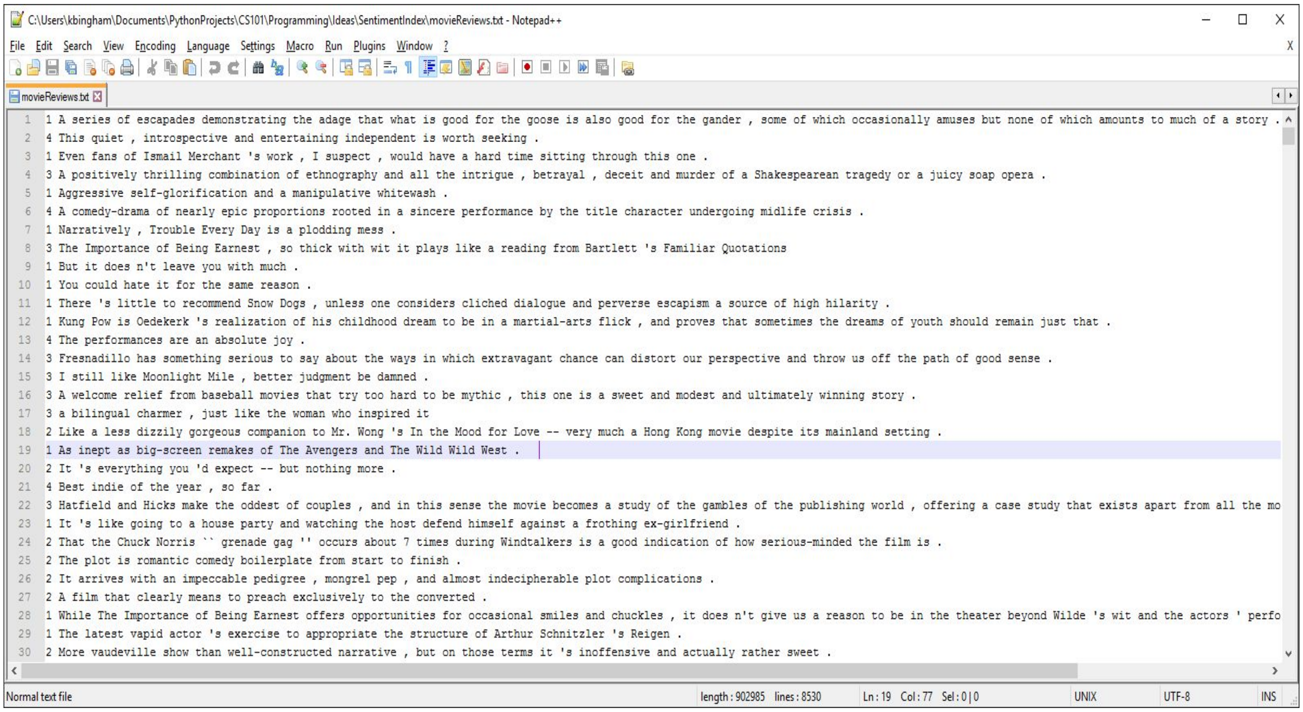
CS 101
Spring 2016
Program Assignment # 7 - Sentiment Analysis
Algorithm Due : Sunday, April 17, 2016
Program Due : Sunday, April 24, 2016

Sentiment Analysis

Sentiment Analysis is a Big Data problem which seeks to determine the general attitude of a writer given some text they have written. For instance, we would like to have a program that could look at the text “The film was a breath of fresh air” and realize that it was a positive statement while “It made me want to poke out my eye balls” is negative.

One algorithm that we can use for this is to assign a numeric value to any given word based on how positive or negative that word is and then score the statement based on the values of the words. But, how do we come up with our word scores in the first place?

