# Assignment2 Report

Junhan Liu(20848916)

## Running instruction

1, `cd #the assignment folder`, i.e. `cd /user/junhanliu/614_a2`

2, In your command line type `python3 main.py` or `python3 main.py /xxx(directory)/.../#assignment folder/data` to initialize script

3, The driver script(main.py) takes only one parameter or no parameters. If passing in a parameter, feed the data folder path to the script. If none is given, the script is going to use a default path of the current running directory using `os.getcwd()+'/data'`

4, The data splits inside my `/data` folder are different from the ones generated for assignment 1. Additional labels are appended to each sentences(tokens) in each .csv files; labeled splits are re-generated using assignment 1 script. If one uses the old `/data` folder from assignment 1, it is not going to work since there are no labels.

5, Every file needed are in `/data` folder.

## Evaluation of Performance

The following table shows the performance of each scenario. Each scenario is repeated 10 times and calculate the mean accuracy percentage. For example, using unigram with stopword untouched, it achieves an average of 81.0053558914806% accuracy after repeating the prediction for 10 times.

Table 1: Accuracy of Model(with stopwords)

| unigram with stopwords | bigram with stopwords | uni/bi-gram with stopwords |
|---|---|---|
| 80.84893382720993% | 82.11282410651717% | 83.12894183601962% |

Table 2: Accuracy of Model(without stopwords)

| unigram without stopwords | bigram without stopwords | uni/bi-gram without stopwords |
|---|---|---|
| 80.59865852437682% | 77.82560816898588% | 82.32305536089699% |

## Discussion

The evaluation output of all three text features indicates the accuracy is higher when stopwords are not removed. The evaluation seems contradict our common understanding. It makes perfect sense to one that "the accuracy should go up without stopwords". However, the procedure should be done with more care.

It depends on different text analysis tasks. For sentiment analysis, for example, stopwords like “did not”, “was not”, “should not” may express negative sentiment in some sentences. The review **“I didn’t like the product.”** express a clear negative sentiment. But after removing stopwords in it, it becomes to **“like product”**. The later filtered sentence express a neutral or positive sentiment instead. This could be a contributive reason of the higher accuracy of the sets with stopwords in it.

By checking the table horizontally, we evaluate the performance under different text features. Unigram + bigram performs better than unigram or bigram solely. It might be the reason that unigram and bigram catches more features than solely using bigram or unigram. For example, the sentence “i dont like it and i am not happy”. A word’s context can matter nearly as much as its presence. By adding bigram, “dont like” and “not happy” passes negative dimension of the sentence. By using a combination of unigram and bigram, it seems more feature can be used for decisions.