LING 406 Term Project Questions

Question 1

In building the baseline system, what shallow text features make sense for the task as a first-stab approach? (a language modeling approach is a standard way to tackle this problem using, for example, a bag-of-words / a unigram model approach);

For the baseline system, I preprocessed the text with the a unigram model approach. Specifically, I counted the frequencies of a word if it were to be negative or positive, convert it into probabilities, and then multiply all probabilities of a word in a sentence to see which particular sentiment is much more likely to happen for this sentence. It makes sense as a particularly simple first-stab approach since it is not only really easy to implement and it yields a comparably high accuracy, but this approach also takes all of the words in a sentence into account without leaving any word in a sentence behind. (please refer to baseline.ipynb)

Question 2

What machine learning models are suitable for conducting sentiment analysis? (i.e., compare 3-4 learning algorithms of your choice). Which performs best given your features?

I trained and tested the nltk movie reviews data set with three different machine learning models (GaussianNB, Decision Tree and K-Nearest Neighbors). Before training, I preprocessed the dataset in the following manner:

- Encoded each *type* of word (not word tokens) with a float value ranging from 0.00 to 327.00, assigned sentences with positive sentiment to 1, and assigned sentences with negative sentiment to 0.
- Took the first 95 words of each sentence for training and testing. (Empirically proven to be most effective in my case, elaborated in later sections.)
- Shuffled the dataset in a random fashion.
- Split the dataset to a training and a testing set using 20-fold cross validation. (That is 95% of the dataset belongs to the training set and 5% of the data belongs to the testing set)

The end result seems to be pretty shocking:

```
Now classifying Naive_Bayes
        Average Macro F1 for Naive_Bayes:               0.651112244403391
        Average Macro Precision for Naive_Bayes:        0.6622860857311929
        Average Macro Recall for Naive_Bayes:           0.6571552683098333
        Average Macro Accuracy for Naive_Bayes:         0.6577575757575758
Now classifying Decision_Tree
        Average Macro F1 for Decision_Tree:             0.9506558435490311
        Average Macro Precision for Decision_Tree:      0.9522443639291465
        Average Macro Recall for Decision_Tree:         0.952784090909091
        Average Macro Accuracy for Decision_Tree:               0.9517171717171719
Now classifying Nearest Neighbors
        Average Macro F1 for Nearest Neighbors:         0.5734981794565313
        Average Macro Precision for Nearest Neighbors:  0.7273066210969042
        Average Macro Recall for Nearest Neighbors:             0.6241351862095972
        Average Macro Accuracy for Nearest Neighbors:           0.6202373737373736
```

Decision Tree has an accuracy of 95%!!!! That's a 30% increase from the other two models, which only yields around 60% of accuracy. (refer to machine_learning.ipynb)

Question 3

How would you improve the baseline model? (i.e., what text features are most beneficial to the task of sentiment analysis?) Experiment with at least 4 different features that go beyond a language modeling representation.

Here is the list of all features I tried for my baseline model:

- Takes all tokens from a sentence as unigram. (feature: all words in a sentence)

```
Accuracy 81.75 %
Precision 84.65608465608466 %
Recall 78.43137254901961 %
F1-Score 81.42493638676845 %
```

- Takes all non-trivial tokens from a sentence as unigram. (stripped off punctuation marks)

```
Accuracy 82.0 %
Precision 84.70588235294117 %
Recall 75.78947368421053 %
F1-Score 80.0 %
```

- Takes all trivial-tokens from a sentence as unigram.

```
Accuracy 58.25 %
Precision 57.89473684210527 %
Recall 55.83756345177665 %
F1-Score 56.84754521963824 %
```

- Takes only nouns, verbs, advs and adjs from a sentence as unigram.

```
Accuracy 81.75 %
Precision 84.04907975460122 %
Recall 74.45652173913044 %
F1-Score 78.96253602305475 %
```

At this point, I fail to see which feature will significantly boost my accuracy, these features yield similar accuracy, precision, recall and F1-score (except the obvious outlier) and I am not sure which approach is going to be the most optimal.
(refer to baseline.ipynb)

Question 4

How does the size of the various feature sets you are experimenting with influence the performance of sentiment analysis? Compare the performance of different feature sets under the same feature selection scenario and machine learning algorithm.

