CIS668- Natural Language Processing

Processing and Classification of Sentiment or other Data

(Kaggle competition movie review phrase data)

Siri Chandana Sambatur
SUID- 829955672

Overview

This dataset was produced for the Kaggle competition, described here: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews , and which uses data from the sentiment analysis by Socher et al, detailed at this web site: http://nlp.stanford.edu/sentiment/. The data was taken from the original Pang and Lee movie review corpus based on reviews from the Rotten Tomatoes web site. Socher's group used crowd-sourcing to manually annotate all the subphrases of sentences with a sentiment label ranging over: 0- "negative", 1- "somewhat negative", 2- "neutral", 3- "somewhat positive", 4- "positive".

The two files mainly used for classification and building of our model are "train.tsv" and "test.tsv". For most of our training process the cross validation technique has be used. This will help us in generalizing the accuracy of our model. Also, since our model is being built on a sample of data, cross validation helps us build a better classifier.

Sampling of the data is necessary because the training data consists of 156,060 phrases. The technique that has been used to sample the data is random sampling. This random sampling technique samples data based on the limit number. The main reason for selecting random sampling technique is that each phrase of the training data has been given a fair chance to be selected.

Data Preprocessing

The phrases present in the data have to be transformed as this will help us build useful models. This is technique has been applied to cleanse our data and bring it to the desired format. Before applying the preprocessing steps the accuracy of our model is around 49% for a sample of size 2000. The following are the preprocessing steps that have been applied on the training data-

1. Tokenizing- the phrases in training data have been tokenized using the nltk package of python.
2. Lowercase conversion- all the tokens of a phrase have been converted to lowercase as the case of our token doesn't decide the sentiment of the phrase.
3. Removal of punctuations/non-alphabetic words- for the training data that is being used the emphasizes of punctuation that normal use like excessive question marks or full stops are absent. So, it is assumed that punctuations present in the phrase also don't effect the sentiment of the phrase and hence, they are removed.
4. Removal of stop words- the english stop words present in the nltk package have been used for this purpose. The reason behind this step is that even stop words like 'the', 'is' etc. don't help in deciding the sentiment of the phrase.

Feature Conversion

This is an essential step for our model creation because our model will be trained on the features that it has been given. Based on the sample that has been chosen the features for our model have been defined. Also, each phrase of the sample will be designed based on these features. The technique that has been applied is called the "bag of words"

Bag Of Words- The words that have been used in the sample selected are found and these set of words became the features based on which each phrase of the sample is analyzed based on this feature. For training our model two different types of features have been used-

- All the words that have been used in phrases present in our sample
- The 500 most common phrases that have been used in phrases present in our sample

Feature Set Functions

A wide variety of feature set functions have been used on the phrases represent in our sample and given for our model to train on. The model that has been used to train our sample is the Naive Bayes Classifier present in NLTK. The following is the different types of functions feature set that have been used for a sample size of 2000-

1. Word Vector/ Unigram Feature Set- The phrases in the sample are converted to a vector which defined by the presence of absence of a word. Each phrase has vector of trues and false and the length of the vector is equal to the length of the bag of words. The features are given to the Naive Bayes classifier of NLTK, using cross validation (n=5) the following are the results (accuracy and confusion matrix) obtained for the 2 different features-

   - With all words in the phrases taken as the word features

```
(0, 0.54)
The confusion matrix
  |  0   1   2   3   4 |
--+--------------------+
0 |  <.>  4  12   2   . |
1 |   . <11> 63   8   . |
2 |   .   5<189>  7   1 |
3 |   .   6  51 <16>  . |
4 |   .   .  16   9  <.>|
--+--------------------+

(1, 0.53)
The confusion matrix
  |  0   1   2   3   4 |
--+--------------------+
0 |  <1>  2  15   5   . |
1 |   .  <5> 56   5   1 |
2 |   .   6<192>  6   1 |
3 |   1   4  59 <13>  . |
4 |   .   .  14  13  <1>|
--+---------------------+

(2, 0.5275)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  5   8   1   . |
1 |   .  <9> 49   4   1 |
2 |   .   6<189> 11   . |
3 |   .   4  72 <13>  1 |
4 |   .   1  19   7  <.>|
--+---------------------+

(3, 0.505)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  3  13   1   . |
1 |   . <10> 62   5   2 |
2 |   .  10<177>  5   1 |
3 |   .   7  69 <15>  . |
4 |   .   1  17   2  <.>|
--+---------------------+

(4, 0.5525)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  3  13   2   . |
1 |   2 <11> 42   9   1 |
2 |   .   3<198> 10   1 |
3 |   .   2  72 <11>  1 |
4 |   .   1  13   4  <1>|
--+---------------------+

('mean accuracy', 0.531)
```

