CSCI 141 Project 3

Project Due Friday, November 11th 2022 by 2200

For the third project, you will be creating a suite of functions to process and analyze Twitter data. You will use your functions to analyze a Twitter data set. The data you will use for this project is an English-language subset of the tweets which were labeled with the NewsFeed class.

For the first part of the project, you will write 2 functions and correct 1 function that you will use in a main program to process and analyze the data. The second part of the project is to write the main program.

You are provided with several text files, as well two Python skeleton files (Project_3.py, Project_3_Main.py. There are 3 items which you will submit: the Project_3.py file with your code for the functions, the Project_3_Main.py with the main program for Part 2, and a .pdf write-up.

About Twitter data and Tweets

Tweets are submission to the social media platform Twitter. They are 140 characters in length or less. Tweets often contain hashtags which are words or phrases with the prefix #, for example, #avocadotoast. Tweets may reference other Twitter users, as indicated by the @, for example, @realDonaldTrump. Users may re-post a tweet posted by another user – this is referred to as re-tweeting, and is signified by RT followed by the user who originally posted it, for example, RT @POTUS. The tweets you will be examining for the project will contain links which have the prefix https or http.

Part One: Processing and analyzing tweets

All of the code for Part 1 should be submitted in the Project_3.py file. You will fill in your functions under the definition lines.

Task 1: *distill_tweet(str_tweet, punct = '.,;!?":/')* correction

You have been given code for a function called *distill_tweet* which does not work properly. Correct the code you have been given so that it meets the following specifications. You are not permitted to add or delete or move any lines, and you must correct the lines in place.

The required argument **str_tweet** is a string that is a tweet. The optional argument **punct** is a string of punctuation symbols to remove. This function reformats the tweet (Pre-process the tweet) to make it ready for further processing as follows:

- This function will always remove web links from the tweet. These can be identified as having the prefix (this just means starting with) http or https – they will often be at the end of the tweet, but could be located anywhere, and there may be multiple links in one tweet.
- This function will remove stopwords from the tweet using the list of English stopwords called **ENG_WORDS** provided for you in Project_3.py. Do not change the code given to you.
- Standalone numbers should be removed from tweets. For example, if the tweet says something like '3 people arrested', the 3 would be removed.
- It will also deal with punctuation. Delete whatever punctuation is specified by the optional argument **punct**. Keep in mind that the symbols # and @ have special meanings in Twitter and you should not remove them. You should replace apostrophes (‘ and ') and dashes with a space instead of deleting them.
- All of the text should be put into lower case.
- **The function returns a list** that is the remaining words from the processed tweet. The function should print nothing.

Here are some examples, note this is not actual code, and words like “Example 1” should not print when your function runs.

Example 1: Taco trucks on every corner? Business group wants them at every polling site instead. https://t.co/7YKcn1vzNf #politics

Default value of **punct**; **distill_tweet returns:**

['taco', 'trucks', 'every', 'corner', 'business', 'group', 'wants', 'every', 'polling', 'site', 'instead', '#politics']

Example 2: Smithsonian Celebrates 'Star Trek's' 50th Anniversary https://t.co/HatHjsTHkG

Default value of **punct**; **distill_tweet returns:**

['smithsonian', 'celebrates', 'star', 'trek', '50th', 'anniversary']

Example 3: Patricia Barry, daytime-television and film actress, dies at 93 https://t.co/fD3gFivZkx

punct is ',;$'; **distill_tweet returns:**

['patricia', 'barry', 'daytime', 'television', 'film', 'actress', 'dies']

Task 2: ***top_entries(tweets, min_count = 1, hashes = False, mentions = False, punct = '.,;!?":/')***

Write a function called **top_entries** which **returns a dictionary**. The dictionary returned will contain strings as keys and integer counts as values.

The required argument **tweets** is a **list** of tweets, that is, a list of strings that are ‘raw’ tweets. **Within this function, you MUST use your distill_tweet function to pre-process the tweets so that web links, stopwords, standalone numbers, and appropriate punctuation are removed and all text is in lower case.** If you re-write the code from distill_tweet to do this inside of this function, you will lose points. The dictionary that is created and returned varies depending on the values of the optional arguments. This is best explained by example:

- When hashes is **True** (the value of mentions can be **True** or **False**), the function will only consider hashtags in the tweets, that is words that start with the # character. The returned dictionary will contain each hashtag as a key, and the associated value will be the number of times that hashtag occurred in the list of tweets. Here is an example with the default value of 1 for min_ count:

