UNIVERSITY OF ABERDEEN

Machine Learning

CS5062

Continuous Assessment on Sentiment Analysis

Name: Avinash Vinayak Bagul.

Student ID: 51987820

Course: Degree of Master of Science in Artificial Intelligence

Index

Introduction

Task Description:

Sentiment Analysis Assessment is a structured and well organized task for analysing the given python code for sentiment analysis on Film reviews and Nokia Phone revies. Collection of the positive and negative words are given in a text file. Two different approaches are used in code to analyse the text reviews, These approaches are Naïve Bayes and Dictionary approach i.e. statistical and rule based systems are used for sentiment analysis.

What is Sentiment Analysis?

Sentiment analysis is a process of identifying and categorizing emotions computationally in a text. To determine the attitude of writer towards a specific topic or product. Sentiments in a text can be categorised as [positive, negative, neutral]. Sentiment analysis use AI technique to identify polarity of the text. Sentiment analysis is mostly used in social media, product reviews, survey responses and making data driven decisions.

There are may types of sentiment analysis, few of them are listed below:

- ✓ Fine-grained sentiment analysis.
- ✓ Emotion Detection.
- ✓ Aspect based sentiment analysis.
- ✓ Intent Analysis
- ✓ Multilingual sentiment analysis.

Three types of algorithms are used for sentiment analysis: Rule-based system, Automatic system and Hybrid system. And following classification algorithms can be used for sentiment analysis:

- ✓ Naïve Bayes.
- ✓ Linear Regression.
- ✓ SVM.
- ✓ Deep Learning.

Why is Sentiment Analysis Interesting?

I find sentiment analysis interesting because it can extract the meaning of the sentence from the given data, which can be used for many applications. Apart from this, Sentiment analysis is very challenging. Natural language is ambiguous and can have syntactic and semantic ambiguity. Negation is a challenge, understanding sarcasm and irony is another challenge. Few words in sentence can change the whole meaning of the sentence but most of the sentiment analysis applications does not cover this.

Example: “Time given to solve assessments is not sufficient”

If sufficient is a positive word then there is good chance that sentiment for this sentence will be wrongly classified as positive.

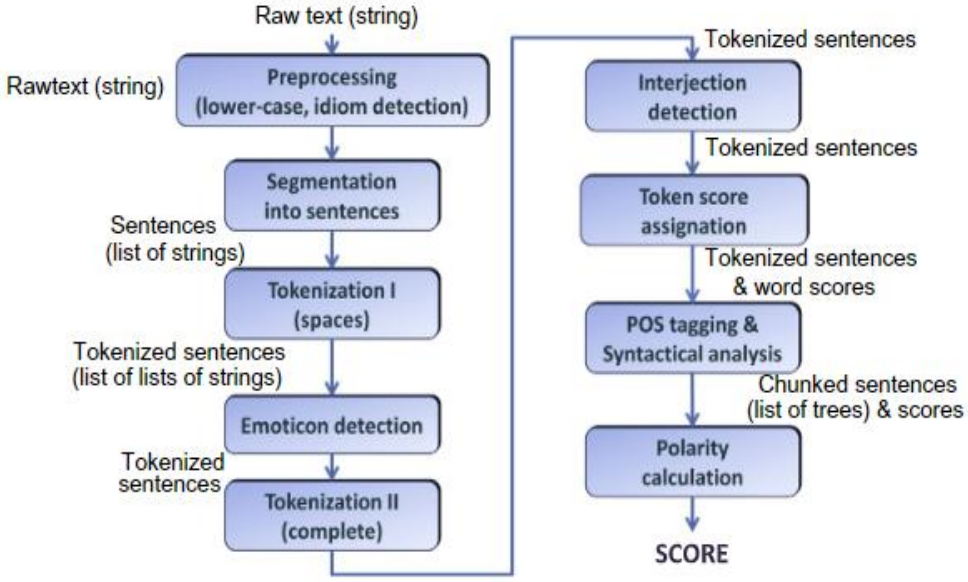
What Kind of methods people use for this task?

In paper ”Social media sentiment analysis: lexicon versus machine learning”
Dhaoui, Chedia; Webster, Cynthia M; Tan, Lay Peng.The Journal of Consumer Marketing; Santa Barbara Vol. 34, Iss. 6, (2017): 480-488. DOI:10.1108/JCM-03-2017-2141

Key approaches to automated sentiment classification:

For achieving high accuracy in content classification for social media it is crucially important to select a right sentiment analysis method. Two prominent approaches are available for text classification, lexicon-based and machine learning. Both techniques classify the given text data into positive, negative and neutral. Lexicon based method depends on a dictionary on opinion words also known as a sentiment dictionary or a sentiment lexicon to classify the sentiment as positive or negative.

Another approach is using Machine Learning, RTextTools is a package in R language for automatic text xlassification. Package includes maximum entropy, SVM, bagging, decision tree and random forest, etc.



Fig(1.1)

Fig(1.1) describes the procedure of how score is calculated for classifying sentiment for a sentence.

Paper “Feature engineering for sentiment analysis in e-health forums”:
Carrillo-de-Albornoz, Jorge; Javier Rodríguez Vidal; × Laura Plaza ×.PLoS One; San Francisco Vol. 13, Iss. 11, (Nov 2018): e0207996. DOI:10.1371/journal.pone.0207996

Sequential Minimal Optimization (SMO), an algorithm to implement Support Vector Machines Which solve quadratic programming problem which is encountered while training model. Support Vector Machine is a binary classifier which does not rely on probability. SVM try to achieve best data separation with hyperplane. It was found that SVM performed better than Naïve Bayes and Random forest. Sequential Minimal Optimization produces balanced precision and recall.