- With the 500 most common words in the phrases taken as word features

```
(0, 0.5075)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  15   3   . |
1 |   2  <9> 67   3   1 |
2 |   .   8<186>  8   . |
3 |   1   2  62  <8>  . |
4 |   .   1  20   4  <.>|
--+---------------------+

(1, 0.5225)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  3  19   .   . |
1 |   .  <7> 52   7   1 |
2 |   .   5<189> 10   1 |
3 |   .   4  59 <12>  2 |
4 |   .   .  22   6  <.>|
--+---------------------+

(2, 0.515)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  2  11   1   . |
1 |   1  <3> 55   4   . |
2 |   4   2<194>  5   1 |
3 |   1   1  79  <9>  . |
4 |   .   1  25   1  <.>|
--+---------------------+

(3, 0.48)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  1  14   2   . |
1 |   .  <5> 65   7   2 |
2 |   .   5<181>  6   1 |
3 |   .   1  81  <6>  3 |
4 |   .   .  16   4  <.>|
--+---------------------+
(4, 0.5325)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  17   1   . |
1 |   3  <5> 47  10   . |
2 |   .   4<199>  9   . |
3 |   1   2  72  <9>  2 |
4 |   .   1  17   1  <.>|
—+---------------------+

('mean accuracy', 0.5115000000000001)
```

2. <u>NOT Feature Set</u>- Along with the unigram features, NOT features is also considered for each phrase in the sample. The NOT feature is the negation of words that are followed by the negate words like not, never etc. The features are given to the Naive Bayes classifier of NLTK, using cross validation (n=5) the following are the results (accuracy and confusion matrix) obtained for the 2 different features-

- With all words in the phrases taken as the word features

```
(0, 0.505)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  18   .   . |
1 |   .  <.> 82   .   . |
2 |   .   .<202>  .   . |
3 |   .   .  73  <.>  . |
4 |   .   .  25   .  <.>|
--+---------------------+
(1, 0.5425)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  23   .   . |
1 |   .  <.> 67   .   . |
2 |   .   .<205>  .   . |
3 |   .   .  77  <.>  . |
4 |   .   .  28   .  <.>|
--+---------------------+
(2, 0.515)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  14   .   . |
1 |   .  <.> 63   .   . |
2 |   .   .<206>  .   . |
3 |   .   .  90  <.>  . |
4 |   .   .  27   .  <.>|
--+---------------------+

(3, 0.5525)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  17   .   . |
1 |   .  <.> 79   .   . |
2 |   .   .<193>  .   . |
3 |   .   .  91  <.>  . |
4 |   .   .  20   .  <.>|
--+---------------------+

(4, 0.549)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  18   .   . |
1 |   .  <.> 65   .   . |
2 |   .   .<212>  .   . |
3 |   .   .  85  <.>  1 |
4 |   .   .  19   .  <.>|
--+---------------------+
('mean accuracy', 0.547)
```

- With the 500 most common words in the phrases taken as word features

```
(0, 0.5075)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  15   3   . |
1 |   2  <9> 67   3   1 |
2 |   .   8<186>  8   . |
3 |   1   2  62  <8>  . |
4 |   .   1  20   4  <.>|
--+---------------------+

(1, 0.5225)
The confusion matrix
```

```
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  3  19   .   . |
1 |   .  <7> 52   7   1 |
2 |   .   5<189> 10   1 |
3 |   .   4  59 <12>  2 |
4 |   .   .  22   6  <.>|
--+---------------------+

(2, 0.515)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  2  11   1   . |
1 |   1  <3> 55   4   . |
2 |   4   2<194>  5   1 |
3 |   1   1  79  <9>  . |
4 |   .   1  25   1  <.>|
--+---------------------+

(3, 0.48)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  1  14   2   . |
1 |   .  <5> 65   7   2 |
2 |   .   5<181>  6   1 |
3 |   .   1  81  <6>  3 |
4 |   .   .  16   4  <.>|
--+---------------------+

(4, 0.5325)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  .  17   1   . |
1 |   3  <5> 47  10   . |
2 |   .   4<199>  9   . |
3 |   1   2  72  <9>  2 |
4 |   .   1  17   1  <.>|
--+---------------------+

('mean accuracy', 0.5115000000000001)
```

