Project 4

Ali Harb

Your assignment in Project 4 is to answer either 6.10 exercise 3 or 6.10 exercise 4 from Natural Language Processing with Python.¶

Library

Headers used to perform the sentiments and the network analysis

In [57]:
```
import nltk.classify.util
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import movie_reviews
```

Sorted map function using iteritive method

In [58]:
```
def extract_features(word_list):
    return dict([(word, True) for word in word_list])
```

Laod Data

Get the positive and negative data from the movies reviews dataset

In [59]:
```
positive_fileids = movie_reviews.fileids('pos')
negative_fileids = movie_reviews.fileids('neg')
```

Let’s separate the newelty creeated fields into positive and negative reviews to pass them to the training and testing dataset.

In [60]:
```
features_positive = [(extract_features(movie_reviews.words(fileids=[f])),'Positiv
features_negative = [(extract_features(movie_reviews.words(fileids=[f])), 'Negati
```

Devide the dataset into trainging and testing datasets with 80% train and 20% test.

In [61]:
```
# Split the data into train and test (80/20)
threshold_factor = 0.8
threshold_positive = int(threshold_factor * len(features_positive))
threshold_negative = int(threshold_factor * len(features_negative))
```

Extract the training and testing features

In [62]:
```
features_train = features_positive[:threshold_positive] + features_negative[:thre
features_test = features_positive[threshold_positive:] + features_negative[thresh
print("\nNumber of training datapoints:", len(features_train))
print("Number of test datapoints:", len(features_test))
```

```
Number of training datapoints: 1600
Number of test datapoints: 400
```

Create a Naive Bayes classifier of the giveing traininig feature object.

In [63]:
```
classifier = NaiveBayesClassifier.train(features_train)
print("\nAccuracy of the classifier:", nltk.classify.util.accuracy(classifier, fe
```

```
Accuracy of the classifier: 0.735
```

Display the top most informative words

```
In [64]: print("\nTop 30 most informative words:")
         classifier.show_most_informative_features(30)
```

```
Top 30 most informative words:
Most Informative Features
             outstanding = True           Positi : Negati =     13.9 : 1.0
               insulting = True           Negati : Positi =     13.7 : 1.0
              vulnerable = True           Positi : Negati =     13.0 : 1.0
               ludicrous = True           Negati : Positi =     12.6 : 1.0
             uninvolving = True           Negati : Positi =     12.3 : 1.0
                  avoids = True           Positi : Negati =     11.7 : 1.0
              astounding = True           Positi : Negati =     11.7 : 1.0
             fascination = True           Positi : Negati =     11.0 : 1.0
               animators = True           Positi : Negati =     10.3 : 1.0
                  seagal = True           Negati : Positi =     10.3 : 1.0
                  darker = True           Positi : Negati =     10.3 : 1.0
                  symbol = True           Positi : Negati =     10.3 : 1.0
               affecting = True           Positi : Negati =     10.3 : 1.0
                    anna = True           Positi : Negati =     10.3 : 1.0
                 idiotic = True           Negati : Positi =      9.8 : 1.0
               represent = True           Positi : Negati =      9.0 : 1.0
                  annual = True           Positi : Negati =      9.0 : 1.0
                bothered = True           Negati : Positi =      9.0 : 1.0
               illogical = True           Negati : Positi =      9.0 : 1.0
                  hatred = True           Positi : Negati =      9.0 : 1.0
                   mulan = True           Positi : Negati =      9.0 : 1.0
                palpable = True           Positi : Negati =      9.0 : 1.0
                 offbeat = True           Positi : Negati =      9.0 : 1.0
               strengths = True           Positi : Negati =      9.0 : 1.0
                fairness = True           Negati : Positi =      8.3 : 1.0
                   naval = True           Positi : Negati =      8.3 : 1.0
                   moody = True           Positi : Negati =      8.3 : 1.0
                  hudson = True           Negati : Positi =      8.3 : 1.0
                seamless = True           Positi : Negati =      8.3 : 1.0
                 studies = True           Positi : Negati =      8.3 : 1.0
```

Test the classifier with few input sentances

```
In [65]: input_reviews = [
             "It is an amazing movie",
             "This is a dull movie. I would never recommend it to anyone.",
             "it is an outstanding movie"
             "The cinematography is pretty great in this movie",
             "The direction was terrible and the story was all over the place",
             "The location of cinematography is close to my house",
             "unbelievable movie",
             "Best movie i've ever seen",
           ]
```

In [66]:
```
print("\nPredictions:")
for review in input_reviews:
    print("\nReview:", review)
    probdist = classifier.prob_classify(extract_features(review.split()))
    pred_sentiment = probdist.max()
    print("Predicted sentiment:", pred_sentiment)
    print("Probability:", round(probdist.prob(pred_sentiment), 2))
```

```

Predictions:

Review: It is an amazing movie
Predicted sentiment: Positive
Probability: 0.61

Review: This is a dull movie. I would never recommend it to anyone.
Predicted sentiment: Negative
Probability: 0.77

Review: it is an outstanding movieThe cinematography is pretty great in this mo
vie
Predicted sentiment: Positive
Probability: 0.97

Review: The direction was terrible and the story was all over the place
Predicted sentiment: Negative
Probability: 0.63

Review: The location of cinematography is close to my house
Predicted sentiment: Positive
Probability: 0.54

Review: unbelievable movie
Predicted sentiment: Negative
Probability: 0.7

Review: Best movie i've ever seen
Predicted sentiment: Positive
Probability: 0.59
```

Can you explain why these particular features are informative? Do you find any of them surprising?

It is informative because it tells us what words are being used to indicate strong reactions. If you look at the top informative words, you can see that we have words such as “outstanding” to indicate positive reviews and words such as “insulting” to indicate negative reviews.

Some of the words prediction are incorrect such "moody" being positive while it should be a negative. The word "represent", "uninvolving”, “symbol", for example wouldn’t tell about the negativity or positivity as informative word.

In [ ]: