Fall 2022
Homework 8

Released: November 8, 2022
Due: November 18th, 2022 @ 9pm ET
Late Due Date (1-48 hours late): November 20th, 2022 @ 9pm ET

Homework 8: 

Introduction
Star Wars is the quintessential struggle between the forces of good and evil. In this homework we apply our data science skills to explore the story of a young lad who dreams of adventure in a galaxy far far away.

Expectations
We expect you to write your code in a *modular style* using a well-defined collection of functions. The functions you implement are totally at your discretion. As always, you will be graded on a) correctness, b) code design and efficiency, and c) documentation and style. You may not use Pandas, numpy, or any library such as NLTK that would trivialize some parts of the implementation. *Do your best. Or do not do your best. There is no try.*

What you're practicing in this assignment
- String processing and sentiment analysis
- dictionaries and other data structures
- Writing functions

Requirements

Read the data. Load file, starwars.txt, contains the entire script of the Star Wars original movie, but in a nicely formatted .CSV file with three columns:

- **line_number**: 1, 2, 3, ….
- **character**: The name of the character who is speaking, e.g., LUKE, VADER, etc.
- **dialog**: The character's spoken line.

For example:

.
.
641|LEIA|This is some rescue. When you came in here, didn't you have a plan for getting out?
642|HAN|He's the brains, sweetheart.
643|LUKE|Well, I didn't...
644|HAN|What the hell are you doing?
645|LEIA|Somebody has to save our skins. Into the garbage chute, wise guy.
.
.

Note that each column is delimited by a vertical bar '|' character rather than a comma because commas in the dialog column might otherwise cause you some difficulty when parsing the dataset.

Read the data into a list of dictionaries, as you did with the Blue Bikes data in HW7. Each dictionary represents one line from the movie where keys are column names (line_number, character, dialogue) and values are the values for that line of dialogue. Do not clean or modify the dialogue text as you store it in your data structure. Simply leave it as is.

Here is an example of the first two lines of dialogue (out of 1010) read in with this strategy:

```
[{'line_number': 1, 'character': 'C3PO', 'dialogue': "Did you hear that? They've shut down the main reactor.  We'll be destroyed for sure.  This is madness!"}, {'line_number': 2, 'character': 'C3PO', 'dialogue': "We're doomed!"}]
```

Separately read the lists of positive words (positive-words.txt) and negative words (negative-words.txt) into lists or, for better performance, into *sets*. (Either is fine!)

Analysis and Reporting Requirements

1. Compute a sentiment score for each line by adding the number of positive words and subtracting the number of negative words in that line. The file positive-words.txt contains a list of positive words. The file negative-words.txt contains a list of negative words. Update each dictionary in your list of dictionaries with the sentiment score (key = 'sentiment'). Be sure to CLEAN each spoken line of text by removing all punctuation and other non-alphanumeric characters in the following list of characters:

   ```
   PUNC = ["!", '"', "#", "$", "%", "&", "'", "(", ")",
           "*", "+", ",", "-", ".", "/", ":", ";",
           "<", "=", ">", "?", "@", "[", "]", "^", "_",
           "`", "{", "}", "|", "~"]
   ```

   Make sure to convert the dialogue text to lowercase letters before calculating the sentiment score. (Do not alter the dialog stored in your script data.)

2. (**we should see this when we run your program**) Report the most negative line and the most positive line in the movie along with its corresponding sentiment score and the character who speaks the line. In the case of ties, report any ONE of the lines with minimal or maximal sentiment scores. When we run your program, we should see output similar to:

   ```
   Most positive line:
   character: ???
   dialogue: ???
   score: ???

   Most negative line:
   character: ???
   dialogue: ???
   score: ???
   ```

3. REPORTING EXTRA CREDIT (up to +5 points, **we should see this when we run your program**): Generate a table that displays the minimum, average, and maximum sentiment score for lines spoken by the following characters: 'DARTH VADER', 'LEIA', 'C3PO', 'LUKE', 'OBIWAN', 'HAN SOLO'. Hint: Convert your script data to a dictionary mapping character to a list of sentiment scores for that character. Use formatted strings (link to documentation) to make your table look nice and readable. Include a table header. The rows should be listed in the order given above, from most-negative character to most-positive.

