Nick Benevento

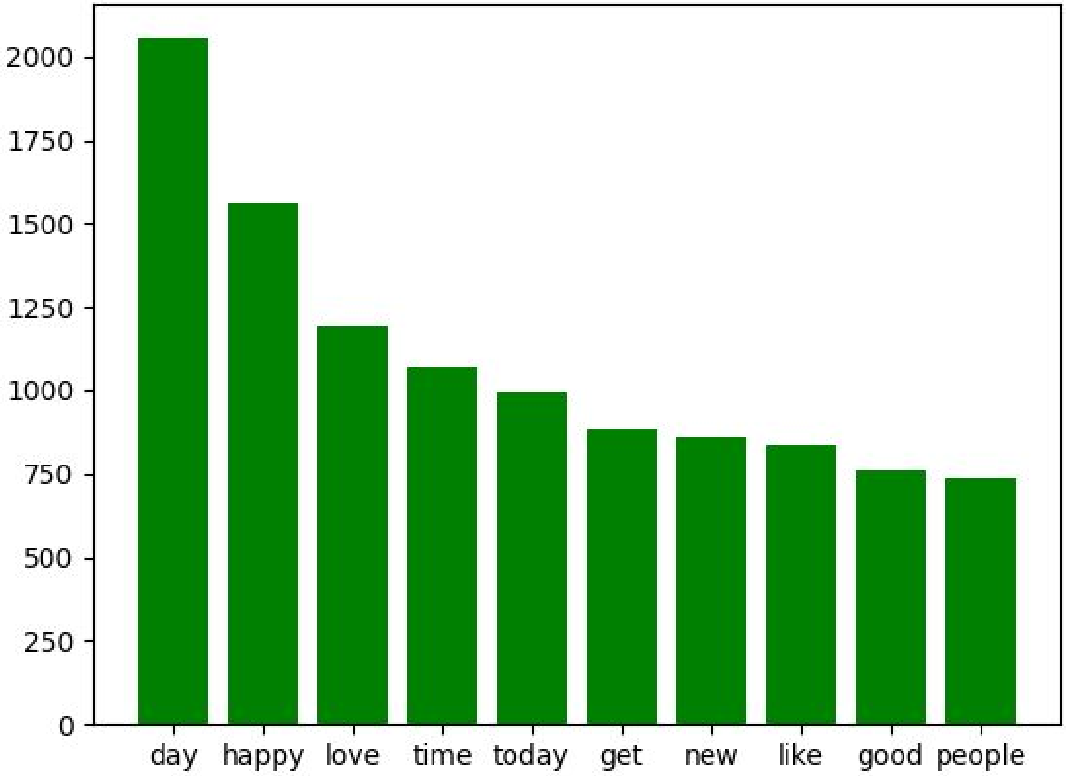
NLP Assignment 3

Part 1

1. Most similar to 'hate': [('words', 0.9434675574302673)]

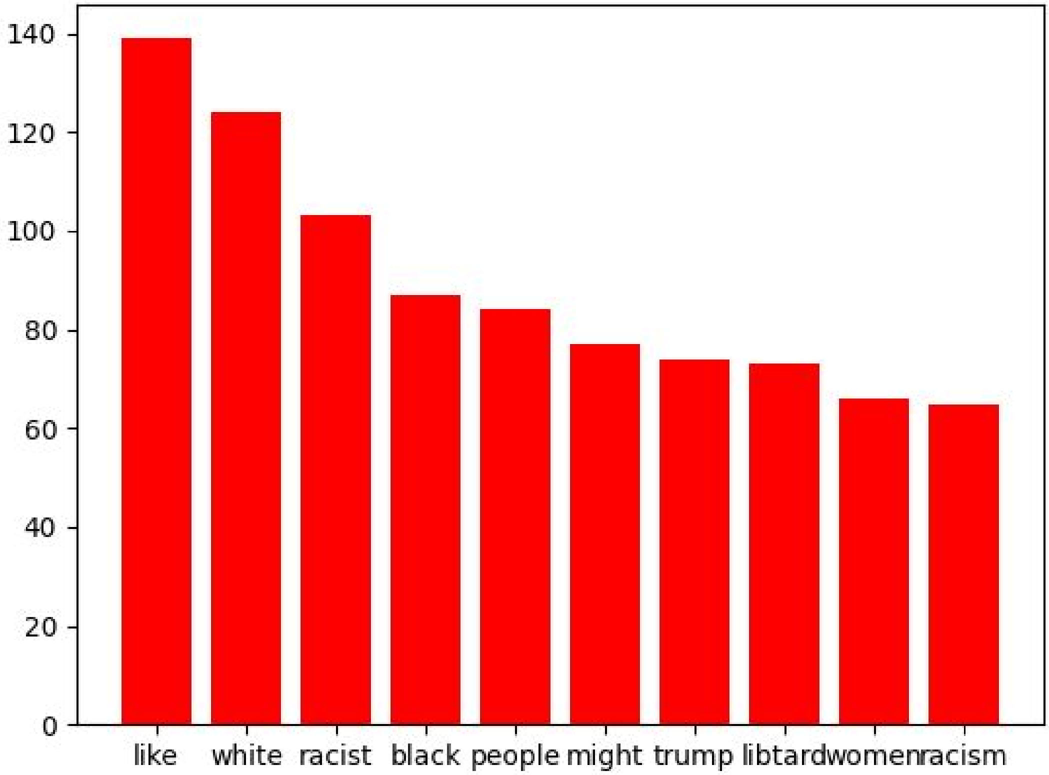
   Most similar to 'like': [('looks', 0.864230751991272)]
2. Plots:

Figure 1: Non-hate words



[('day', 2056), ('happy', 1561), ('love', 1195), ('time', 1072), ('today', 997), ('get', 885), ('new', 857), ('like', 839), ('good', 760), ('people', 737)]

Figure 2: Hate words



[('like', 139), ('white', 124), ('racist', 103), ('black', 87), ('people', 84), ('might', 77), ('trump', 74), ('libtard', 73), ('women', 66), ('racism', 65)]

3. (Using imdb dataset): Analogy for love + life - hate: [('marriage', 0.5004991888999939)]

Part 2:

[Disclaimer: When trying to make label predictions with the models, I kept getting an error saying that my test data array had a mismatch in its core dimension when using separate train and test files. I could not figure out the solution to this, but combining the train and test data into one file and using the "train_test_split" method worked with no issues, so that is what I used for these results. It should give the same results, as I split the data so that there are 10,000 test tweets, which reflect the original "test.csv" file]

1. Naive-Bayes classifier:

Accuracy: 0.9439056094390561

f1-score: 0.624246483590087

Additional Output:

```
          precision    recall  f1-score   support


       0       0.98      0.96      0.97      9327
       1       0.57      0.69      0.62       674


    accuracy                           0.94     10001
   macro avg       0.77      0.83      0.80     10001
weighted avg       0.95      0.94      0.95     10001
```

2. Logistic Regression, binary bag-of-words

Accuracy: 0.9543045695430457

f1-score: 0.5772432932469935

Additional Output:

```
          precision    recall  f1-score   support


       0       0.96      1.00      0.98      9278
       1       0.87      0.43      0.58       723


    accuracy                           0.95     10001
   macro avg       0.91      0.71      0.78     10001
weighted avg       0.95      0.95      0.95     10001
```

3. Logistic Regression, binary bag-of-ngrams (unigrams and bigrams)

I have implemented this model, but when I ran it the process kept getting killed. I looked in my system resources and it was using all 30gb of my ram, as well as all 37 gb of my swap memory. I could not find a reason as to why it was using so much memory, and trying the limited solutions that I found online did not work either, so I am unable to report accuracies for this model.