Example 1: ['Donald Trump inauguration: Protests begin to turn violent https://t.co/HSM2dzQTkW', 'Trump pitches $20 billion education plan at Ohio charter school that received poor marks from state https://t.co/z0QvxHqiNl #politics', 'FIRST Global Challenge visas granted to 99% of teams https://t.co/RdpueSdV60 https://t.co/fD99GAEL5R', 'Largest federal employee union endorses Clinton #politics', "New report examines Idaho's Medicaid mental health manager #health"]

Default value of **punct**; **Return value**: {'#health': 1, '#politics': 2}

The min_count parameter provides a lower bound, i.e., a hashtag must occur at least min_count times to be added to the dictionary. If we use the same input and change the value of min_count to 2, we would get the following output:

Return value: {'#politics': 2}

#health no longer appears in the output because it is less than min_count.

- When hashes is **False** and mentions is **True**, the function will only consider mentions in the tweets, that is, words that start with the @ character. The returned dictionary will contain each mention as a key, and the associated value will be the number of times that mention occurred in the list of tweets. Here is an example with the default value of 1 for min_count:

Example : ["@willhoerter We can't afford another democrat in the Office",
"@minasmith64 @willhoerter Law-abiding citizen and responsible gun owner",
'@Davesmyname @MalcusD @CNN Puppets, never waste your time watching CNN Politics',
'@Cameron_Gray @WarDamnGunners Omg, and they claim to be open-minded?']

Default value of **punct**; **Return value**: {'@willhoerter': 2, '@minasmith64': 1, '@davesmyname': 1, '@malcusd': 1, '@cnn': 1, '@cameron_gray': 1,
'@wardamngunners': 1}

Here is the same example with min_count = 2:

Return value: {'@willhoerter': 2}

- When hashes is **False** and mentions is **False**, the function will consider all words in the tweets that do not start with @ or #. Keep in mind that the tweets MUST be processed to remove web links and punctuation as well as to standardize case prior to dictionary creation. Here is an example using min_count equal to 1:

Example : ['7 Ways to Stay Cool and Safe During the Heat Wave https://t.co/vsazA5brGa https://t.co/kDLlarujBV',
'Bevin wants to downsize scope of KentuckyWired project #business #news',
'Dodgers outright pitcher Brandon Beachy to TripleA #baseball',
'Michigan couple starts pillow project for ill children #health',
'Trump files objection to Stein recount request in Michigan https://t.co/3xYiGHVyV0 https://t.co/WwOZ22DFBW']

Default value of **punct**; **Return value**: {'ways': 1, 'stay': 1, 'cool': 1, 'safe': 1, 'heat': 1, 'wave': 1, 'bevin': 1, 'wants': 1, 'downsize': 1, 'scope': 1, 'kentuckywired': 1, 'project': 2, 'dodgers': 1, 'outright': 1, 'pitcher': 1, 'brandon': 1, 'beachy': 1, 'triplea': 1, 'michigan': 2, 'couple': 1, 'starts': 1, 'pillow': 1, 'ill': 1, 'children': 1, 'trump': 1, 'files': 1, 'objection': 1, 'stein': 1, 'recount': 1, 'request': 1}

Here is the result if we change min_count to 2:

Return value: {'project': 2, 'michigan': 2}

Note that the order of entries in the dictionary is not important. In the example above, if your output had been: {'michigan': 2, 'project': 2}, that is still correct. Dictionaries are inherently unordered in Python.

Task 3: *tweets_from_file(filename)*

The two functions, distill_tweet() and top_entries() take a single tweet and a list of tweets respectively as inputs. Where do these tweets come from? They will need to be read in from a file.

Write a function called tweets_from_file(filename) **using list comprehension.** This function takes as input a **string that is a filename**. You have been given one sample file to refer to write this function. This file is called SAMPLE.txt. The format of the input file for tweets_from_file has one tweet per line. Take a look at the SAMPLE.txt file for an example.

Your tweets_from_file function should read in the tweets from the file and store them in a list. The function should **return the list of tweets and print nothing**.

The list returned from this function when called on the SAMPLE.txt file should look like this:

['Pay It 4ward: Woman helps veterans cope with PTSD https://t.co/L8YJsycFJ1 https://t.co/pezYS9oyXS', 'New Mexico lawmakers attempt to tackle state's car theft problem https://t.co/tNwl8ZtW0E https://t.co/iaz9cTY0bR', "Bills to reinstate New Mexico's solar tax credit move ahead https://t.co/8F5s6XreFp https://t.co/D20GWKPgLb",
"Senate confirms Trump's nominee for US ambassador to the UN https://t.co/vasbxjsBeZ https://t.co/rb3XP81ids"]

Part 2: Main Program

For this part of the assignment you will use your functions to process two Twitter data sets, stored in two separate files. These are the SEATTLE_POST data set (SEATTLE_POST.txt) and the RICHMONDVOICE data set (RICHMONDVOICE.txt).

