Homework 3: Sentiment Analysis Report

Stuart Harley

Warmup: NB Counts

	Comedy (+)	Action (-)
Fun	3	1
Couple	2	0
Love	2	1
Fast	1	2
Furious	0	2
Shoot	0	4
Fly	1	1
Total	9	11

Fast, couple, shoot, fly

P(+)P(S|+) = (2/5) * (1+1)/(9+20) * (2+1)/(9+20) * (0+1)/(9+20) * (1+1)/(9+20)

= (2/5) * (2*3*1*2)/(29^4) = 6.79 * 10^-6

P(-)P(S|-) = (3/5) * (2+1)/(11+20) * (0+1)/(11+20) * (4+1)/(11+20) * (1+1)/(11+20)

= (3/5) * (3*1*5*2)/(31^4) = 1.95 * 10^-5

Classified as Action

Assignment

After training the original model using only the provided training set, the following were the predictions of the model on the test set.

The program does what it should do. : Positive
It functions adequately. : Neutral
The program sucks. : Positive
This thing runs like a pregnant cow. : Negative
It was a little slow, but not too bad. : Negative
Slow. Slow. SLOW! : Negative
Great software! : Neutral
Worth the trouble to install : Negative

After looking at these initial results, the classifier did not seem to be very accurate. Accuracy was 3/8 or 37.5%. Macro averaged precision was 33%, and macro averaged recall was 30.3%. Therefore, the macro averaged F-measure was 31.8% using a Beta of 1 since precision and recall are equally important for this scenario.

For the next section I chose the document “The program sucks”, which was incorrectly classified as Positive. The sentiment probabilities for this document are calculated as follows (the word “sucks” was not in the training set, so it is not included in the calculations):

P(Pos)P(doc|Pos) = (6/18) * (5+1)/(51+165) * (2+1)/(51+165) = 1.2860e-04

P(Neu)P(doc|Neu) = (6/18) * (3+1)/(55+165) * (1+1)/(51+165) = 5.5096e-05

P(Neg)P(doc|Neg) = (6/18) * (3+1)/(59+165) * (1+1)/(51+165) = 5.3146e-05

The difference between the probability sums of the correct Negative class and the predicted Positive class is 7.5454e-05.

The term that is mostly causing the system to misclassify the document is “the” because “the” occurs 5 times in positive reviews but only 3 times in negative reviews. However, the main reason that this document is being incorrectly classified is that the word “sucks” is not in the training set, so it is not included in the calculations. The word “sucks” is generally a negative word and if that word had been included in any of the negative training documents, this document would likely have been classified correctly.

The document I am choosing to add to the training set is a negative review stating “Sucks. Sucks. Sucks!”. I am choosing to add this document because the calculated Positive probability sum will now be multiplied by a small probability of the word “sucks” whereas the Negative probability sum will be multiplied by a much larger probability of the word “sucks” which should change the classification. Also, because I am only using this one new word in the document, it shouldn’t change any of the other calculated probability sums by much.

After adding this document to the test set, the following were the predictions of the model:

The program does what it should do. : Positive
It functions adequately. : Neutral
The program sucks. : Negative
This thing runs like a pregnant cow. : Negative
It was a little slow, but not too bad. : Negative
Slow. Slow. SLOW! : Negative
Great software! : Neutral
Worth the trouble to install : Negative

When comparing these results to the previous results we can see that the results are as I predicted. The “The program sucks.” document is now correctly predicted as Negative, but no other classifications have changed. The overall accuracy of the system has improved and is now 4/8 or 50%.

After adding the MPQA Subjectivity Cues Lexicon to the system the following were the predictions of the model on the test set:

The program does what it should do. : Neutral
It functions adequately. : Negative
The program sucks. : Negative
This thing runs like a pregnant cow. : Negative
It was a little slow, but not too bad. : Negative
Slow. Slow. SLOW! : Negative
Great software! : Positive
Worth the trouble to install : Negative

After adding this lexicon, the accuracy of the model is now 5/8 or 62.5%.

Looking at the results, the document “Great software!” is now correctly classified as positive when it was incorrectly classified as Neutral before. The word “great” did not previously exist in the training set but it is included in the lexicon with a Positive rating. Because of this, the document is now calculated to be positive.

After training a new model on the Amazon training set and testing it on the Amazon testing set, the results were as follows.

	Predicted Positive	Predicted Negative
Actual Positive	110062	89938
Actual Negative	63910	136090

Looking at these results, the overall accuracy of the model is 61.538%. The precision of the model is 55.031%. The recall of the model is 63.264%. And the F-measure is 58.861%.

In this homework it was cool to see how when we added new information to the model it improved. However, the amazon reviews did not seem to be as classified as well as I thought they would be after a very long training session. I thought that all of that training data would have allowed that model to perform very well. I’m guessing that because the reviews were so long, the importance of some words was lessened.

In the future, I would maybe only give us a subset of the Amazon dataset to use. I didn’t write my code to be the most efficient because the initial sets were small. So when I got to the massive Amazon dataset, it took me 9.27 hours to train the model and classify the test set.