Restaurant Review Classification Using Naive Bayes

Andrew Guenther, Matt Parker, Matt Newsom

Our project involved using classmate generated restaurant reviews to train a classifier to predict 1-5 ratings on food, service, venue, and overall appeal, as well as predict the author of a given review. This is a topic of much study and focus inside the discipline of natural language processing. A quick search of Google Scholar for "restaurant sentiment analysis" yields over 33,000 results. Our particular task was made more difficult given the limited training data available from which to draw our conclusions. The data was also malformed with various HTML formatting, multiple reviews in one file, and misplaced paragraphs.

Exercise 1

Exercise 1 involves predicting the four paragraphs' 1-5 ratings based on a sentiment analysis of the paragraph. We trained a Naive Bayes classifier for this task, training it on primarily one feature: individual word sentiment scores. We would then classify each word in a test paragraph, and return the value of the most common word score. For example, if 40% of the words in a paragraph are rated 3, 20% are 4's, etc. then the paragraph would be given a rating of 3. We found this to be more accurate that taking an average of all the word scores. To determine sentiment, we utilized the SentiWordNet dictionary, which lists a positive and negative sentiment score for a large dictionary of words. Single word lookups took a very long time in the default SentiWordNet. In order to increase performance, any word with a neutral score was removed from the dictionary along with all synonym information. This made word lookups almost instant.

Due to the nature of the available training data, it seems that most feature sets we attempted to use with Naive Bayes resulted in the classifier outputting a score of 4 for nearly everything. This isn't too absurd, because it is actually fairly accurate, as the reviews tended to be filled with scores of 3, 4, and 5. With a classifier guessing 4 every time, we can achieve an average rms as low as 1.02 Other features we attempted to use involved average paragraph, sentence, and word length, bigrams, and word suffixes. None of these seemed to help or hurt the outcome much and often increased run time by orders of magnitude.

Exercise 2

Our methods for exercise two were fairly straightforward. The most effective classification method we found was to simply apply our method from exercise one to a given review as a whole. For this exercise, we found that it was more accurate to take the average of the word scores rather than the most prominent. This is likely because a review contains approximately four times more words than a given paragraph, making it more resistant to outliers. Seeing as our individual paragraph sentiment classifier gave a value of 4 for nearly everything, our weighted mean was close to 4 every time as well. That being said, this method produces an average rms of 0.84.

Exercise 3

Attempting authorship attribution in exercise 3 was an interesting challenge with such a small data set. Our initial attempts involved a feature set including average word, sentence, and paragraph length, unique vocabulary density, and word frequency. None of these features, or any combination of them, gave us any amount of accuracy for authorship prediction. Eventually we found that doing POS tagging on the review and counting the sum of adjectives and adverbs was quite a useful feature, giving us our first taste of some correct guesses. To this we added average sentence length, and unigram word presence, and saw an RMS of about .85. Adding comma density to this brought our average RMS to a respectable 0.83: far above the chance baseline for authorship attribution. Interestingly enough, when experimenting with the first feature set, our classifier took a liking to Eriq and Aldrin's reviews. Eric had four reviews, and it was hard to prevent this from skewing the classifier into always picking him. When we switched feature sets to our final choices, we found that Joseph White was predicted more often than anyone else. Some attempts to narrow the binning of adjective count and average sentence length helped mitigate this problem, but it still persisted to some degree.

Conclusion

It was hard to prevent the Naive Bayes classifier from predicting 4 on this strange data set, and in retrospect, it may have been better to attempt other classifiers. Interestingly, we found that all attempts made to use more complex features proved futile and only offered very minor increases in accuracy. Overall, our project achieved average results for exercise 1 and 2, and did fairly well on exercise 3.