Grammar is also used as feature of classification in sentiment analysis. Verb tense, part of speech and negation.

Verb tense: Facts are expressed using tense verb, it was proposed to check whether this feature helps indirectly in predicting polarity.

Part of speech: classification features also include frequency of nouns, adverbs and adjectives.

Negation: it's a binary feature which points towards negation in the sentence. It is detected with the help of negation tokens.

DESIGN

Rule-based systems means manually designed rules for sequence of words, or a different way of representing words in a sentence, and match these sequences with the text data. Rule-based systems indirectly specify a mathematical model. Rule-based system is not always the best but it can be combined with statistical approach used by Machine Learning Algorithms to give better accuracy.

- Existing Rule-based System:

testDictionary() is an rule-based classifier which does not produce good accuracy for Film Dataset. To obtain better accuracy we need to implement few changes to the rule based system or add new rules to make good predictions.

- New Rule-based System :

It's crucially very important to understand how rule-based systems work for sentiment analysis before designing new system for better metric results.

Rule based system mainly depends on the human generated rule to solve a particular problem, this mathematical rules help the program to decide the correct classification for the given sentence. In existing system, Accuracy metric depend on total score of the sentence and the user defined threshold. Same will be the case for our new system.

Understanding semantics:

Let's consider an example, suppose we have a review " Nokia phone is good for price but it has problem with camera and the phone often hangs", so in case if number of positive words in sentence are more than negative words will give us positive score, after looking at sentence

we can say that the review is negative. Phone has one good features and 2 bad features which logically makes the review negative.

In such cases, results will be wrong and it will affect the metric scores. So what to do to avoid such mistake that is relying just on the number of positive and negative words in the sentence.

“but” is neither positive nor negative but it changes the meaning of the sentence completely. So if we split the list of sentences based on the word “but” in data and save the left side split in a list. For our example sentence left_split list will contain following words:

Left_split = [‘Nokia’,’phone’,’is’,’good’,’for’,’price’]

This list contains all the words in the sentence before “but” word. Now we will check the score for number of positive words and negative words in the Left_split, and we find that score is positive. Use of “but” word always change the meaning of sentence on right split, if score on the left side is positive that means whole review sentence has negative sentiment.

Similar to above approach, I can think of many such words which change the meaning of sentences.

If there is no “not” word in positive-words.txt and negative-words.txt but when we use not before a word which is in the dictionary it should change the polarity of the word and update the score accordingly.

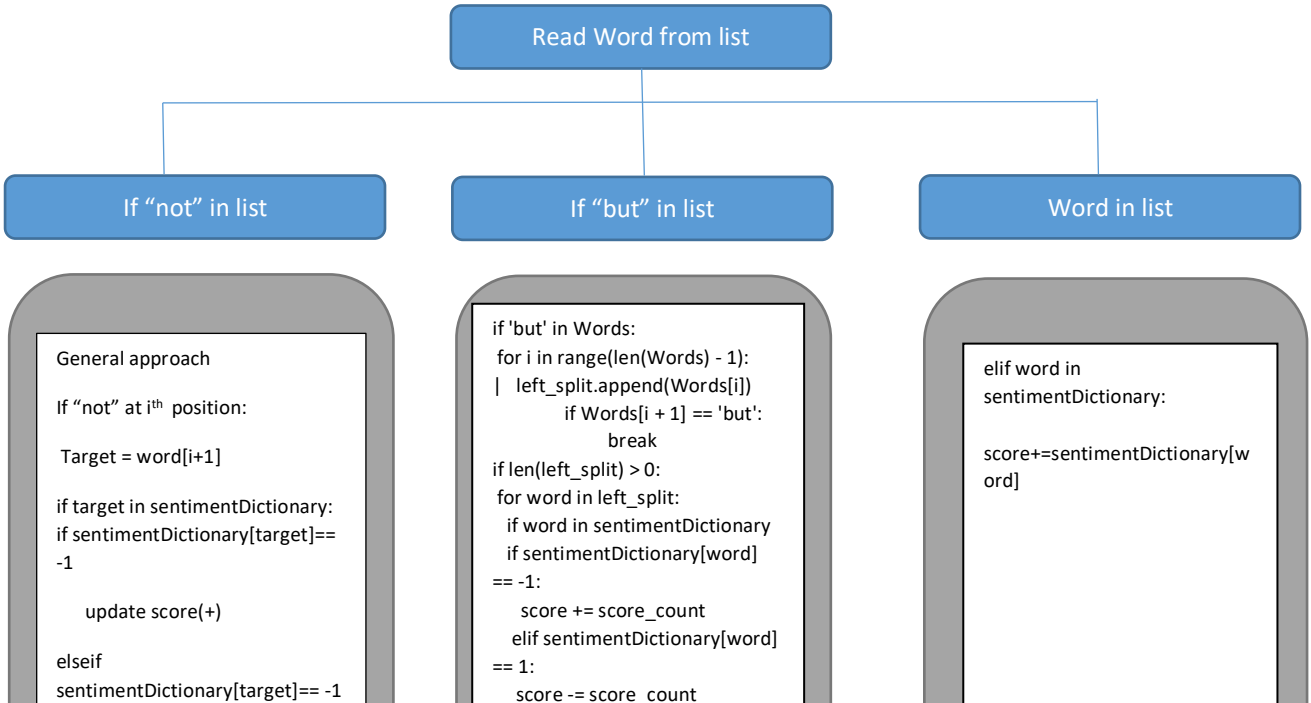
Example:

Good = Positive word in sentiment.

