NLP Project 1 Report

Justin Postigo and Logan Williams

Exercise 1

For predicting the binary ratings of each paragraph, we used NLTK's Vader sentiment intensity analyzer. This gives us a percentage for how negative and positive the given text is, which we use to predict the binary rating of each paragraph. If the sentiment ratings are more negative, we predict 0 for the paragraph; if the sentiment ratings are more positive, we predict a rating of 1. This method has been giving us an average of around 65% accuracy for predicting the binary ratings of each paragraph. We believe that the accuracy is fair considering that many paragraphs in the corpus that are not in the same order as their corresponding numerical ratings.

Exercise 2

We were interested in discovering which words were used more often in positive reviews as well as negative reviews. In order to do so, we split the reviews into two sets: good ratings (overall rating of 4 or 5) and bad ratings (overall rating of 1, 2, or 3). We then calculated the FreqDist on all of the words, not including the stopwords. Then we took the difference of the frequency of words occurring in both good and bad sets. For example, if "yummy" showed up 20 times in good reviews and 5 times in bad reviews, we would adjust the frequency for "yummy" in the good reviews to 15 and 0 for bad reviews. Therefore, we have an updated list of word frequencies reflecting the comparative occurrences in good ratings and bad ratings. These are the three interesting things we discovered
as a result:

1) Reviews that have a positive overall rating of a restaurant tend to have more sentences than negative reviews. We discovered this by noticing that good reviews have consistently more periods ('.') than bad reviews (typically 100+ more). Probably not surprisingly, good reviews typically contained more occurrences of the word "good". Also, good reviews tend to contain more occurrences of "I", possibly indicating that reviewers write in the first person when they liked the restaurant.

2) Surprisingly to us, reviews that have a negative overall rating of a restaurant did not have as many distinguishing words as the positive ratings; the most frequent words usually had 10 or less occurrences, compared to the most frequent words in the positive reviews occurring around 30 times. The most notable recurring word used in negative reviews was "n't", probably used in phrases like "didn't like" or "wasn't good".

3) This method of distinguishing between reviews with good ratings and reviews with bad ratings proved to be beneficial in predicting the binary overall rating of reviews based on the text alone. To confirm this, we used the most common compared word frequencies for good and bad ratings as features to train a Naive Bayes classifier. When given

the tokenized words of a review, the classifier was able to correctly predict the overall rating of the review with an average accuracy of 64%. We found this interesting because it is nearly identical to the accuracy we achieve using NLTK's VADER sentiment analyzer.

Exercise 3
For predicting the overall rating of each review, we used NLTK's Naive Bayes classifier. The features we are using are the ratings of all of the other categories in the review: venue, food, and service. This has the classifier pick a score between 1-5, taking into account all of the other ratings so far. This method has been giving accuracies between 60% and 70%. We consider this to be a pretty high accuracy given our small amount of features.

Exercise 4
For predicting the authorship of each review we used NLTK's part-of-speech tagging. For every review, we get the 30 most common parts of speech used in that review, and label that with its author for the training set. For testing we wrote a function that finds the author who has the closest amount of similar part-of-speech tags. The closest author is then associated with that review for the test set. Our average accuracy for this is about 30%. We believe this to be fairly good given that part of speech plays a big role in author identification and that identifying authors is a very difficult thing to do.

Exercise 5
The confusion matrix was ran on 30 trials. For the confusion matrix, there were some people that got classified to themselves, which is good. For others, they don't even get classified to themselves and instead got classified to other people. There is an issue with some people's reviews that had improper HTML tags. If that was the case their review would show up "empty" and they would classify to one other person. So there are some people who are classifiying to one person 30 times. Most of the people were being classified to Vivian Fong, Alanna Buss, Nupur Garg, and Timothy Chu.

Logan Williams and Justin Postigo

	Gavin Scott	Tobias Bleisch	Vivian Fong	Alanna Buss
Gavin Scott	0	0	1	2
Tobias Bleisch	0	12	0	0
Vivian Fong	0	0	6	0
Alanna Buss	0	0	0	9
Ryan Smith	2	0	1	0
Logan Williams	0	0	0	0
Jonathan Sleep	0	0	8	3
Jon Doughty	0	0	2	0
Michael Williams	30	0	0	0
Brandon Cooper	0	0	1	6
Nicole Martin	30	0	0	0
Sean Bayley	0	0	0	0
Kishan Patel	0	0	1	0
Aditya Budhwar	0	0	0	0
Adam Calabrigo	0	0	0	0
Sage Maxwell	2	0	0	7
Justin Postigo	0	0	0	0
Ivan Pachev	0	0	3	0
Christian Durst	9	0	0	0
Brandon Livitski	0	0	0	3
Ryan Gelston	30	0	0	0
Joseph Wilson	0	0	0	2
Cody Hunt	0	0	0	10
Jeffrey McGovern	0	1	0	0
Nupur Garg	0	0	1	0
Daniel Kauffman	0	0	8	0
Samuel Lakes	0	0	1	0
Joel Dentici	0	0	0	11
Timothy Chu	0	0	0	0
Jeremy Kerfs	0	0	4	0
Miguel Aguilar	0	0	0	0