1. Create and print a dictionary containing all of the hashtags (#) with their number of appearances from the SEATTLE_POST data set using default value of **punct**. Be sure to assign a variable name to this dictionary.
2. Create and print a dictionary containing all of the hashtags (#) with their number of appearances from the RICHMONDVOICE data set using default value of **punct**. Be sure to assign a variable name to this dictionary.
3. Create and print a dictionary containing all of the words (no hashtags or mentions) that appear at least 20 times in tweets with their number of appearances from the SEATTLE_POST data set using **punct** that is '.:;$'. Be sure to assign a variable name to the dictionary.
4. Create and print a dictionary containing all of the words (no hashtags or mentions) that appear at least 20 times in tweets with their number of appearances from the RICHMONDVOICE data set using **punct** that is '.:;$'. Be sure to assign a variable name to the dictionary.
5. Find and print all of the hashtags that are in the SEATTLE_POST data set (obtained from 1) but not the RICHMONDVOICE data set (obtained from 2) and store them in an object – print out the object you've stored in them so we can see them. You don't need to include their number of appearances. There are many ways to do this, but don't overthink it. (Hint: what data structure did we learn about, but haven't actually used in this project yet?)
6. Find and print all of the words (no hashtags or mentions) that are shared by the two data sets (obtained from 3 and 4) - i.e. the words (no hashtags or mentions) that can be found in both, and store them in an object – print out the object you've stored in them so we can see them. You don't need to include their number of appearances. There are many ways to do this, but don't overthink it. (Hint: what data structure did we learn about, but haven't actually used in this project yet?)

You MUST use your functions. You should not repeat code from your functions inside of the main program. There should be no def lines in your main program file.

SUBMISSION EXPECTATIONS

Project_3.py Your implementations/correction of the functions in Part 1.

Project_3_Main.py Your main program for Part 2.

Project_3.pdf A PDF document containing your reflections on the project. **You must also cite any sources you use. Please be aware that you can consult sources, but all code written must be your own. Programs copied in part or wholesale from the web or other sources will result in reporting of an Honor Code violation.**

POINT VALUES AND GRADING RUBRIC

- distill_tweet correction (20 pts)
- top_entries (22 pts)
- tweets_from_file (20 pts)
- Main program (21 pts)

- Writeup (1.8 pts)
- Autograder (15.2 pts)

N.B.: Your code should be concise and efficient. For example, you should not write ANY extraneous structures, should not specify default arguments, should get rid of unnecessary extra variables, and should comment out debugging lines. You must use the structures we have learned in this class to complete this assignment. For grading purposes, you can expect significant grade penalties if you import anything or write any function definition that is not needed in this project.

RUBRIC FOR MANUAL GRADING

Grade Level	Appropriateness	Style	Code Efficiency
A	Code uses current structures and modules discussed in class; meaningful variable names; commented as appropriate; any references used cited in write-up	Follows Python style guidelines we have discussed to include appropriate naming of variables and spacing; Proper use and naming of constants	Only contains structures necessary to accomplish the task at hand; Uses the most concise implementation possible
B	Code uses structures and modules discussed in class; variable names meaningful; references cited in write-up	Largely follows Python style guidelines but may be a few minor deviations	Only contains structures necessary to accomplish the task at hand; Implementation is generally concise
C	Code uses structures and modules discussed in class with minor additions from outside sources (e.g. string formatting) appropriately cited; most variable names meaningful	Several deviations from Python style guidelines; constants not used properly	Contains a small number of unnecessary structures such as casts that are not needed; Implementation somewhat concise
D	Code uses some structures and modules taken from outside sources with citations; variable names and code structure hard to decipher	Only loosely follows Python style guidelines	Contains a large number of unnecessary structures such as unused variables and casts; Implementation somewhat concise but could be improved, code may be hard to follow
F	Code uses structures and modules largely from outside sources only; variable names hard to decipher; code written in other programming languages or Python 2.7	Code appears to have been written with no consideration for style	Contains many unnecessary structures; Code is difficult to follow and overly verbose

Data Source: The raw data is available at https://github.com/fivethirtyeight - the data file you have was created from the .csv files, so do not use the original data. The original data was classified according to the type of tweet.