If we add ‘not’ in front of ‘Good’ then it changes the polarity of the context.

Not Good = negative sentiment.

So if we add rules which include the above points should give us good accuracy for classification of sentiment. We may think on implementing bigrams and trigrams if they improve the final accuracy for given dataset.



Above is the way we can improve our rules to get more accuracy.

After implementing the above rules for score calculation I found the following result for the Nokia phone Dataset:

```
testD(sentencesNokia, "Nokia   (All Data, Rule-Based)\t", sentimentDictionary, 0)
```

```
Nokia   (All Data, Rule-Based)   Accuracy (All)=0.79 (210/266)

Nokia   (All Data, Rule-Based)   Precision (Pos)=0.80 (175/220)
Nokia   (All Data, Rule-Based)   Recall (Pos)=0.94 (175/186)
Nokia   (All Data, Rule-Based)   F-measure (Pos)=0.86
Nokia   (All Data, Rule-Based)   Precision (Neg)=0.76 (35/46)
Nokia   (All Data, Rule-Based)   Recall (Neg)=0.44 (35/80)
Nokia   (All Data, Rule-Based)   F-measure (Neg)=0.56
```

Implementing Bigrams and Trigrams:

I think implementing bigrams and trigrams won't help in our case because our system is designed for dictionary based approach, every word is compared with the word in sentiment dictionary. If there is no bigrams and trigrams in the sentiment dictionary, score will not be updated.

Use of Negation tokens can help detecting negation in the sentence, if negation is tackled and correctly addressed then accuracy for my new system can improve.

Analysis

Sentiment analysis is extracting the emotions and subjective information from the text reviews, survey and social media interactions using Natural Language Processing, Text Analysis and Computational Linguistics.

- **Task 1:**

This task familiarize with the data on which the Sentiment analysis depends. Assessment provides three different categories of data i.e., Movie reviews from rotten tomato website, Reviews on Nokia phones and a sentiment dictionary with positive and negative sentiment words, with each category of data containing two different files, one with positive reviews/word and another one with negative reviews/words. Below is the description on given files.

Name of file	Number of data
rt-polarity.pos.txt	5331 (reviews)
rt-polarity.neg.txt	5331 (reviews)
nokia-pos.txt	193 (reviews)
nokia-neg.txt	79 (reviews)
positive-words.txt	2006 (words)

negative-words.txt	4783 (words)

Table(4.1.1)

SentimentPract2.py is the python file given in assessment which implements Naive Bayes classifiers to classify the given sentence as positive or negative. After going through the python file I discovered that regular expressions are playing important role in splitting and separating the data. Text files are filtered and the data is stored in new lists. Below is the screenshot for raw text file.

```
;    1. The appearance of an opinion word in a sentence does not necessarily
;       mean that the sentence expresses a positive or negative opinion.
;       See the paper below:
;
;       Bing Liu. "Sentiment Analysis and Subjectivity." An chapter in
;          Handbook of Natural Language Processing, Second Edition,
;          (editors: N. Indurkhya and F. J. Damerau), 2010.
;
;    2. You will notice many misspelled words in the list. They are not
;       mistakes. They are included as these misspelled words appear
;       frequently in social media content.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

2-faced
2-faces
abnormal
abolish
abominable
abominably
abominate
abomination
abort
```

Below is the code for reading words from text file not starting with ';' sign and storing it into a new list (negWordList).

```
negDictionary = open('negative-words.txt', 'r', encoding="ISO-8859-1")
negWordList = negDictionary.readlines()
negWordList = [line.strip() for line in negWordList if not line.startswith(";") and not line == '\n']
```

➢ Task 2:

This task is about understanding the code in SentimentPract2.py and running this python file to get the result Accuracy, Precision, Recall and F-measure for the test data.

Evaluation metric	figures
Accuracy (All)	0.79
Precision (Positive)	0.78
Recall (Positive)	0.79
F-measure (Positive)	0.79
Precision (Negative)	0.80
Recall (Negative)	0.78
F-measure (Negative)	0.79

Table(4.2.1)

❖ Accuracy:

We get the total accuracy 0.79 after dividing the correct classified test data and the total test data. Accuracy tell us about the efficiency of the model and in our program the accuracy of the model is around 79% (which is good but can be improved further for better analysis).

❖ Precision:

But we cannot judge the model only on accuracy, it has some drawbacks. There is high cost of having mis-classified actual positive or false negative, So, Precision, recall and F-score is used for further evaluation of model. To get the Precision score for positive, correctpos (true positive) was divided by totalpospred (total number of positive predicted). Similarly Precision for negative, correctneg (true negative)/ totalnegpredicted (total predicted negative).

❖ Recall:

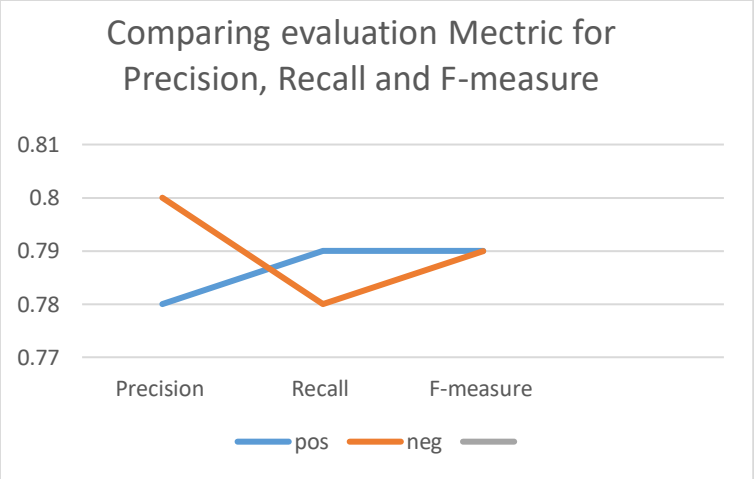
Recall actually calculates number of the Actual Positives that are capture by our model through labelling it as Positive (True Positive). So if the review for movie is good (actual positive) and goes through the test and it is predicted as bad review (predicted negative). Cost associated with this outcome can be extremely bad. Recall can also be used to select the best model. Recall is measured by the following formula:

