Zoher Ghadyali and Cynthia Chen
Software Design 2014
Homework 5: Mining Text from the Web

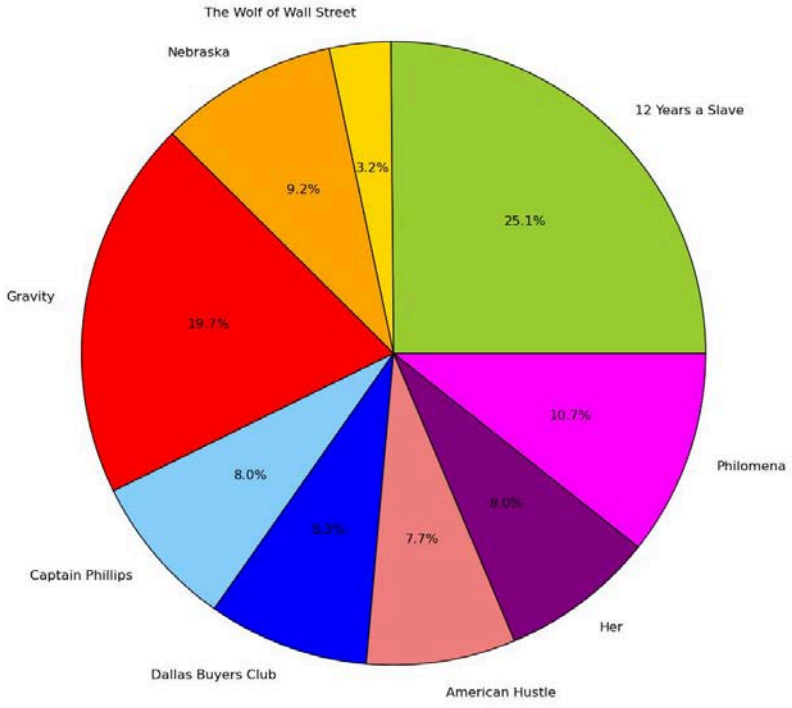
PROJECT OVERVIEW: Our goal in this project was to produce an Oscar Predictor by scanning relevant tweets using Python's Pattern module. We wanted our final output to be a pie chart that predicts the probability of each movie, actor, actress, and director winning the Oscars. Our general approach was to loop through all of our gathered tweets, searching for the relevant information, and then performing sentiment analysis on the relevant tweets.

IMPLEMENTATION: Our first goal in this project was to gather data from Twitter. We decided that we did not want to repeatedly mine the data as this would increase the run time for our program. We initialized a dictionary with keys that were the movies, actors, actresses, and directors which corresponded to words such as "Oscars", "will", and "win". We searched through Twitter for these relevant tweets and, for each category, we stored them as plaintext files.

We then opened each file as a list and looped through each tweet, searching for the relevant information. We found all of the indices of the relevant tweets and using these indices, our next goal was to do sentiment analysis on each relevant tweet. We stored all of these sentiments in a list and, looping through the list, we found all of the positive/negative sentiment values. We only paid attention to these values because we did not care about the subjectivity of each tweet.

Finally, we summed up all of the positive/negative sentiment values for each movie, actor, actress, or director. We then normalized these sentiments by adding up all of the total sentiments for each category and dividing the total sentiment of each movie, actor, actress, or director by this sum. Using the matplotlib module in Python, which allows us visualize data in much the same way as we do in MatLab, we produced a pie chart of these normalized sentiments, which correspond to the probabilities of each movie, actor, actress, or director winning their respective category in the Oscars.

RESULTS:



Above is our pie chart that predicts who will win the Oscar for Best Picture. This pie chart predicts that the Oscar will go to 12 Years a Slave but there is also a high chance that Gravity could win the Oscar. Overall, this pie chart seems to make sense but the probabilities of Philomena and Nebraska winning the Oscars seems overestimated. This may because in our total dataset of 104 tweets, Philomena and Nebraska were mentioned only a few times but overall positively which meant that the normalized data attributed too much significance to those few tweets. We also tried using the average total sentiment but we found that greatly overestimated the probabilities of Philomena and Nebraska winning this Oscar. We produced these pie charts for Best Picture, Best Actor, Best Actress, and Best Director and running our program will produce these charts.

REFLECTION: We feel that we worked well together. One thing that I think contributed to this is that we always worked together. This meant that we probably spent a greater time on this project than we would have individually but we made sure that the other always understood why we were writing the functions that we were writing. This means that you could ask either of us questions about the code and we would both be able to answer.

In terms of using Github and version control, we wrote all of our code on one laptop so we repeatedly pushed all of our code to one repository that we set up on Github. Consistently working together at the same time was somewhat of a double-edged sword. We definitely worked for longer but we also understood and learned more from each other. In terms of improvement, I think we spent a lot less time thinking about what would be the most efficient, best coding practices over what would work so our code is redundant and repetitive at times, while still being functional. We reached the constraints of for loops where we would loop through each tweet but needed to perform a different action for each different movie.

In terms of debugging and unit testing, we primarily used print statements to see what was happening within our dataset. With 104 tweets, print statements were doable but if we increases the scope of our project, it would no longer be realistic or efficient. Having two people working on the same screen also helped with debugging because, while the one partner wrote code, the other helped debug and watched for semantics and logic errors.

I think we chose a realistic and possible project and we will find out tonight the legitimacy of our predictions.