3. Bigram Feature Set- The bigram combinations of the words in a phrase are considered along with the unigram features. The Bigram Association Measures from NLTK package has been used and the measure to decide on the top bigrams used is the chi test. The features are given to the Naive Bayes classifier of NLTK, using cross validation (n=5) the following are the results (accuracy and confusion matrix) obtained for the 2 different features-

- With all words in the phrases taken as the word features

```
(0, 0.5175)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 | <.>   3   8   3   . |
1 |   . <9>  54   8   1 |
2 |   .  10<184> 10   1 |
3 |   .   5  68 <13>  . |
4 |   .   1  14   7  <1>|
--+---------------------+

(1, 0.4975)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 | <.>   5  11   2   . |
1 |   1 <7>  54   4   . |
2 |   .  12<177> 14   . |
3 |   .   5  72 <14>  3 |
4 |   .   1  12   5  <1>|
--+---------------------+

(2, 0.5675)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 | <.>   5   8   .   . |
1 |   .<13>  51   9   . |
2 |   .   5<196>  8   1 |
3 |   .   6  52 <17>  1 |
4 |   .   3  13  11  <1>|
--+---------------------+


(3, 0.565)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  4   8   1   . |
1 |   .  <6> 44   8   . |
2 |   .   8<201>  5   2 |
3 |   .   2  69 <17>  2 |
4 |   .   3  14   4  <2>|
--+--------------------+

(4, 0.48)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  6   9   .   . |
1 |   .  <6> 55   7   3 |
2 |   .   6<172> 13   2 |
3 |   .   6  74 <14>  1 |
4 |   .   .  17   9  <.>|
--+--------------------+
('mean accuracy', 0.5255)
```

- With the 500 most common words in the phrases taken as word features

```
(0, 0.5025)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  1  10   3   . |
1 |   .  <4> 61   7   . |
2 |   1   6<188>  8   2 |
3 |   1   6  70  <9>  . |
4 |   .   1  18   4  <.>|
--+--------------------+
(1, 0.49)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  5  10   3   . |
1 |   2  <2> 58   3   1 |
2 |   .   9<185>  9   . |
3 |   1   4  78  <9>  2 |
4 |   .   1  17   1  <.>|
--+--------------------+

(2, 0.5275)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  2  11   .   . |
1 |   .  <5> 59   9   . |
2 |   1   1<198> 10   . |
3 |   .   5  65  <6>  . |
4 |   .   3  18   5  <2>|
--+---------------------+
(3, 0.545)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  .  11   1   . |
1 |   .  <3> 52   3   . |
2 |   .   4<203>  8   1 |
3 |   .   2  79  <9>  . |
4 |   .   2  14   5  <2>|
--+---------------------+

(4, 0.4675)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  1  10   4   . |
1 |   .  <3> 56  10   2 |
2 |   .   5<174> 13   1 |
3 |   .   2  83  <9>  1 |
4 |   .   1  21   3  <1>|
--+---------------------+
('mean accuracy', 0.5065)
```

4. POS tagging Feature Set- The POS tags that have be used are Adverbs, Adjectives, Verbs and Nouns. The tags for each word in the phrase are also considered to enhance the feature set. The features are given to the Naive Bayes classifier of NLTK, using cross validation (n=5) the following are the results (accuracy and confusion matrix) obtained for the 2 different features-

- With all words in the phrases taken as the word features

```
(0, 0.535)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  7  16   2   . |
1 |   . <15> 45   8   1 |
2 |   2  14<175> 11   3 |
3 |   .  9  51 <23>  1 |
4 |   .  3  10   3  <1>|
--+--------------------+

(1, 0.4825)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  6   4   2   1 |
1 |   .  <7> 63  16   . |
2 |   .  9<169> 14   1 |
3 |   .  7  56 <17>  . |
4 |   .  2  18   8  <.>|
--+--------------------+

(2, 0.5)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <.>  6  11   3   . |
1 |   . <10> 54   9   1 |
2 |   .  5<176> 12   . |
3 |   .  6  71 <12>  . |
4 |   .  2  15   5  <2>|
--+--------------------+

(3, 0.58)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <1>  1   6   1   . |
1 |   . <16> 49   4   . |
2 |   . 11<200>  9   4 |
3 |   1   5  56 <15>  1 |
4 |   .  1  13   6  <.>|
--+--------------------+

(4, 0.59)
The confusion matrix
  |   0   1   2   3   4 |
--+--------------------+
0 |  <1>  2   6   2   1 |
1 |   . <14> 36   9   1 |
2 |   . 12<198> 16   . |
3 |   1   5  50 <23>  1 |
4 |   .   2  13   7  <.>|
--+---------------------+

('mean accuracy', 0.5375)
```

