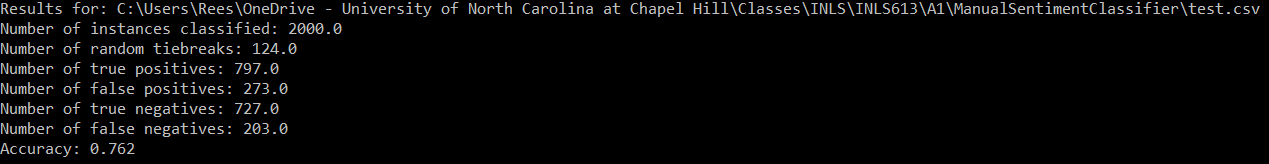
Rees Braam

INLS 613

1/29/2019

Assignment 1

*Peak classifier accuracy: 0.7595*



cd C:\Users\Rees\OneDrive - University of North Carolina at Chapel Hill\Classes\INLS\INLS613\A1\ManualSentimentClassifier

java -jar ManualSentimentClassifier.jar train.csv test.csv positive.txt negative.txt

**Q1**

I used three different approaches when trying to improve the classifier: through intuition, the internet, and programming. At first, I just tried to add words by intuition to see if there would be some easy, obvious words. Doing this had a minimal effect and felt like very slow going, so I decided I would take the suggestion to look online. Looking online, I initially found what I thought was a decent wordlist­.[1] However

**Q2**

With an even distribution of positive and negative reviews, it would be expected that merely guessing should result in a 50% accuracy rate. This is because for each review, no matter what you guess, you always have a 50% chance of being correct because there are only 2 possible outcomes. As such, any accuracy lower than 50% should be an immediate red flag that essentially says you’ve trained your model in the opposite direction, or at least that something somewhere is going wrong.

If I knew that test.csv were split 80-20 between positive and negative reviews, this would not affect my strategy for improving the classifier whatsoever. When creating a model, you should never look at the test data, and even if you did, you should not modify your classifier based on it. If you did, then your test data would really just become alternative training data, and your classifier would not perform as well on data it had never seen before.

**Q3**

[1] http://www2.imm.dtu.dk/pubdb/views/publication\_details.php?id=6010