❖ F-measure: (f1-Score)

F1 score is used when we want balance between precision and recall. F1 score and accuracy score are mostly same, F1 score is better than accuracy if there is uneven class distribution (more number of actual negatives). Following formula was used to obtain the f1 score:

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

The graph below shows the precision, recall and f1 score for positives and negatives.



Graph(4.2.1)

➢ **Task 3:**

Task three was about running Naïve Bayes on training data and Nokia phone reviews. After uncommenting line number 330 and 332 in SentimentPract2.py I saw the following results:

Accuracy, Precision, Recall, F-measure after applying Naïve Bayes on Films training data,

Films training data on Naive Bayes	
Accuracy	0.93
Precision (positive)	0.94
Recall (Positive)	0.92
F-measure (Negative)	0.93
Precision (Negative)	0.92
Recall (Negative)	0.94
F-measure (Negative)	0.93

Table(4.3.1)

After running program for several times, I found that passing film training data to Naïve Bayes give constant results for every run and accuracy is really good i.e., 0.93 (93%).Overall comparison with output of film test data, output accuracy, precision, recall and f-measure is better. After experimenting with the pPos(fraction of positive reviews) and changing the values give output accuracy in range (0.91-0.93) for different fraction value. The reason behind good accuracy is that the model is already trained on the data so the prediction is really good.

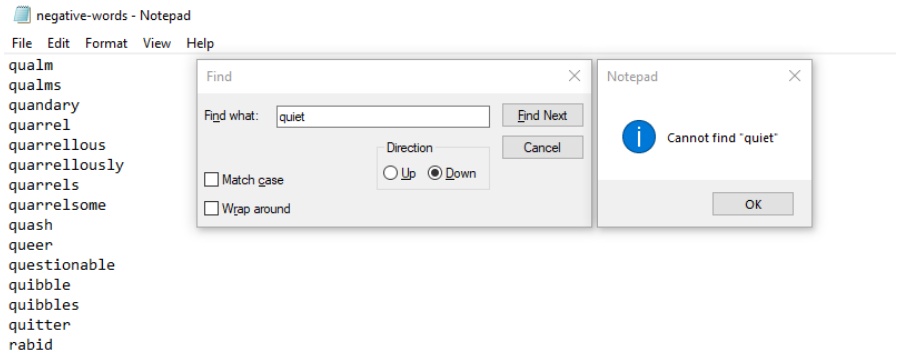
Unlike Film training data, Nokia Review data when passed to Naïve Bayes produce random output for Accuracy, Precision, Recall and F-measure for each individual run. Output accuracy value is always around 0.59. One more thing which I noticed is Precision and F-measure for negative are comparatively lower than positive.

Nokia Reviews on Naive Bayes	
Accuracy	0.59
Precision (positive)	0.81
Recall (Positive)	0.54
F-measure (Negative)	0.65
Precision (Negative)	0.40
Recall (Negative)	0.70
F-measure (Negative)	0.51

Table(4.3.2)

Nokia data is having less accuracy and according to my observation Nokia Data is never trained using trainBayes() method so the accuracy is around 50% and 60% but I also think that, most of the negative words in Nokia phone reviews are not present in the dictionary i.e., negWordList.

So, I think the reason behind poor result for Nokia phone reviews is lack of negative words in dictionary which are used in the review. For Example (“I have excellent hearing but the volume level on this phone is especially quiet.”) this is a negative review from the nokia-neg file.



The word which makes the review negative is word ‘quiet’, volume level on phone is especially quiet, but there is no ‘quiet’ word or any other word from the sentence in the negative-words file.

This problem can be addressed with the use of bigram and trigrams where words together make some sense, instead of a single word ‘quiet’, ‘quiet volume level’ makes more sense as negative review in case of Nokia phone, because word ‘quiet’ can be used for good reviews as well.

- **Task 4:**

The most useful words for predicting sentiments are the words which can make great impact on the prediction. Words which repeatedly occur in the dataset of positive or negative sentences may be useful to predict the sentiment. So technically the words with more prediction value are useful for sentiment analysis.

As Naïve Bayes classifier is one of the best approach to address sentiment analysis. Other algorithms such as Linear Regression, Support Vector Machine (SVM) and Deep Learning can also be used for sentiment analysis. Naïve Bayes is the only probabilistic model that uses Bayes theorem to predict the category of text. This classifier assume that value of particular factor is independent of another feature.

trainBayes(sentencesTrain, pWordPos, pWordNeg, pWord) calculate probability of P(word | positive), P(word | negative) and P(word) with the help of freqPositive, freqNegative, negWordsTot, posWordsTot and allWordsTot. Based on these probability results testBayes works and predicts the sentiment for the sentence.