- With the 500 most common words in the phrases taken as word features

```
(0, 0.51)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  4  15   6   . |
1 |   .  <9> 46  14   . |
2 |   3  12<176> 12   2 |
3 |   2   6  50 <18>  8 |
4 |   1   3   9   3  <1>|
--+---------------------+
(1, 0.4675)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  6   4   2   1 |
1 |   3  <9> 57  17   . |
2 |   1   7<167> 15   3 |
3 |   4   8  55 <11>  2 |
4 |   1   4  15   8  <.>|
--+---------------------+
(2, 0.4775)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  3  14   2   1 |
1 |   2  <7> 50  14   1 |
2 |   .  14<167> 12   . |
3 |   1   6  65 <17>  . |
4 |   2   2  16   4  <.>|
--+---------------------+

(3, 0.565)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <2>  2   4   1   . |
1 |   3 <13> 46   6   1 |
2 |   2  10<195> 15   2 |
3 |   4   5  55 <14>  . |
4 |  .  2 10  6 <2>|
--+----------------------+

(4, 0.565)
The confusion matrix
  |  0   1   2   3   4 |
--+----------------------+
0 | <1>  1   6   3   1 |
1 |  3 <16> 37   4   . |
2 |  .  13<192> 21   . |
3 |  2  10  48 <17>  3 |
4 |  .   2  11   9  <.>|
--+----------------------+
('mean accuracy', 0.517)
```

5. Sentiment Lexicon: Subjectivity Count Features- With this particular feature set function along with uniform features, 2 additional features positive count and negative count are added to the feature set. For this feature set the definition of the readSubjectivity function from the Subjectivity.py module which is provided by Professor is being used. It creates a Subjectivity Lexicon that is represented here as a dictionary, where each word is mapped to a list containing the strength and polarity, mainly how the SL feature is defined. The features are given to the Naive Bayes classifier of NLTK, using cross validation (n=5) the following are the results (accuracy and confusion matrix) obtained for the 2 different features-

- With all words in the phrases taken as the word features

```
(0, 0.525)
The confusion matrix
  |  0   1   2   3   4 |
--+----------------------+
0 | <.>  6  16   3   . |
1 |  . <11> 50   7   1 |
2 |  .  12<176> 14   3 |
3 |  .  11  51 <22>  . |
4 |  .   .  10   6  <1>|
--+----------------------+

(1, 0.55)
The confusion matrix
  |  0   1   2   3   4 |
--+----------------------+
0 | <.>  8   5   .   . |
1 |  .  <9> 68   9   . |
2 |  .   5<175> 13   . |
3 |   .   5  54 <20>  1 |
4 |   .   1  18   9  <.>|
--+---------------------+

(2, 0.53)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  7  12   1   . |
1 |   . <11> 55   7   1 |
2 |   .   5<182>  6   . |
3 |   .   6  73 <10>  . |
4 |   .   2  15   6  <1>|
--+---------------------+

(3, 0.575)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  1   6   1   . |
1 |   . <12> 49   8   . |
2 |   .  13<202>  6   3 |
3 |   .   5  57 <15>  1 |
4 |   .   .  11   9  <.>|
--+---------------------+

(4, 0.5825)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  3   7   1   1 |
1 |   . <11> 42   7   . |
2 |   .  11<202> 13   . |
3 |   1   5  54 <20>  . |
4 |   .   1  12   9  <.>|
--+---------------------+
('mean accuracy', 0.5405)
```

