MP3 Data Mining

Project Overview [Maximum 100 words]

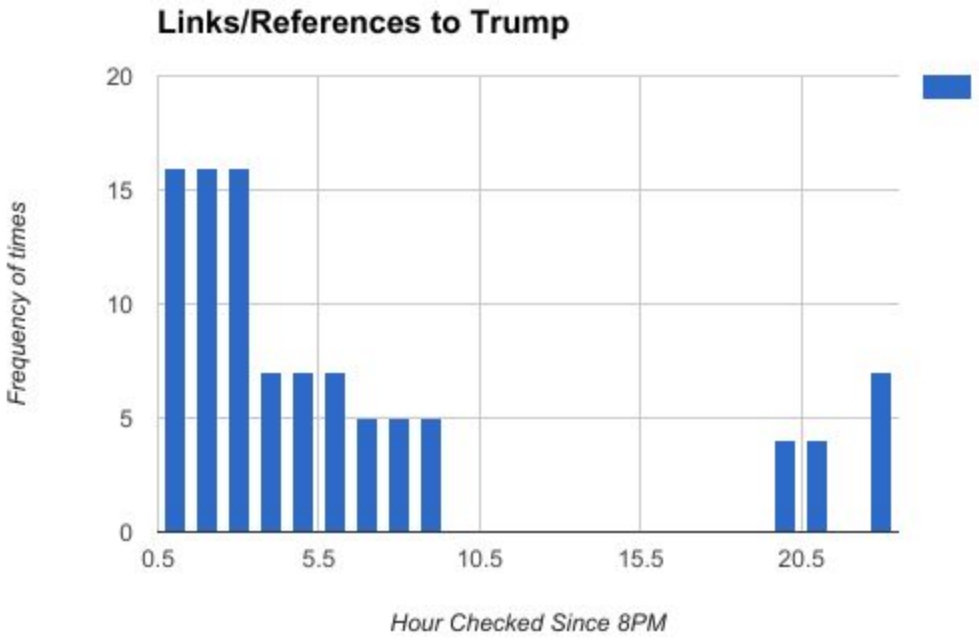
In this project, I used cnn.com and analyzed their webpage for the word frequencies using a simple search. I extracted text using the beautiful soup. I hope to learn about accessing websites using my code.

Implementation [~2-3 paragraphs]

My implementation involved all the main steps. I wanted to extract a website from the internet and take its html code and pickle it. Then with the pickled file, I analyzed the file for the frequency of certain words. I then used this code and kept track of the data as time went on to get a bar graph of the fluctuations of the mentions of Trump throughout the day

One of the design decisions I made was to keep things simple, so I used the beautifulsoup parser/html interpreter. I found this package to be very useful in my program, even though I did not use the package to its full potential. I only used the get.text function of the beautiful soup, and parsed through that text file using their parser.

Results [~2-3 paragraphs + figures/examples]



After checking for the word frequencies, there is actually a lot more data that can be found from checking for mentions of “Trump” other than the frequency of the name. Looking at the figure, there is an overall consistency in news about Trump. The period of gaps was when I went to sleep and was unable to check the program.

Another Interesting thing that I found was that by looking at how often I updated my program and called the html code from the CNN website, I found that CNN generally updates their news every three hours.

Reflection [~1 paragraph]

I think that the project went pretty well. I could improve by widening my range of techniques to analyze data. One of the interesting things was that I put a lot of trust into the beautifulsoup parser and get text commands, and perhaps it would have been best to check over that. My program was simple and had good places to test the loops and if the html was being pickled correctly, and I hope to expand my knowledge on html parsers and interpreters for the future.