After uncommenting the mostUseful() function I got list of positive words and negative words as output.

```
In [111]: # print most useful words
          mostUseful(pWordPos, pWordNeg, pWord, 50)

NEGATIVE:
['boring', 'generic', 'badly', 'unfunny', 'waste', 'mediocre', 'ill', 'routine', 'poorly', 'disguise', 'mindless', 'should_have
_been', 'pointless', 'of_the_characters', 'tiresome', 'the_only_thing', 'meandering', 'apparently', 'annoying', 'offensive', 'p
lodding', 'bore', 'dull', 'stupid', 'unless', 'junk', 'inept', 'stale', 'dreary', 'harvard', 'wasted', 'shoot', 'product', 'lif
eless', 'pinocchio', 'numbers', 'supposed_to_be', 'chan', 'retread', 'pathetic', 'trite', 'sadly', 'seagal', 'flat', 'by_the_nu
mbers', 'supposed', 'wants_to_be', 'stealing', 'collection', 'if_it_were']

POSITIVE:
['literary', 'richly', 'subversive', 'has_created_a', 'format', 'captivating', 'capture', 'unexpected', 'jealousy', 'the_same_t
ime', 'grown', 'heartwarming', 'record', 'film_that_is', 'movie_with_a', 'warm', 'provides', 'tender', 'iranian', 'resonant',
'tour', 'sides', 'respect', 'nicely', 'makes_up_for', 'portrait_of_a', 'wonderfully', 'playful', 'detailed', 'spare', 'polishe
d', 'vividly', 'dazzling', 'wonderful', 'lively', 'wry', 'realistic', 'mesmerizing', 'chilling', 'powerful', 'riveting', 'gem',
'vivid', 'refreshingly', 'haunting', 'refreshing', 'inventive', 'captures', 'engrossing', 'of_the_best']
```

This selected words by model are good for sentiment prediction because this words have highest prediction value based on the probability. After performing if else conditions on the pWordPos and pWordNeg, predict power is computed/declared and then the sorting is done for positive and negative words.

Hence, this words are useful for sentiment prediction.

I created a small program to calculate how many of these selected word by model are in sentiment dictionary used in Dictionary based approach. This program just check each word in the model generated list with the words in Sentiment Dictionary, if word matches then counter variable is incremented. This process is done separately for Positive words and negative words.

We have different results(counts) for unigrams, bigrams and trigrams, see table 4.4.1.

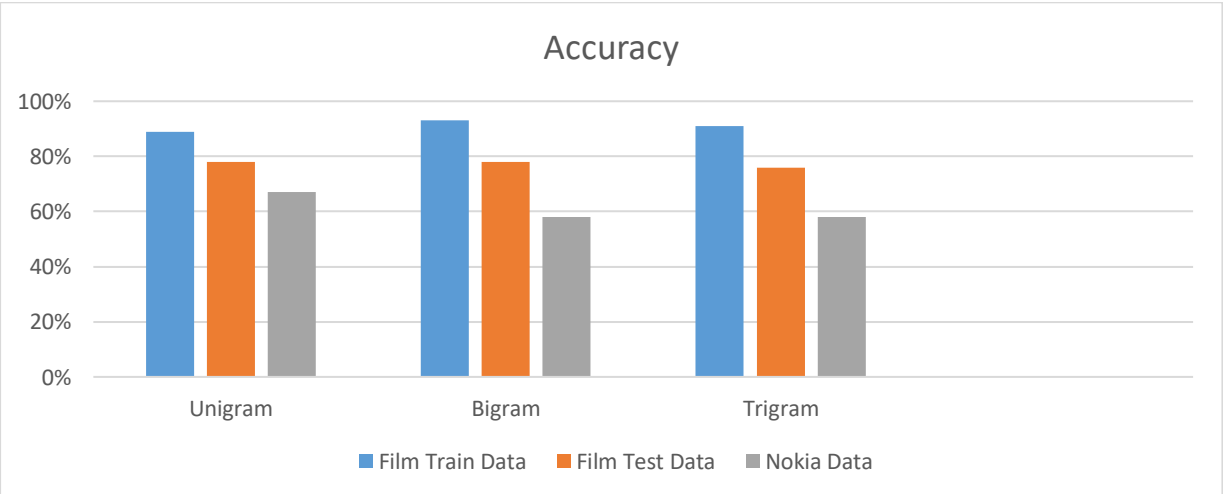
words	count	Total useful words
Unigram:		
Positive words in Sentiment Dictionary	28	50
Negative words in Sentiment Dictionary	26	50
Bigrams:		
Positive words in Sentiment Dictionary	18	50
Negative words in Sentiment Dictionary	19	50
Trigrams:		
Positive words in Sentiment Dictionary	30	50
Negative words in Sentiment Dictionary	25	50

Table(4.4.1)

I created trigram list using similar approach used for bigram in trainBayes() and testBayes(), and I don't see an effective improvement to the model because the results are similar to Unigram and Bigram. Use of trigram does not give better results than bigram.

```
trigramList=wordList.copy()
for x in range(len(wordList)-2):
    trigramList.append(wordList[x]+"_"+ wordList[x+1]+"_"+ wordList[x+2])
```

I found the following results after implementing Unigram, Bigram and Trigram separately for testBayes( ) classifier at different runs:



Graph(4.4.1)

