

Jason Holdener

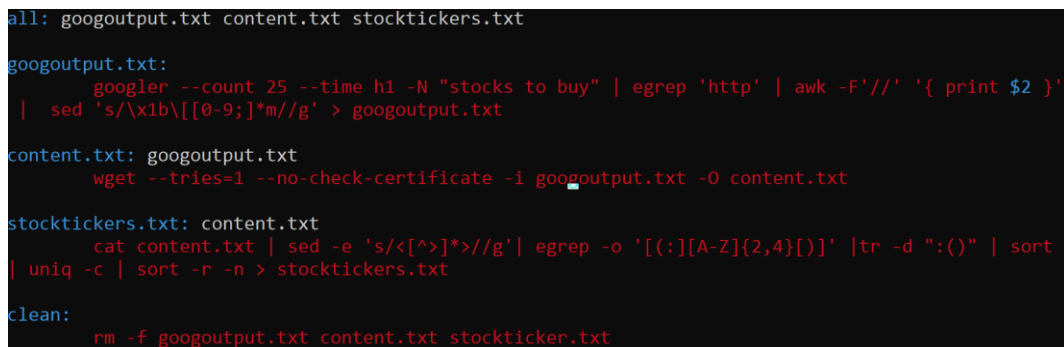
4/26/19

UNIX Tools Semester Project

“Stock-Finder”

For some time, I have thought about what steps one could in take in using computers to produce better stock market trading decisions. As a former penny-stock investor (I no longer like taking such large risks) knowing what stocks to buy within the hour is a crucial step in buying during the right time(s) before large spikes in prices occur or on betting whether stocks will go down. Given this brief description above and the skills/UNIX tools I’ve learned in the course I built a makefile that can search google news for stock articles released in a specific timeframe and can then output a sorted list of the stock tickers (ex. GOOG, FB, etc.) into a “stocktickers.txt” file with the most discussed or mentioned stocks appearing at the top of the list.

The total makefile can be seen via a screenshot below and the actual code can be found at my github account using the following link - <https://github.com/jholdener/Stock-Finder>



```
all: googoutput.txt content.txt stocktickers.txt

googoutput.txt:
    googler --count 25 --time h1 -N "stocks to buy" | egrep 'http' | awk -F'/' '{ print $2 }'
    | sed 's/\x1b\[0-9;]*m//g' > googoutput.txt

content.txt: googoutput.txt
    wget --tries=1 --no-check-certificate -i googoutput.txt -O content.txt

stocktickers.txt: content.txt
    cat content.txt | sed -e 's/<[^>]*>//g' | egrep -o '[:][A-Z]{2,4}[()]' | tr -d "()" | sort
    | uniq -c | sort -r -n > stocktickers.txt

clean:
    rm -f googoutput.txt content.txt stockticker.txt
```

First off, I discovered a nice tool called “googler” that allows one to run google from the command line. I should also mention that I can only confirm that this “googler” tool works with ubuntu Linux. “--count” allows for you to select the number of articles you would like to pull along with “--time” indicating how long ago the articles were published. So “--count 20 --time h6” would grab 20 articles from google published within the last 6 hours based upon your search term. By default, I found it best to set the count to 25 and I leave the time at “h1” because I am most curious as to what people are talking about within the hour. Finally, “-N” specifies that we search google news (this works better for finding financial articles) and your search term is by default “stocks to buy”. However, changing the search term to “pot stocks to buy” or “buy these stocks now” or “good short-term stocks” all give you different answers as you would expect. The rest of the first line of the makefile, “egrep 'http' | awk -F'/' '{ print \$2 }' | sed 's/\x1b\[0-9;]*m//g' > output.txt” simply grabs the http links, cleans them so that they can be used with “wget” and then outputs the collection of links into a text file called “googoutput.txt”.

The second line of code in the makefile uses wget to set the googoutput.txt file (the http addresses of our google search) as input using “-i”. I set “--tries = 1” meaning that it will only try to wget the contents of each article once as when you search many articles, say 100, it can take quite some time to try more than once. “--no-check-certificate” is used to get around Google’s attempts to stop

querying the search engine using tools like wget as it allows you to skip the certificate verification. The output of this line of code, which is the content of the articles themselves, is then stored in the “content.txt” file. You will also see that some of the articles could not be scrapped with wget which is why a sample size of 25 is best in my opinion. From playing with the tools and reading some blogs about wget it seems that using the “- - no-check-certificate” option causes this to occur with a small number of http addresses.

Great, so now we have a file called “content.txt” that contains all of the content from our financial articles. If we take a look at this file, we can see we need to clean the contents to just report the stock tickers. In financial articles, stock tickers are always referred to with parenthesis showing the stock exchange they are located within (usually NYSE or NASDAQ in the United States) followed by a semicolon and finally the stock ticker itself (Ex. “(NYSE:GOOG)”). Therefore, I used “sed -e 's/<[^>]*>//g’” to grab the texts of the articles followed by “egrep -o '[:][A-Z]{2,4}[)]’” to scrape the stock tickers and “tr -d “:()”” to delete the semicolons and parentheses and finally “sort | uniq -c | sort -r -n > stocktickers.txt” to sort the stock tickers in numerical order and then store them in the stocktickers.txt file.

When I ran the makefile at on 4/26/19 at 2:02PM (central time) we can see the “PEB” seemed to be mentioned most in the past hour with the 25 articles I scraped. We can also see that so far on 4/26/2019 PEB has gone up 4.12% which is quite a significant gain for one day of trading. (see the screenshots below)

```
jholdener@LAPTOP-FHQ74QV1:/mnt/c/Users/Jason's PC/Desktop/Unix Tools/Project/Code$ cat stocktickers.txt
s.txt | more
32 PEB
19 LTHM
18 BCRX
15 HASI
15 ECHO
12 XRP
12 XLM
12 GSK
12 ETH
12 BOOM
11 GRUB
10 WWE
10 LPI
10 HUBS
10 EHC
8 PPR
8 LKQ
8 JPM
7 VLO
7 HBI
7 FTR
6 XMR
6 XEM
```



8 minutes later I also check the 2nd (LTHM) and 3rd (BCRX) most mentioned stocks. As we can see below LTHM has shown good growth today while BCRX was most likely discussed due to its dramatic 4.26% loss. BCRX could be a good buying opportunity once it has stopped decreasing in price yet knowing when that will happen exactly is anyone's guess.

