**CS 6320.002: Natural Language Processing**

**Homework 2**

**Xiaojie Zhu**

**XXZ180012**

**Writeup Question 1.1:** There are two other ways we could have handled the square brackets. We could have deleted those words completely, or we could have removed the square brackets but left the words untagged. Why did we do it the way we did (with tags)? What are some pros and cons of the three different ways of handling the square brackets? Give at least one pro or one con for each way.

Answer: by tagging the words, we are making sure that we will not lose information in the piece of text, and we can explicitly mark the origin of the words.

Pros:

Tag all words: we will keep all the information and also distinguish them from the words from the commentator.

Deleted words completely: easy to achieve, and can trim the text so that only the original words will be preserved.

Removed the square brackets: retain all information, and also easy to achieve.

Cons:

Tag all words: it takes efforts to achieve, and analysis of the extra words would be expensive.

Deleted words completely: we will lose some information by doing this, sometimes it might contain important information.

Removed the square brackets: we will not be able to distinguish it from the original text, and it may cause bias.

**Writeup Question 1.2:** What data type did you use to save the negation-ending tokens? Explain your choice.

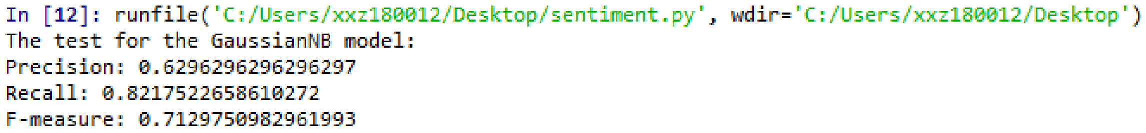
Answer: I used a set to store the negation-ending tokens. Because a set can provide me with $O(1)$ complexity of verifying whether a token belongs to it.

**Writeup Question 3.1:** Looking at the formulae for precision, recall, and f-measure, what does each of them measure? Why do we need all three of them?

Answer: ***Precision*** means among all the labels that are predicted true, the ratio of the labels that are actually true. ***Recall*** means among all the labels that are actually true, the ratio of the labels that are also predicted true. ***F-measure*** is the measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test. We need all three of them so we can make a generic evaluation of the model, plus, the calculation of ***f-measure*** also requires the calculation of ***precision*** and ***recall***.

**Writeup Question 3.2:** What is the performance of the GaussianNB model?

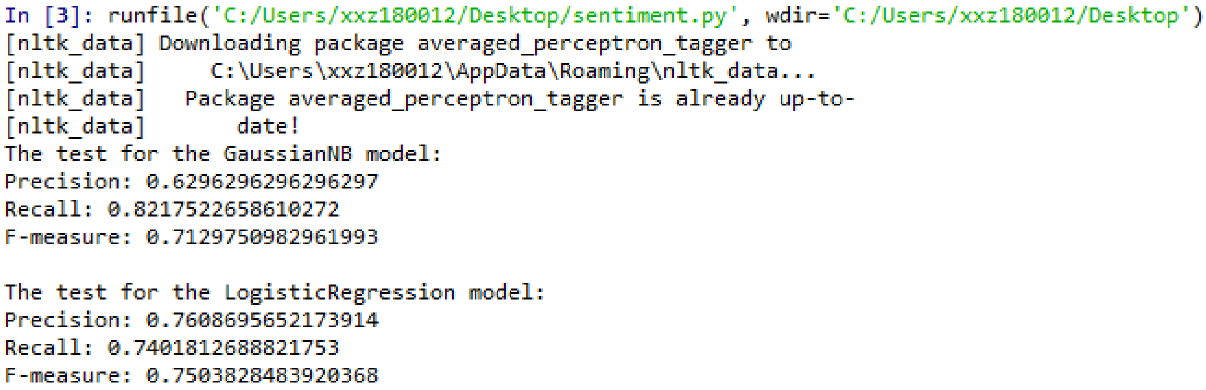
Answer: The results are below, as we can see, the model predicted the test data with 0.63 precision, 0.82 recall, and 0.71 F-measure, in general, this could be considered as good prediction.



```
In [12]: runfile('C:/Users/xxz180012/Desktop/sentiment.py', wdir='C:/Users/xxz180012/Desktop')
The test for the GaussianNB model:
Precision: 0.6296296296296297
Recall: 0.8217522658610272
F-measure: 0.7129750982961993
```

**Writeup Question 3.3:** What is the performance of the Logistic-Regression model? Discuss which model performed better on this data and why you think that might be.