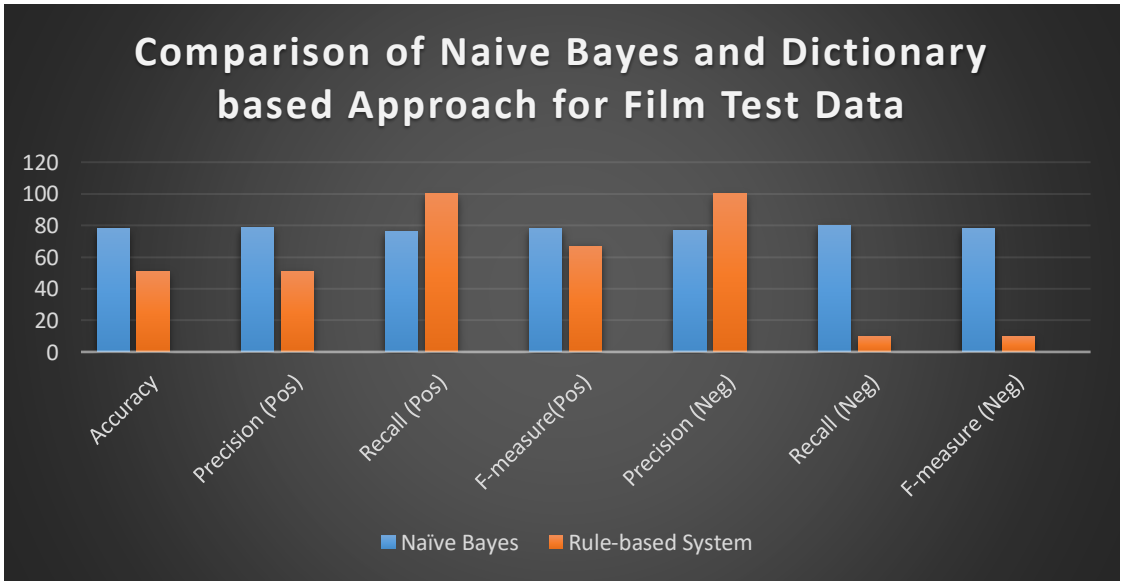
- **Task 5:**

Dictionary-based sentiment analysis is a computational approach for predicting the feeling that a text data i.e., Film review and Nokia phone review. In our case, sentiment has a binary classification: positive or negative. Dictionary based method relies on a pre-defined list (or dictionary) of Positive and Negative words. After uncommenting the call to testDictionary(), I encountered an error for Nokia data i.e., divide by zero error after fixing it(changing threshold), following results were generated by testDictionary():

```
In [508]: #run sentiment dictionary based classifier on datasets
          testDictionary(sentencesTrain,  "Films (Train Data, Rule-Based)\t", sentimentDictionary, -4)
          testDictionary(sentencesTest,  "Films  (Test Data, Rule-Based)\t",  sentimentDictionary, -4)
          testDictionary(sentencesNokia, "Nokia   (All Data, Rule-Based)\t", sentimentDictionary, -3)

Films (Train Data, Rule-Based)    Accuracy (All)=0.51 (4835/9565)

Films (Train Data, Rule-Based)    Precision (Pos)=0.50 (4795/9518)
Films (Train Data, Rule-Based)    Recall (Pos)=1.00 (4795/4802)
Films (Train Data, Rule-Based)    F-measure (Pos)=0.67
Films (Train Data, Rule-Based)    Precision (Neg)=0.85 (40/47)
Films (Train Data, Rule-Based)    Recall (Neg)=0.01 (40/4763)
Films (Train Data, Rule-Based)    F-measure (Neg)=0.02

Films  (Test Data, Rule-Based)    Accuracy (All)=0.48 (530/1099)

Films  (Test Data, Rule-Based)    Precision (Pos)=0.48 (527/1093)
Films  (Test Data, Rule-Based)    Recall (Pos)=0.99 (527/530)
Films  (Test Data, Rule-Based)    F-measure (Pos)=0.65
Films  (Test Data, Rule-Based)    Precision (Neg)=0.50 (3/6)
Films  (Test Data, Rule-Based)    Recall (Neg)=0.01 (3/569)
Films  (Test Data, Rule-Based)    F-measure (Neg)=0.01

Nokia   (All Data, Rule-Based)    Accuracy (All)=0.70 (186/266)

Nokia   (All Data, Rule-Based)    Precision (Pos)=0.70 (186/266)
Nokia   (All Data, Rule-Based)    Recall (Pos)=1.00 (186/186)
Nokia   (All Data, Rule-Based)    F-measure (Pos)=0.82
Nokia   (All Data, Rule-Based)    Precision (Neg)=0.00 (0/0)
Nokia   (All Data, Rule-Based)    Recall (Neg)=0.00 (0/80)
Nokia   (All Data, Rule-Based)    F-measure (Neg)=0.00
```

Results are very poor in accuracy for film train data and film test data and reason which I think is responsible for unsatisfactory result is weak Rule-based system. A Rule-based system is a system which performs sentimental analysis is based on set of rules which are manually written by programmer. Few changes to the rule-based system can give us better output as compared to current output.

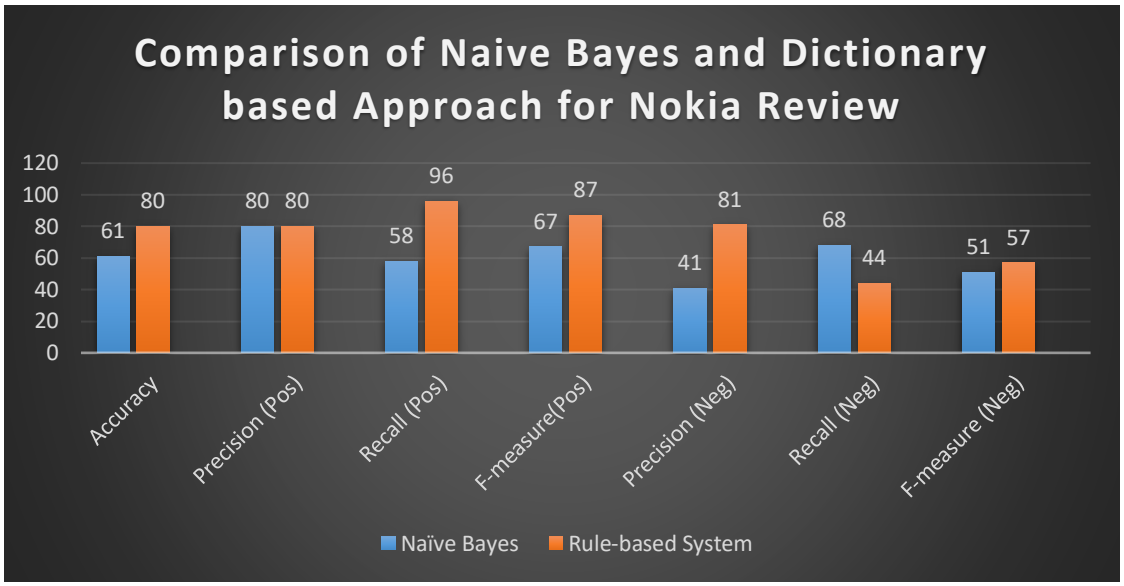
Comparison for Naïve Bayes and Dictionary Based system for Film review data is described with the use of graph(4.5.1).



