HW1 CS-584 Report Deepak Kanuri(G01070295)

User name: thor

Rank and accuracy: Rank is 36(Because of many users) and accuracy is 78%(tried a lot)

Approach:

1)Read the training data and storing it in truthful and deceptive lists respectively.

2)Removed the stop words.

3)Removed punctuation

4)Normalized the text by lemmatizing it.

5)Found the regular expression of the normalized data.

6)Vectorized the data using 5000 features and ngram range of (1,2)

7)The data was fit and transformed for both truthful and deceptive lists.

8)Read the test data.

9)Removed the stop words.

10)Removed punctuation

11)Normalized the text by lemmatizing it.

12)Found the regular expression of the normalized data.

13)Vectorized the data using 5000 features and ngram range of (1,2)

14)The cosine similarity of the test data and truthful data is found

15)The cosine similarity of the test data and deceptive data is found

16)For every test data, the cosine similarities with truthful and deceptive are taken and put in a dictionary with its respective labels.

17)The dictionary is sorted.

18)Based on the ‘k’ value the last ‘k’ items and values of the dictionary are taken and the values are added to find the sum. Based on the sum, the number of 1’s are counted and thus if the sum is high meaning 1’s are high so the text file is deceptive, else it is truthful.

The reason I took this approach is that I thought that when comparing an unknown text we have to compare it to values we know for sure that is truthful or deceptive. That was my logical thinking when I started this assignment. I had a hard time going through the training data to read truthful and deceptive files separately. I then stored the text data as a list of lists in truthful and deceptive lists separately. While reading the files only I preprocessed the text data and stored it in the list. In the preprocessing I removed stop words, punctuation, lemmatized the data and found its regular expressions. After storing the data in the separate lists, I vectorized the truthful and deceptive data based on the ngram\_range=(1,2) and vectors of 5000. The reason I did this is because I saw many articles that said grouping words in the groups of two was more convenient that going for groups of three or four. The features I took as 5000 because the feature size of both truthful and deceptive lists were below and above 6000 respectively. So I just took 5000 as it was the nearest.

Later on I started reading the test data, thinking I was doing it sequentially. After lots of effort and many days of hard-work I found out the test data was not being read sequentially so I had to find an alternative to read the file, as there was no time. While reading the files only I preprocessed the text data and stored it in the list. In the preprocessing I removed stop words, punctuation, lemmatized the data and found its regular expressions.

Now I found the cosine similarity between the test and truthful data and the test and deceptive data. So now for every test data I took the cosine similarities and stored it in a dictionary and assigned their respective labels. Next I sorted the dictionary based on their keys and took the top k=95 values and sorted it and added the values. If the values was a high number then it was deceptive otherwise it was truthful.

For k-fold validation I manually split the training truthful and deceptive data and found the similarities and checked for different ‘k’ values. I initially tried a small value, then the TA said use a higher value, more precise square root of size of training data. Yet it was a smaller value. Then I increased it to 95 and saw that it was producing a substantially better output than the previous ones and stuck to 95.

On the whole it was a tiresome and tensed assignment. The leader-board was a pain in the neck. The continuous and numerous submissions made it difficult to keep up. I personally feel there should not be any leader-board, as it created self-doubt in coding the assignment and made me make multiple changes , making me wonder whether I was on the right path. After many sleepless nights I got an output. Even though I may not be on the top of the board, I think my approach of dividing the data into truthful and deceptive was unique.