CPSC 1010

Programming Assignment #2

Simple Twitter Stream Analysis

In this project, you will make use of several newly learned features of the C language to do some simple analyses of two sets of tweets, one from Donald Trump and one from Hillary Clinton, that I collected around the end of the first week of October.

I collected the tweets at the web site **snapbird.org** from the timelines for @realDonaldTrump and @HillaryClinton by grabbing the most recent 500 tweets in each timeline and cut-and-pasting into one text file for each candidate (donald.txt and hillary.txt). I then manually “cleaned” the files, removing retweets, special characters, date lines, and the leading twitter handles.

For example, a set of 3 tweets like this:

realDonaldTrump MY PRO-GROWTH Econ Plan: ✅Eliminate excessive regulations! ✅Lean government! ✅Lower taxes! #Debates … twitter.com/i/web/status/7853067820… 10 Oct from Twitter for iPhone

Donald J. Trump

realDonaldTrump Hypocrite: @HillaryClinton is the single biggest beneficiary of Citizens United in history, by far. #debate #bigleaguetruth 10 Oct from Twitter Web Client

Donald Trump Jr.

DonaldJTrumpJr Ironic since Hillary has gotten a lot more of that "dark unaccountable money" into her campaign. #debates 10 Oct from Twitter for iPhone retweeted by realDonaldTrump

Would look like this after “cleaning”:

MY PRO-GROWTH Econ Plan: Eliminate excessive regulations! Lean government! Lower taxes! #Debates … twitter.com/i/web/status/7853067820…

Hypocrite: @HillaryClinton is the single biggest beneficiary of Citizens United in history, by far. #debate #bigleaguetruth

I then used another online tool to perform a form of *sentiment analysis* known as *polarity analysis*, to evaluate those tweet files. According to Wikipedia,

**Sentiment analysis** (also known as **opinion mining**) refers to the use of [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [text analysis](https://en.wikipedia.org/wiki/Text_analytics) and [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics) to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from [marketing](https://en.wikipedia.org/wiki/Marketing) to [customer service](https://en.wikipedia.org/wiki/Customer_relationship_management).

Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation (see [appraisal theory](https://en.wikipedia.org/wiki/Appraisal_theory)), affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader).

Polarity analysis seeks to determine whether the writer's attitude is positive, negative, or neutral. The website **help.sentiment140.com** provides a service by which you can invoke their polarity analysis tool on a file of tweets and it will produce a new file in which the tweets have been annotated with a leading integer value (0 or 2 or 4) in which 0 indicates negative attitude, 2 indicates a neutral attitude and 4 indicates a positive attitude, as demonstrated in the tweets. For example, the command seen below sent my local donald.txt file to the service at the listed URL and wrote the output to a file named donald\_sentiments.txt.

curl --data-binary @donald.txt "http://www.sentiment140.com/api/bulkClassify" > donald\_sentiments.txt

The files **donald\_sentiments.txt** and **hillary\_sentiments.txt,** whichcontain the results of running this analysis on the **donald.txt** and **hillary.txt** files, respectively.

The first few lines of the hillary.txt file are:

Trump wants to bring NYC's old, unconstitutional stop-and-frisk policy—aka racial profiling—to a city near you. hrc.io/2cRNHDw

When Ruline was born, women couldn't vote. Yesterday, at 103, she voted for Hillary. Make sure you're registered:… twitter.com/i/web/status/7819987382…

@BillClinton on pre-debate jitters (his) and Hillary's favorite TV shows, now playing on With Her:… twitter.com/i/web/status/7819802138…

and the associated sentiment file (hillary\_sentiments.txt) contains:

"2","Trump wants to bring NYC's old, unconstitutional stop-and-frisk policy—aka racial profiling—to a city near you. hrc.io/2cRNHDw"

"4","When Ruline was born, women couldn't vote. Yesterday, at 103, she voted for Hillary. Make sure you're registered:… twitter.com/i/web/status/7819987382…"

"4","@BillClinton on pre-debate jitters (his) and Hillary's favorite TV shows, now playing on With Her:… twitter.com/i/web/status/7819802138…"

to indicate that the first tweet is considered neutral and the second and third tweets are considered positive. As you can see, the tool is not perfect. See the site if you are interested to tweak the query (you can pass additional parameters).

I wrote a program (consisting of several files) to process these files and to produce reports with statistics about the collections of tweets. I also created a **Makefile** to make the job of compiling and running the programs to create the reports easier. The reports files are known as **donald\_report.txt** and **hillary\_report.txt**. Your job in this assignment is to recreate part of the program that produces the reports.

What I provide:

Text files:

* donald.txt and hillary.txt – the tweets
* donald\_sentiments.txt and hillary\_sentiments.txt – the sentiment analysis
* donald\_report.txt and hillary\_report.txt – the resulting reports

Code files:

* tweetStats.c – a code skeleton wit comments to guide you
* Makefile – with the needed rules to build the executable and the Donald-related reports. Note:
  + “make clean” removes old reports
  + “make” builds the tweetStats executable
  + “make donald” runs tweetStats with the command line parameters and output redirect needed to generate donald\_report.txt

What you need to produce:

* tweetStats.c – with code filled in to accomplish the desired functionality
* tweetFunctions.c – a code file containing the functions called from tweetStats.c
* tweetFunctions.h – a header file containing function prototypes for the functions in tweetFunctions.c
* an updated Makefile that also builds the Hillary-related files as well as the Donald-related files (so that “make hillary” creates the report hillary\_report.txt)

**Details:**

tweetStats.c takes in command line arguments. To run the program, the user should enter three parameters, as seen below. In this example, “Donald\_Trump” is the name that will appear in the report, “donald.txt” is the name of the file containing the cleaned tweets, and “donald\_sentiments.txt” is the name of the file containing the results of the polarity analysis.

tweetStats Donald\_Trump donald.txt donald\_sentiments.txt

I provided code to check that the correct number of parameters has been entered and to produce an error message if this is not the case. Also provided is code to open the files and to produce error messages if the file open is not successful.

The tweetStats.c file is heavily commented to explain what you need to do. Use the provided report files as guidelines for how to format the output. (Yours should match.)

**Due Dates:**

Part I – In part 1, implement everything except the character\_frequency and sentiment functions. Also, create “stubs” for the character\_frequency function and the sentiment function. A “stub” receives the correct number and type of parameters, but the function body just returns without doing anything. See web page for due dates.

Part 2 – Replace the character\_frequency and sentiment functions with their actual implementations. See web page for due dates.

**Implementation Notes and Rubric** – check back to the assignment web site periodically to find a grading rubric and some implementation notes.