Graph(4.5.1)

These is the comparison of the obtained result with the use of bar graph, here blue bar indicates result for Naïve Bayes and orange bar indicates Rule-based approach i.e., testDictionary(). Performance of Dictionary approach for film review data is poor and we can

see lot of difference between scores of matrices used. This shows that our rule-based system is not satisfactory and need to be improved, to improve rule based system we can add few rules to make prediction more reliable and correct.



Graph(4.5.2)

Based on these results I have learnt that statistical models use machine learning algorithms such as Naïve Bayes, SVM or Deep Learning techniques which generate better results that Rule-based model which is based on set of rules. But Rule-based models can be used in places where specific information is to be extracted from user data. If this both techniques are combined with each other, they can make a strong system for sentiment analysis.

Improvement to new Rule-based system:

After making few changes for rule based system I managed to get 13% more accuracy than the existing test Dictionary. But still accuracy for the new system is not satisfactory. After adding some rules which makes semantic logic clear and gives better accuracy that existing system.

- Following are the results for new system on Nokia Review:

```
Nokia   (All Data, Rule-Based)  Accuracy (All)=0.79 (210/266)

Nokia   (All Data, Rule-Based)  Precision (Pos)=0.80 (175/220)
Nokia   (All Data, Rule-Based)  Recall (Pos)=0.94 (175/186)
Nokia   (All Data, Rule-Based)  F-measure (Pos)=0.86
Nokia   (All Data, Rule-Based)  Precision (Neg)=0.76 (35/46)
Nokia   (All Data, Rule-Based)  Recall (Neg)=0.44 (35/80)
Nokia   (All Data, Rule-Based)  F-measure (Neg)=0.56
```

- **Task 6:**

After setting PRINT_ERRORS = 1, and uncommenting calls to testDictionary() & testBayes( ) except for the call to testBayes() with film test data, I found following errors:

```
ERROR (pos classed as neg 0.21):
ERROR (neg classed as pos 0.73):
```

This is not an actual error, but we print this Error with the probability count (prob), when the s entiment for the sentence is positive and still the predicted value for probability of (pPosW/ (pPosW+pNegW)) is less the 0.5, we get error as: (pos classed as neg <prob>). Similarly when the sentiment for the sentence is negative and the calculated probability is gre ater than 0.5, prints the error (neg classed as pos <prob>).

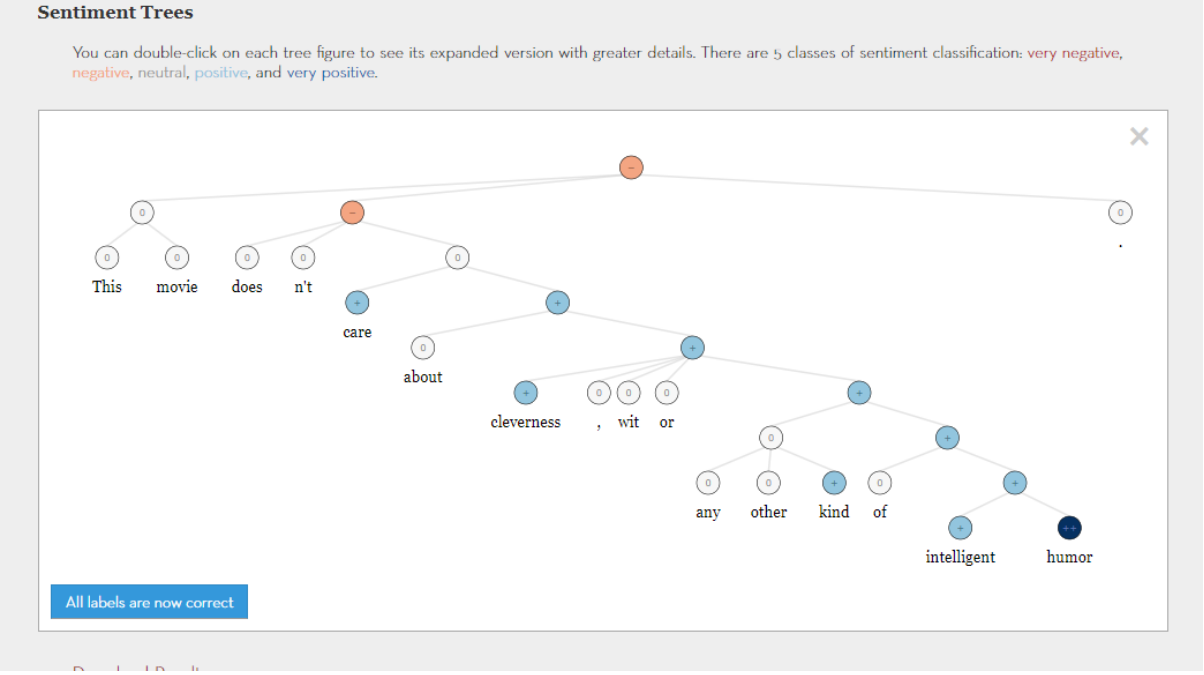
Let us understand with an example: we have a review “if this movie were a book, it would be a page-turner , you can't wait to see what happens next .”
This review is labeled as positive in Dictionary. But when we calculate probability based on pPosW and pNegW, we get probability less than 0.5.
Mostly sentences with negation are not classified correctly by Naïve Bayes classifier. Also th ere are not sufficient words in the sentiment dictionary to give a good result.

```
Probability  : 0.999999984647635 pPosW  : 1.7758025857290037e-230 pNegW  : 2.726276993120662e-238
Probability  : 0.6953730603023605 pPosW  : 1.567486113365673e-57 pNegW  : 6.866796040754117e-58
Probability  : 0.07679043784222539 pPosW  : 3.836038670824302e-153 pNegW  : 4.611860123767415e-152
ERROR (pos classed as neg 0.08):if this movie were a book , it would be a page-turner , you can't wait to see what happens ne
xt .
Probability  : 0.9725122414052408 pPosW  : 2.622916223196191e-152 pNegW  : 7.413591817961652e-154
Probability  : 0.9998993640049902 pPosW  : 5.002000088706027e-159 pNegW  : 5.034319193375935e-163
Probability  : 0.9999999999298197 pPosW  : 1.4246704558504537e-180 pNegW  : 9.998385540696694e-191
```

