CSCI 5832: Natural Language Processing
Assignment 3: Text classification

Paramjot Singh

Assignment Report

Introduction

I used pure python implementation for this assignment. Since, data set wasn't large enough for using other mechanisms, and Naive-Bayes was already coming pretty accurate. So starting with the implementation decisions. Similar to previous assignment, I split the data based on Pareto principle. 80% for Training set(150 reviews) and 20% for Dev set(40 reviews) and tried cross validation later on to check the accuracy.

Segmentation

While creating the dictionary for positive and negative reviews, I removed occurences of punctuations (wife, -> wife) and converted them to lower case and did split by spaces. I know this won't help much while calculating accuracy, but it will help in reducing size of vocabulary and also will help in removing high frequency words. After calculating the word frequency dict for positive and negative category. I clipped on the high frequency(>100) and single occurence words. This removed words like 'the', 'and', 'a', 'was', etc which don't help in sentiment analysis.

Naive-Bayes and Add-one Smoothing

For the main text analysis, I calculated the probability against words in both category by using Naive-Bayes with add one smoothing. To avoid underflow, I converted the probabilities to log and add them up. I skipped the Prior while calculation (since it's same for both i.e. 0.5 or log(0.5)). After the probability calculation, compared the prob for both categories and saved value for higher number. Tried the code and it worked with accuracy of approx 92%. Although removing high frequency words didn't help much (probability was almost similar), but still kept the piece of code.On trying with different test data inputs, code gave accuracy values from 90 - 95%.

Observations and Conclusion

So based on final counts the accuracy is coming around 92% (average). Hoping to see good number with the final test data !