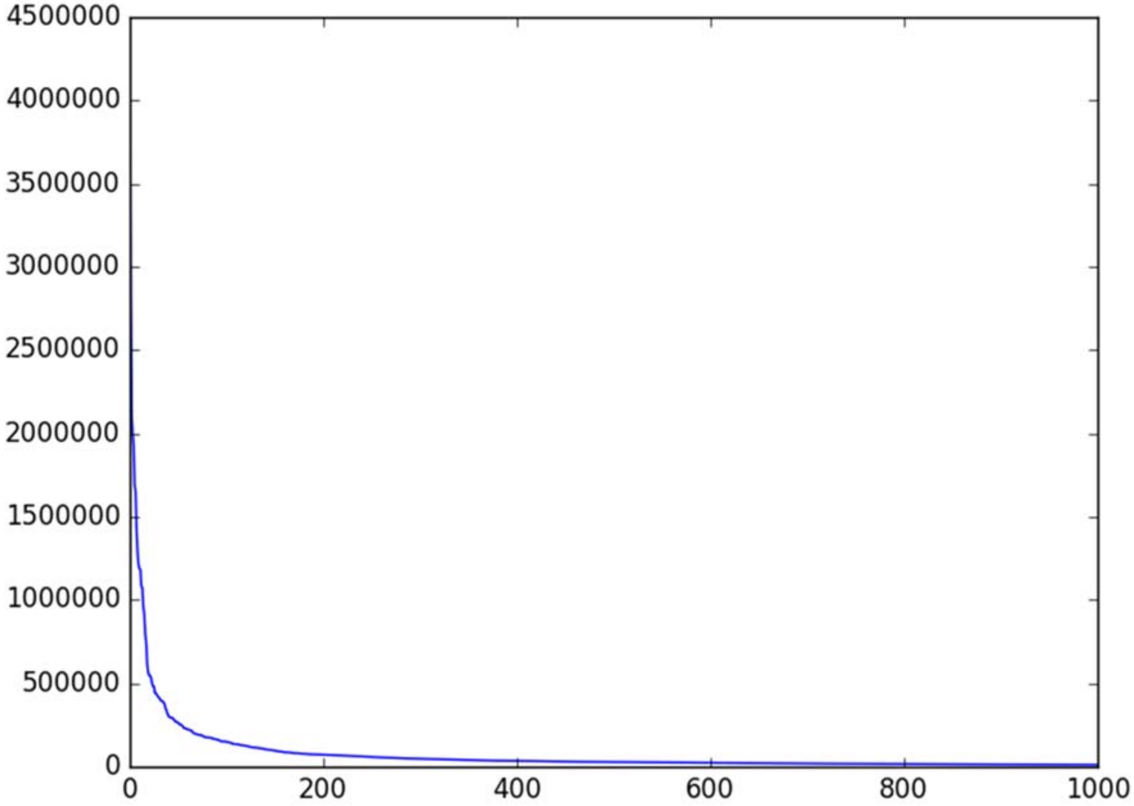
This initially started out as a gigabyte of raw data in a CSV. In order to make this data more accessible I worked to reduce the clutter and size of the data. By removing everything that wasn't text, I reduced it to 600 megabytes, and by reducing it to a dictionary of word frequencies I brought it below 100 megabytes. This did cause me to lose the ability to do Markov chain analysis, but I decided that the speed gained from being able to operate on the dictionary was worth the loss of information. Given more time I would still be interested in generating average GitHub pull request comments using Markov text synthesis, but for this project I decided that being able to handle data quickly was more valuable.

Using a dictionary seemed to pay off because to increase the speed of my code I ended up doing most of my analysis on the 1000 most common words in the dataset. Having the complete list is interesting, but oftentimes the sheer volume of it would obscure any meaning. For example, the graph of word frequencies looks like a right angle along the axis when done with a larger dataset.

Results:

At first glance the GitHub data seems largely uninteresting, but that is because so many common English words are still included. Two ways to get around that are to either to analyze the data for general trends or to remove the common words.

An interesting way to look at the general trends in a list of word frequencies is to graph it. Here I've graphed the frequency of words on the y axis versus their rank (from most common to least) on the x axis. This produces a graph similar to y = 1/x, which is the same as what a graph of word common in general English would resemble. The graph here is slightly steeper – the word the occurs more than twice as often as the second most common word in GitHub, compared to slightly less than twice as often [Wiktionary]. The most common words also differ in order between GitHub and English, but that's not captured by the graph. One other thing gathered from the graph is that some words are grouped. Manual analysis shows that this is often 'words' like 'https' and 'com', seeing as they are so frequently in urls.



Analyzing the data by removing 127 of the most common words in English starts to show some other interesting features of the data. In the list of the 50 most common uncommon words, words unique to computers and coding start to come up, as well as related words. 'Code and 'GitHub' are pretty obviously related to coding, as well as many of the other words having strong coding implications.

The 50 most common words beyond the common ones are :use, think, need, would, like, line, don't, it's, code, 1, one, com, could, also, add, test, make, github, https, change, method, name, file, instead, get, see, want, using, i'm, function, remove, new, used, sure, case, 0, good, comment, 2, since, way, please, class, something, better, right, check, return, maybe, doesn't,

Reflection:

I think I did a good job analyzing the data set I had as a whole. If I could go back and redo it, I would settle for a smaller data set and try to do the Markov analysis of the text. I would also probably pick commit messages instead off pull request comments. I think the overall analysis went well and I’m particularly proud of how much of the data I could cut down and how well I could condense it. Unit testing is also something I’m proud of. While not captured in my code, I ran unit tests by building toy datasets that were in the same format as what I was trying to analyze , but smaller, so that I could test quickly.