- With the 500 most common words in the phrases taken as word features

```
(0, 0.555)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  6   8   .   . |
1 |   2  <5> 57   9   . |
2 |   1   2<192> 11   . |
3 |   1   4  54 <24>  . |
4 |   .   .   9  14  <1>|
--+---------------------+

(1, 0.5225)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  4  11   5   1 |
1 |   .  <9> 53   6   . |
2 |   1   6<181> 16   2 |
3 |   1   8  49 <17>  3 |
4 |   .   1  16   8  <1>|
--+---------------------+

(2, 0.515)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <1>  4   9   5   . |
1 |   1 <15> 37  11   1 |
2 |   .  11<170> 13   1 |
3 |   2  11  58 <19>  6 |
4 |   1   1   7  15  <1>|
--+---------------------+

(3, 0.525)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <.>  2  12   4   . |
1 |   1  <6> 55   9   1 |
2 |   .   7<186> 12   1 |
3 |   1   6  54 <16>  3 |
4 |   .   .  11  11  <2>|
--+---------------------+

(4, 0.5425)
The confusion matrix
  |   0   1   2   3   4 |
--+---------------------+
0 |  <2>  2   7   .   1 |
1 |   .  <3> 55  13   . |
2 |   .   6<184> 19   2 |
3 |  2  6 47 <26>  2 |
4 |  .  1 12  8  <2>|
--+--------------------+

('mean accuracy', 0.532)
```

Comparison

Out of all the different feature set functions that have been used, the accuracy of the feature set functions can be viewed in the table below. This is will help us analyze which feature set is performing better. Also, the above results obtained are for sample size of 2000. It also has been tested with a sample size of 5000 and 9000 and the following are the results which have been obtained-

Feature Set	Accuracy with all words as features in %	Accuracy with 500 most common words as features in %
Unigram Feature Set(Sample size= 2000)	53.1	51.1
Unigram Feature Set(Sample size= 5000)	53.6	51.4
Unigram Feature Set(Sample size= 9000)	54.9	52.01
Not Feature Set(Sample size= 2000)	54.7	51.5
Not Feature Set(Sample size= 5000)	55.1	52.1
Not Feature Set(Sample size= 9000)	55.4	52.3
Bigram Feature Set(Sample size= 2000)	52.5	50.65
BIgram Feature Set(Sample size= 5000)	52.6	50.5
Bigram Feature Set(Sample size= 9000)	52.9	50.6
POS tag Feature Set(Sample size= 2000)	53.75	51.7
POS tag Feature Set(Sample size= 5000)	53.8	51.7
POS tag Feature Set(Sample size= 9000)	54.1	51.8

Feature Set	Accuracy with all words as features in %	Accuracy with 500 most common words as features in %
SL Feature Set(Sample size= 2000)	55.25	53.2
SL Feature Set(Sample size= 5000)	55.7	53.9
SL Feature Set(Sample size= 9000)	55.8	54.1

Among the different feature set functions, the Sentiment Lexicon and the Not features seem to be doing a better job at classifying the sample. The actually accuracy according to the Kaggle competition leader boards was around 62% and our model is close to this score in case of Sentiment Lexicon Subjectivity scores. Also, the accuracy is more when all the words are considered when compared to the 500 most common words. Since, the model is being trained on the sample it is performing good, it could perform better if it where trained on the entire dataset as it can be seen that as the sample size increases, accuracy also seems to be increasing.

Training with Weka

The csv file created with the unigram features "trainWeka.csv" has been loaded into weka and different classifier techniques have been applied on it. Previously the NLTK Naive Bayes has been applied to train the model. Now, the techniques in Weka that have been applied are J48, Ada Boost, Logistic Regression and ZeroR. These classifiers are applied on a sample size of 5000 and the word features used are 200. The results of these classifiers which performed better then others are stored in the "resultBuffers" folder and brief summary of the observations is given below-

Technique	Accuracy	Precision	Recall	F-measure
AdaBoost	52.14%	0.272	0.521	0.358
J48	52%	0.374	0.520	0.366
Bagging	48.24%	0.347	0.482	0.375
Logistic	51.96%	0.409	0.520	0.376
Naive Bayes	51.98%	0.405	0.520	0.374
ZeroR	52.16%	0.272	0.522	0.358

Among the various techniques used for unigram and word-vector features, the ZeroR and AdaBoost technique seem to be slightly doing better than NLTK naive babes classifier. So, when given the NOT features set and the SL features, the performance could be improved upon the results that have been obtained using the NLTK Naive Bayes classifier.

Conclusion

After preprocessing the data, the accuracy of our model has increased by 3% and hence it can be seen that preprocessing of the phrases is a crucial step in building our model. Also, among the various feature sets, NOT features and SL features seem to giving better results. It can also be seen that other than Naive Bayes, other techniques in Weka like AdaBoost and ZeroR seem to be performing better with the sample phrases. Our best model has achieved an overall accuracy of 56% which is good results.