Ryan Smith	Logan Williams	Jonathan Sleep	Jon Doughty	Michael Williams	Brandon Cooper
7	0	0	0	3	0
0	0	0	0	0	0
0	0	4	0	0	0
0	0	3	0	0	0
6	0	2	2	0	0
0	0	0	3	0	0
0	0	3	0	0	0
0	5	1	2	0	0
0	0	0	0	0	0
1	0	0	0	0	13
0	0	0	0	0	0
0	0	0	2	0	0
0	0	0	1	0	0
0	1	3	0	0	2
0	1	0	0	0	3
0	0	3	0	0	0
0	0	0	0	0	0
0	1	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	8
0	6	0	0	0	0
0	2	0	0	0	0
3	1	8	0	0	0
0	0	1	2	0	0
0	0	0	0	0	0
3	0	0	3	0	0
0	2	0	8	0	0
0	4	0	0	0	0

Nicole Martin	Sean Bayley	Kishan Patel	Aditya Budhwar	Adam Calabrigo	Sage Maxwell
0	1	0	0	3	0
0	2	5	0	0	2
0	2	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	5	2	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	2	0
0	0	0	0	0	0
0	1	1	0	0	0
0	0	0	1	0	0
0	0	2	0	0	0
0	0	3	0	9	0
0	0	0	0	0	14
0	0	3	4	0	0
0	0	4	0	0	0
0	2	5	0	0	0
0	5	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	0	6	0
0	0	2	0	2	0
0	0	0	0	0	0
0	5	0	0	0	0
0	8	1	0	0	0
0	2	2	0	0	0
0	1	0	0	0	0
0	0	8	0	0	0

Justin Postigo	Ivan Pachev	Christian Durst	Brandon Livitski	Ryan Gelston	Joseph Wilson
0	4	2	0	0	2
0	0	0	0	0	0
0	9	0	0	0	0
0	0	4	0	12	0
0	0	1	0	0	3
0	4	0	0	0	0
0	0	0	2	0	9
0	0	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	6	0	0	0
0	1	5	0	0	0
6	0	2	0	0	0
2	1	0	0	0	0
0	0	1	0	0	0
8	0	0	0	0	0
0	0	1	0	0	0
0	0	6	0	0	0
0	0	1	12	0	0
0	0	0	0	0	0
0	0	0	0	28	0
0	0	0	9	0	0
0	6	0	0	0	0
0	4	0	0	0	0
0	2	0	0	0	0
0	0	10	1	0	0
0	0	0	0	0	3
0	0	0	0	0	0
0	0	2	0	0	0
4	0	0	0	0	0

Cody Hunt	Jeffrey McGovern	Nupur Garg	Daniel Kauffman	Samuel Lakes	Joel Dentici	Timothy Chu
0	0	0	0	0	0	5
3	5	0	0	0	0	0
0	0	5	0	0	0	4
0	0	0	0	2	0	0
0	0	3	0	1	8	0
0	4	6	0	0	0	0
3	0	0	2	0	0	0
0	0	2	1	5	2	1
0	0	0	0	0	0	0
6	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	6	7	5
0	0	5	0	2	5	3
0	0	11	0	2	0	0
7	0	4	0	0	0	0
0	0	0	0	0	2	1
0	0	7	0	0	0	0
0	2	12	5	0	0	0
0	0	0	0	8	0	0
1	0	0	0	0	2	5
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	6	0	0	0	0	0
0	3	12	2	0	0	0
0	0	0	6	0	0	0
0	0	0	0	5	0	3
1	0	0	0	0	0	6
1	0	0	0	0	4	15
0	0	1	0	1	0	2
0	2	2	0	0	0	0

Jeremy Kerfs	Miguel Aguilar
0	0
1	0
0	0
0	0
1	0
1	5
0	0
8	0
0	0
0	0
0	0
2	0
0	6
0	1
0	0
0	0
0	8
1	0
0	0
0	0
0	0
0	0
0	0
1	4
0	2
2	0
2	0
0	0
0	0
9	0
0	10