Write Up for HW1

For part a), I first created several different features to add onto the existing algorithm. I added the additional txt. Files provided for lists of abbreviations, and if the word in position 3 is in the list of abbreviations, then the output would be “EOS” else it would be “NEOS”. The result is slightly better than baseline with an accuracy of 96% on training data, but the increase in accuracy was not significant. Then I tried experimenting with different machine learning models. I changed from Decision Tree to Logistic Regression, which increased the accuracy to around 96.8% on the validation set. Then I tried to make all the neighboring words in the list as features. For every word in the list, if the word is in the list of abbreviations, then we would append 1 to the feature vector. Else we would append 0. For all the numerical features, it is extracted into the feature vector as is. The feature vector also includes the length of words, whether it is a special character, whether it is the new paragraph symbol ‘<P>’, whether it is a digit, whether it is a letter, and whether it is some frequent special characters such as “-”, “;”, “.”. The accuracy increased to .9904.

For part b), I first looked at the write up to explain different patterns of text classification and followed the procedure. I made specific classifiers for NNHEAD (word that starts with From: , Article: , Subject: etc.) , ADDRESS(matches email address or phone number only if it is not already classified as NNHEAD), QUOTE(matches “:”, “>” or white space followed by one character followed by white space OR the word “wrote”, “write” “says” followed by “:” appears), TABLE(count the number of words from previous line and current line and see if it is the same with fluctuation of 1). Then I extracted general features that might also be useful such as the number of special characters(for GRAPHICS), the number of digits, letters, white spaces

and capitalized words. Then I changed my classifier to SVM with rbf kernel. It yielded an accuracy of .8998 on the validation set.