That’s the problem that we’ll solve in this assignment. You are going to search through a file containing movie reviews from the Rotten Tomatoes website which have both a numeric score as well as text. You’ll use this to learn which words are positive and which are negative. The data file looks like this:


C:\Users\kbingham\Documents\PythonProjects\CS101\Programming\Ideas\SentimentIndex\movieReviews.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
movieReviews.txt
Normal text file
length : 902985 lines : 8530
Ln : 19 Col : 77 Sel : 0 | 0
UNIX
UTF-8
INS

```
1 A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story .
4 This quiet , introspective and entertaining independent is worth seeking .
1 Even fans of Ismail Merchant 's work , I suspect , would have a hard time sitting through this one .
3 A positively thrilling combination of ethnography and all the intrigue , betrayal , deceit and murder of a Shakespearean tragedy or a juicy soap opera .
1 Aggressive self-glorification and a manipulative whitewash .
4 A comedy-drama of nearly epic proportions rooted in a sincere performance by the title character undergoing midlife crisis .
1 Narratively , Trouble Every Day is a plodding mess .
3 The Importance of Being Earnest , so thick with wit it plays like a reading from Bartlett 's Familiar Quotations
1 But it does n't leave you with much .
1 You could hate it for the same reason .
1 There 's little to recommend Snow Dogs , unless one considers cliched dialogue and perverse escapism a source of high hilarity .
1 Kung Pow is Oedekerk 's realization of his childhood dream to be in a martial-arts flick , and proves that sometimes the dreams of youth should remain just that .
4 The performances are an absolute joy .
3 Fresnadillo has something serious to say about the ways in which extravagant chance can distort our perspective and throw us off the path of good sense .
3 I still like Moonlight Mile , better judgment be damned .
3 A welcome relief from baseball movies that try too hard to be mythic , this one is a sweet and modest and ultimately winning story .
3 a bilingual charmer , just like the woman who inspired it
2 Like a less dizzily gorgeous companion to Mr. Wong 's In the Mood for Love -- very much a Hong Kong movie despite its mainland setting .
1 As inept as big-screen remakes of The Avengers and The Wild Wild West .
2 It 's everything you 'd expect -- but nothing more .
4 Best indie of the year , so far .
3 Hatfield and Hicks make the oddest of couples , and in this sense the movie becomes a study of the gambles of the publishing world , offering a case study that exists apart from all the mo
1 It 's like going to a house party and watching the host defend himself against a frothing ex-girlfriend .
2 That the Chuck Norris `` grenade gag '' occurs about 7 times during Windtalkers is a good indication of how serious-minded the film is .
2 The plot is romantic comedy boilerplate from start to finish .
2 It arrives with an impeccable pedigree , mongrel pep , and almost indecipherable plot complications .
2 A film that clearly means to preach exclusively to the converted .
1 While The Importance of Being Earnest offers opportunities for occasional smiles and chuckles , it does n't give us a reason to be in the theater beyond Wilde 's wit and the actors ' perfo
1 The latest vapid actor 's exercise to appropriate the structure of Arthur Schnitzler 's Reigen .
2 More vaudeville show than well-constructed narrative , but on those terms it 's inoffensive and actually rather sweet .
```

Note that each review starts with a number 0 through 4 with the following meaning:

- 0 : negative

- 1 : somewhat negative
- 2 : neutral
- 3 : somewhat positive
- 4 : positive

You will ask the user for the name of a file that contains a list of words. Using the moviereviews.txt file you will compute the average value of the word using the movie reviews. This means you'll have to find all the places where the word is is used in a review and add that movies score to the value for the word. For each word we want to keep track of how many times it's used, the average value of the word as a sentiment, and the standard deviation of the word. The standard deviation is calculated by subtracting the average from the value and then squaring the result. You then sum those results and divide by how many items are in your sequence. For the values 4, 2, 1, 3, 2, 2, 1, 0 the average is 1.874. The standard deviation is calculated in the following manner

$$std = ((4-1.874)^2 + (2-1.874)^2 + (1-1.874)^2 + (3-1.874)^2 + (2-1.874)^2 + (2-1.874)^2 + (1-1.874)^2 + (0-1.874)^2)/8$$

Requirements

- Your programs should make good use of functional decomposition. Make sure you break your program down into functions.
- Your program lets the user choose to analyze another file or Quit. Incorrect choices are warned and prompt for input repeatedly until valid input is obtained.
- The program will ask for the name of the file with a list of words to find the sentiment of. If the name of the file is invalid the the user is warned and prompted until they give a good filename.
- The program will calculate the count, average and standard deviation for each word in the file given.
- The program will let the user choose how to sort the output before displaying it. Either by Average low to high, Average, high to low, Standard Deviation Low to High or Standard Deviation High to Low
- Display the words in the sort method given in a nice table layout.
- If moviewreviews.txt is not found, warn the user and exit the program.

