IST736 HW3

Yicun Deng

Prof. Bei Yu

True-False and Pos-Neg model for deception\_data.tsv

For this assignment, I had used the Jupyter Notebook to code in Python to solve the problem set.

Data processing: First, I had import numpy and pandas packages to read in the data file. Then, I had defined data’s review as X\_train, data’s sentiment as y\_train, data’s lie as z\_train. Then, I used train\_test\_split to split the dataset into X\_train, X\_test, y\_train, y\_test, z\_train, z\_test with a test size of 0.4.

Question A: I used sklearn’s Countvectorizer with a setup of (1,2)gram, stopword=’english’ to vectorize my X\_train data. I choose to use countvectorizer to vectorize my data because of the robustness of this vectorizer. We don’t need extra information other than the word count to perform well in this task. This is also in lane with Occam’s razor—don’t bring in new existence when unnecessary.

Question B: I used sklearn/naïve\_bayes to import MultinomialNB package. Then, I defined nb\_clf1 for the POS-NEG classifier and nb\_cf2 for the T-F classifier. Then, I used the sorted(zip()) method learnt from the class to generate top feature for each of the categories.

Result:

Neg: ‘waitress’, ‘asked’, ‘minutes’, ‘salad’, ‘service’, ‘experience’, ‘went’, ‘place’, ‘food’, ‘restaurant’

Pos: ‘amazing’, ‘need’, ‘fresh’, ‘friendly’, ‘nice’, ‘prices’, ‘definitely’, ‘high’, ‘source’, ‘stuff’

False: ‘menu’, ‘minutes’, ‘went’, ‘like’, ‘services’, ‘best’, ‘experience’, ‘place’, ‘restaurant’, ‘food’

True: ‘life’, ‘ny’, ‘bad’, ‘came’, ‘didn’, ’dish’, ‘favorite’, ‘going’, ‘later’, ‘long’

Question C: I used sklearn.metrics to get precision, recall and classification report for each classifier:

POS-NEG:

[0.83333333 0.84210526]

[0.83333333 0.84210526]

precision recall f1-score support

n 0.83 0.83 0.83 18

p 0.84 0.84 0.84 19

accuracy 0.84 37

macro avg 0.84 0.84 0.84 37

weighted avg 0.84 0.84 0.84 37

T-F:

[0.83333333 0.84210526]

[0.83333333 0.84210526]

precision recall f1-score support

t 0.44 0.41 0.42 17

f 0.52 0.55 0.54 20

accuracy 0.49 37

macro avg 0.48 0.48 0.48 37

weighted avg 0.48 0.49 0.48 37

Question D: I used the method learnt from the class to do error analysis. I got 6 errors for the Pos-Neg classifier and 19 errors for the True-False classifier. Combining my knowledge of the top features of the classifiers. I find that the POS-NEG features makes more sense than TRUE-FALSE features (ny is not a truth feature that we should look at.) To improve the performance of POS-NEG classifier, I believe that we need a larger data-set to both improve the performance without risking of over-fitting. To improve the performance of True-False classifier, we should remove some of the top feature’s word like ‘ny’ and then train the classifier again.

Question E:

True-False classifying is harder. First of all, it is easier to tell a sentence is positive or negative than to tell a sentence is true or false in real life. Truth value of sentence requires external knowledge of the world. Secondly, to better classify true-false, maybe we should use the method we use from hw2, to see how far away each review is from the average of other reviews. This