Visualization Requirements

1. (**luke_leia.pdf/.png**) Create overlapping histograms of the sentiment scores for LUKE (blue bars) and LEIA (red bars) using `bins=10` and `plt.xlim(-6,6)`. Setting `alpha=0.5` in your call to the histogram function will make your histogram bars partially transparent! Note that even though LUKE and LEIA are brother and sister, they express different sentiments. We might come up with a hypothesis like: LUKE, the dreamer and adventurer is generally more positive. LEIA, the princess and diplomat, is more reserved and neutral in her expressed sentiments. Look at your visualization. Does this hypothesis bear out? (No need to include your answer in your homework.)

   Hint: if you call plt.hist() twice, the results will overlap because they will be graphed on the same plot.

2. (**story_arc.pdf/.png**) Visualize the story arc of the Star Wars Movie using sentiment scores. Plot a moving average sentiment score using a window size of 20. A function for converting a list of numbers into a moving average is available on the class schedule in the latest version of `dataproc.py` or copy+pasted below in this assignment.

   The low point occurs during Tarkin's conference aboard the Death Star during which Darth Vader warns: *Don't be too proud of this technological terror you've constructed. The ability to destroy a planet is insignificant next to the power of the Force.* Label your figure at this point with the text "Tarkin's Conference". Here's an example of how to add a label to the graph. You'll want to look at your graph and estimate the best x and y coordinates.

   ```
   plt.text(x_coordinate, y_coordinate, "Tarkin's Conference")
   ```

   The high point occurs when the rebel fighters are getting ready to launch their attack against the Death Star. Label this peak: *Attack!!* Things become desperate during the attack but sentiment analysis shows how the movie ends on a very positive note (the destruction of the Death Star and the celebration that follows.)

   ```
   def avg(L):
       """ Compute the numerical average of a list of numbers.
       If list is empty, return 0.0 """

       if len(L) > 0:
           return sum(L) / len(L)
       else:
           return 0.0
   ```

```
def get_window(L, idx, window_size=1):
    """ Extract a window of values of specified size
    centered on the specified index
    L: List of values
    idx: Center index
    window_size: window size
    """

    minrange = max(idx - window_size // 2, 0)
    maxrange = idx + window_size // 2 + (window_size % 2)
    return L[minrange:maxrange]

def moving_average(L, window_size=1):
    """ Compute a moving average over the list L
    using the specified window size
    L: List of values
    window_size - The window size (default=1)
    return - A new list with smoothed values
    """
    mavg = []
    for i in range(len(L)):
        window = get_window(L, i, window_size)
        mavg.append(avg(window))

    return mavg
```

3. VISUALIZATION EXTRA CREDIT (up to +10 points, (**character_scores.pdf/.png**)): Count the number of positive and negative lines spoken by each character (lines with sentiment score of 0 should be ignored). Now represent each character as a point on a scatter plot:

```
(negative_lines, positive_lines)
```

Your scatter plot should have the following properties:

a. Both the x and y axis should range from 0 to 70.
b. Each point is plotted with a different color and there is a legend to identify each character.
c. Only plot characters where: `negative_lines + positive_lines > 10`.
d. Axes should be clearly labeled and you should have a descriptive title.
e. Overlay a grid.
f. To distinguish the characters that lean positive vs. those that lean negative, draw a diagonal line from (0,0) to (70,70) in any color. Characters above the line tend to be more positive. Characters below the line tend to be more negative.
g. Your scatter plot should be square and rendered at high resolution for clarity. Try:

```
plt.figure(figsize=(6,6), dpi=200)
```

Remember that if you use the `plt.figure()` command, it must be done *before* you plot any points (this is like choosing a canvas before you start painting).

What to Submit

Submit your program (`starwars.py`) and all visualizations (luke_leia.pdf, story_arc.pdf, and, optionally, character_scores.pdf (.png is also okay)). **In your program header, include the program text output produced when you run your program.**

Please Read: It is your responsibility to verify that your program runs (by examining the autograder output), and that ALL required files have been submitted to gradescope. If you forget to submit your code, your visualization, or your program doesn't run, your grade will be affected. If you *resubmit* your assignment for any reason, your resubmission must include ALL required files. Files from previous submissions are deleted from gradescope and not accessible to the graders! Resubmissions after the due date (but before the late due date) will receive the full late penalty.

No homework files will be accepted for grading or regrading after the late due date!