Hence the Probability obtained for the above sentence is 0.08 i.e., less than 0.5, so it is classed as negative.

- **Task 7:**

I tried the Demo for Sentiment Tree from the link given in description. I tested a review ” This movie doesn't care about cleverness, wit or any other kind of intelligent humor.” And found the following Sentiment tree:



After downloading the code for Stanford system, I set up the Java environment for my computer and then trained model from CMD. I passed few reviews which were classified wrong by Naïve Bayes Classifier. All the reviews were negative but classed as positive.

Following are the results for Stanford CoreNLP System:

great battery life , perfect size , but a tid bit quieter than i would like .
Positive
this is a very nice phone , but there is no warranty on it .
Neutral
the gprs connection is sometimes slow , and writing instant messages with the included aol instant messenger software is a pain , but the other t-zones applications are quite useful .
Negative
i spent hours setting up the stations (accepts about 13-14 , i believe) , though the reception is unpredictable .
Negative
i have excellent hearing but the volume level on this phone is especially quiet .
Positive
some of the higher pitched rings are very easy to hear , but not easy to listen to .
Negative
the keypad is a decent size , but the power on/off key is small and difficult to press .
Neutral
after several years of torture in the hands of at&t customer service i am delighted to drop them , and look forward to august 2004 when i will convert our other 3 family-phones from at&t to t-mobile !
Positive
the fact that the " 0 " key is the space key for text input is a bit confusing , as many phones use the " # " key instead .
Negative
so loud , really , that it does n't work terribly well as a silent ringer option .
Negative
i am bored with the silver look .
Negative

polarity	Number of data in result
Positive	**3**
Neutral	**2**
Negative	**6**

So, 6 sentences out of 11 are correct in Stanford model which were incorrectly classified by Naïve Bayes classifier.

Conclusion

Understanding Code was easy and evaluation metrics were useful in comparing and understanding the performance of techniques used for sentimental analysis.

For Naïve Bayes classifier I noticed that accuracy for Film train data and Film test data was overall good but accuracy for Nokia phone review was not good. The reason behind this difference is, the film train data was trained using trainBayes(). So when testing film data against testBayes() gave good results. Nokia phone review data set was not trained and was directly evaluated using testBayes. Due to this the accuracy for Nokia phone review was not good.

I have created a small program to calculate number of useful words in sentiment dictionary. This function print the words and gives the total count for positive and negative words. When

we check most useful words in case of bigrams and trigrams, we find some bigrams and trigrams in the list but when we check availability of those useful words in sentiment dictionary then no bigram or trigram from the list is present in the sentiment dictionary. If dictionary is updated for bigrams and trigrams, results should improve.

Most of the words from Nokia review are not in the sentiment dictionary, due to which sentence is misclassified resulting in low accuracy. To avoid this dictionary with wide range of positive and negative words should be used.

I changed threshold to 0 when encountered with error for Nokia phone dataset, divide by Zero error. After changing threshold code worked fine.

Ways to Improve Sentiment analysis:

1) Sentiment classification could be improved with the use of Negation tokens. Negations can be detected with the help of negation tokens.
2) Using Dictionary With large number of positive and negative words along with bigrams and trigrams.
3) If possible solving the problem using Support Vector Machine algorithm. It is non-probability based algorithm which separates the data using a hyperplane. According to a research paper discussed in Introduction, SVM perform better than Naïve Bayes and Random forest.
4) We can add rules for conjunctions like ‘but’, ‘moreover’, etc. which connects two words. In most of the cases use of ‘but’ changes the sentiment of the sentence.

Difficulties in Sentiment Classification:

1) I think Negation is the biggest challenge for sentimental analysis today.
2) Domain Dependence is another challenge.
3) Need for dictionaries with bigrams and trigrams.
4) Understanding sarcasm and Irony is another challenge in text analysis.