The performance of sentiment analysis is not directly related to the absolute size of the feature set. For this task, I switched back to the Decision Tree Model and I tried playing with various feature sets. My first feature set contains every word in the document, then I took the first 95 words of each review for training and testing. Here is the end result.

```
Now classifying Naive_Bayes
        Average Macro F1 for Naive_Bayes:               0.651112244403391
        Average Macro Precision for Naive_Bayes:        0.6622860857311929
        Average Macro Recall for Naive_Bayes:           0.6571552683098333
        Average Macro Accuracy for Naive_Bayes:         0.6577575757575758
Now classifying Decision_Tree
        Average Macro F1 for Decision_Tree:             0.9506558435490311
        Average Macro Precision for Decision_Tree:      0.9522443639291465
        Average Macro Recall for Decision_Tree:         0.952784090909091
        Average Macro Accuracy for Decision_Tree:               0.9517171717171719
Now classifying Nearest Neighbors
        Average Macro F1 for Nearest Neighbors:         0.5734981794565313
        Average Macro Precision for Nearest Neighbors:  0.7273066210969042
        Average Macro Recall for Nearest Neighbors:             0.6241351862095972
        Average Macro Accuracy for Nearest Neighbors:           0.6202373737373736
```

The result seems to be pretty good, but I wonder what will happen if I try to take each word and its part of speech into account? Here is the end result.

```
Now classifying Naive_Bayes
        Average Macro F1 for Naive_Bayes:               0.5580782156013722
        Average Macro Precision for Naive_Bayes:        0.5690759556014228
        Average Macro Recall for Naive_Bayes:           0.5672194513382942
        Average Macro Accuracy for Naive_Bayes:         0.5642626262626262
Now classifying Decision_Tree
        Average Macro F1 for Decision_Tree:             0.9439293114654232
        Average Macro Precision for Decision_Tree:      0.9440668498168497
        Average Macro Recall for Decision_Tree:         0.9439709415658009
        Average Macro Accuracy for Decision_Tree:               0.9442373737373737
Now classifying Nearest Neighbors
        Average Macro F1 for Nearest Neighbors:         0.67299640885349
        Average Macro Precision for Nearest Neighbors:  0.6865377426456968
        Average Macro Recall for Nearest Neighbors:             0.680391850466464
        Average Macro Accuracy for Nearest Neighbors:           0.6832474747474747
```

It seems a little bit lower for the Decision Tree model, however, the accuracy for Naive Bayes and Nearest Neighbor actually plummeted. This leads me thinking, do we need to take care of

each word at all? Can we replace each word with the part of speech of each word in the feature set? Here is the end result.

```
Now classifying Naive_Bayes
        Average Macro F1 for Naive_Bayes:               0.5945666407104399
        Average Macro Precision for Naive_Bayes:        0.5958112281776623
        Average Macro Recall for Naive_Bayes:           0.5960295423283979
        Average Macro Accuracy for Naive_Bayes:         0.5972828282828283
Now classifying Decision_Tree
        Average Macro F1 for Decision_Tree:             0.9527096415489273
        Average Macro Precision for Decision_Tree:      0.9530015155066428
        Average Macro Recall for Decision_Tree:         0.952961333878887
        Average Macro Accuracy for Decision_Tree:               0.9527474747474749
Now classifying Nearest Neighbors
        Average Macro F1 for Nearest Neighbors:         0.6600389206706108
        Average Macro Precision for Nearest Neighbors:  0.6702061004098779
        Average Macro Recall for Nearest Neighbors:             0.6643329612096799
        Average Macro Accuracy for Nearest Neighbors:           0.6707424242424241
```

It seems that this is the best arrangement we have for this task. i.e. Take the part of speech of each word for training and testing. However, the accuracy for this arrangement isn't higher than other feature sets by a very wide margin.