Development notes

There is a statistics module that will compute the standard deviation and the average easily and quickly. You are NOT allowed to use this module. Instead create your own functions to calculate these values.

Example

```
>>> ================================ RESTART
================================
>>>
```

```
              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> e
You must choose one of the valid choices of 1, Q


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> 1
Enter the name of the file with words to score bad
Could not find the file you specified bad
Enter the name of the file with words to score wordlist1.txt
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> e
You must choose one of the valid choices of 1, 2, 3, 4
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> 1

Word             Occurrence   Avg Score         Std
===================================================
cliched                   8      0.7500      0.4375
thrilling                 5      1.4000      1.8400
mess                     93      1.6344      1.5868
hate                     44      1.7955      1.3445
epic                     40      2.4500      1.5975
```

```
joy                     136      2.6765      1.3806


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> 1
Enter the name of the file with words to score wordlist1.txt
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> 2

Word             Occurrence   Avg Score         Std
===================================================
joy                     136      2.6765      1.3806
epic                     40      2.4500      1.5975
hate                     44      1.7955      1.3445
mess                     93      1.6344      1.5868
thrilling                 5      1.4000      1.8400
cliched                   8      0.7500      0.4375


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> 1
Enter the name of the file with words to score wordlist1.txt
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> 3
```

```
Word             Occurrence   Avg Score         Std
===================================================
cliched                   8      0.7500      0.4375
hate                     44      1.7955      1.3445
joy                     136      2.6765      1.3806
mess                     93      1.6344      1.5868
epic                     40      2.4500      1.5975
thrilling                 5      1.4000      1.8400


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> 1
Enter the name of the file with words to score wordlist1.txt
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> 4

Word             Occurrence   Avg Score         Std
===================================================
thrilling                 5      1.4000      1.8400
epic                     40      2.4500      1.5975
mess                     93      1.6344      1.5868
joy                     136      2.6765      1.3806
hate                     44      1.7955      1.3445
cliched                   8      0.7500      0.4375


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> 1
Enter the name of the file with words to score wordlist2.txt
       Sort Options
    1. Sort by Avg Ascending
    2. Sort by Avg Descending
    3. Sort by Standard Deviation Ascending
    4. Sort by Standard Deviation Descending

===> 3

Word             Occurrence   Avg Score         Std
===================================================
incoherent                8      0.1250      0.1094
tears                     6      3.8333      0.1389
unfocused                 8      0.2500      0.1875
unpredictable             7      3.7143      0.2041
refreshing               24      3.3750      0.2344
muted                     5      1.6000      0.2400
wonderful                37      3.4324      0.4617
mechanical                6      0.6667      0.5556
preachy                  10      1.2000      0.5600
devoid                   14      0.5000      0.6786
witty                    22      2.9545      0.7707
indulgent                15      1.5333      0.7822
formulaic                15      1.0667      0.8622
horrible                 12      0.5833      0.9097
always                   50      2.7800      0.9716
slapstick                14      1.7857      1.0255
eccentric                12      2.3333      1.0556
dull                     66      0.9242      1.1003
dog                      47      1.9362      1.1236
spend                    22      2.0455      1.1343
moving                   61      3.1148      1.1836
barely                   25      1.2000      1.2000
complicated              12      2.6667      1.2222
resolutely                6      0.6667      1.2222
provoking                15      2.8000      1.2267
strong                   58      2.5172      1.2497
sentiment                54      2.0926      1.3433
family                   90      2.6778      1.3517
quirky                   38      2.7105      1.3636
terrible                 17      1.1176      1.3979
interest                147      1.7551      1.4094
cast                    144      2.4653      1.5266
hardly                   21      1.7143      1.5374
style                    79      2.1899      1.5969
```

```
value                    27      2.1852      1.6324
nor                     124      2.0403      1.6355
dialogue                 71      1.5915      1.7064
soulless                  8      0.6250      1.7344
car                     328      1.9421      1.7497
narrative                53      1.9623      1.8476
tale                    176      2.2386      1.8521
writing                  40      1.6750      1.9194
historical               26      2.1538      1.9763
words                    23      2.0870      2.1664


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> e
You must choose one of the valid choices of 1, Q


              Python Sentiment Analysis.

    1. Get sentiment for all words in a file
    Q. Quit

===> q
>>>
```

References

- Standard Deviation https://en.wikipedia.org/wiki/Standard_deviation