Answer:



```
In [3]: runfile('C:/Users/xxz180012/Desktop/sentiment.py', wdir='C:/Users/xxz180012/Desktop')
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\xxz180012\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
The test for the GaussianNB model:
Precision: 0.6296296296296297
Recall: 0.8217522658610272
F-measure: 0.7129750982961993

The test for the LogisticRegression model:
Precision: 0.7608695652173914
Recall: 0.7401812688821753
F-measure: 0.7503828483920368
```

From the results, we can see that the Logistic-Regression model performed better precision than Gaussian-NB model, although the recall of the Logistic-Regression model is lower than Gaussian-NB model, Logistic-Regression model outperformed Gaussian-NB model on F-measure as well. The reason for the difference is that NB Model assumes all the features are conditionally independent, whereas in Logistic-

Regression model, the splits feature space linearly, thus, if some of the features are dependent on each other, the prediction might be poor. In our case, the data is large, and so there might be correlation, and Logistic-Regression model will usually outperform Gaussian-NB model in such situations.

**Writeup Question 3.4:** Report the top 10 features of your LogisticRegression model. Why do we sort by the absolute value of weight, rather than the actual value? Explain why it is important to do so and what it means in terms of the features.

Answer: The top 10 features of the Logistic-Regression model are below:

```
In [39]: print('\n')
    ...: print("The top 10 features of the LogisticRegression model are: ")
    ...: print(top_features(lrModel, 10))


The top 10 features of the LogisticRegression model are:
[('too', -3.34703645412225), ('bad', -2.400217209928155), ('dull',
-2.1317811306129797), ('still', 1.8622723763639955), ('fails',
-1.7855009876939378), ('boring', -1.7718456996713983), ('fun',
1.5995453224664318), ('funny', 1.578866444469395), ('best',
1.5715904672775076), ('enjoyable', 1.4827362629049177)]
```

The reason we use the absolute value instead of the actual value is because the coefficients in a Logistic-Regression model are log odds ratios. The positive or negative sign of each number simply indicates that the effect of the feature is incremental or decremental. The absolute value of the number indicates the scale of the effect. Thus, we need to use the absolute value instead of the actual value.

**Writeup Question 4.1:** What is the performance of the new model with extra features? What are its top 10 features? How does it compare with the old model without extra features? Why do you think that might be?

Answer:

```
The test for the LogisticRegression model:
Precision: 0.6887755102040817
Recall: 0.8157099697885196
F-measure: 0.7468879668049793


The top 10 features of the LogisticRegression model are:
[('too', -3.369537213519854), ('bad', -2.3274458765378423), ('dull',
-2.0341874158558957), ('still', 1.8663422690066203), ('boring',
-1.757727163867287l), ('fails', -1.7457746852384977), ('best',
1.5129676218126484), ('funny', 1.4940968534524472), ('enjoyable',
1.4665829321841217), ('fun', 1.457307673824773)]
```

We can see from the results that the precision and F-measure of the new model decreased but the recall increased. Overall, the changes are small. The top 10 features are same as the previous model, except that the orders are slightly different.

Here, to compute the top 10 features, if we do not skip the index that is not in the vocabulary, then we will get an error, this is because the actual top feature is in one of the last 3 indices of the array, which the vocabulary does not have the words to refer to.

The reason is that the newly added feature can increase the information for the model to predict the snippets, thus increasing the overall recall rate, but the precision might not be affected by the added information.

**Writeup Question 5.1:** How long did this homework take you to complete (not counting extra credit)?

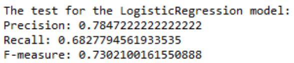
Answer: it took me around 5 hours to write the code and writeup questions, which the time has been distributed in 3 days.

**Writeup Question 5.2:** Did you discuss this homework with anyone?

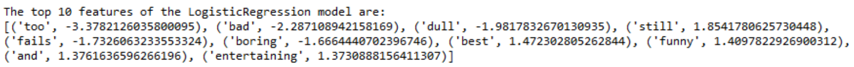
Answer: I discussed the homework with Joel Yin.

**Writeup Question 6.1:** Train, text, and examine the feature weights of a new LogisticRegression model using the new version of score snippet(). What is the performance of the new model with extra features? What are its top 10 features? How does it compare with the model that used the old version of score snippet()?

Answer:



```
The test for the LogisticRegression model:
Precision: 0.7847222222222222
Recall: 0.6827794561933535
F-measure: 0.7302100161550888
```



```
The top 10 features of the LogisticRegression model are:
[('too', -3.3782126035800095), ('bad', -2.287108942158169), ('dull', -1.9817832670130935), ('still', 1.8541780625730448),
('fails', -1.7326063233553324), ('boring', -1.6664440702396746), ('best', 1.472302805262844), ('funny', 1.4097822926900312),
('and', 1.3761636596266196), ('entertaining', 1.3730888156411307)]
```

From the results, we can see that, after adding the functions from the WordNet and use the weights of unfound words' synonyms or negative weights of their antonyms, we increased the precision of the model, nevertheless, the recall has been decreased and the F-measure remained roughly the same as the previous model.

The top 10 features are also changed, the last two features in this model ‘*and’* and ‘*entertaining’*, whereas in previous model, the last two were ‘*enjoyable’* and ‘*fun’*.