MIDS W205: Exercise 2

Student: Aaron Yuen

Date: December 12, 2017

Description

This exercise uses Streamparse and Tweepy to develop an end-to-end Twitter streaming application via the Twitter API. This end-to-end application consists of:

1. Listening to the Twitter API for English tweets
2. Parsing the tweets and breaking down the tweets to words
3. Counting the words
4. Storing and updating the occurrences in a Postgres database
5. Two python scripts to query the database to retrieve the occurences

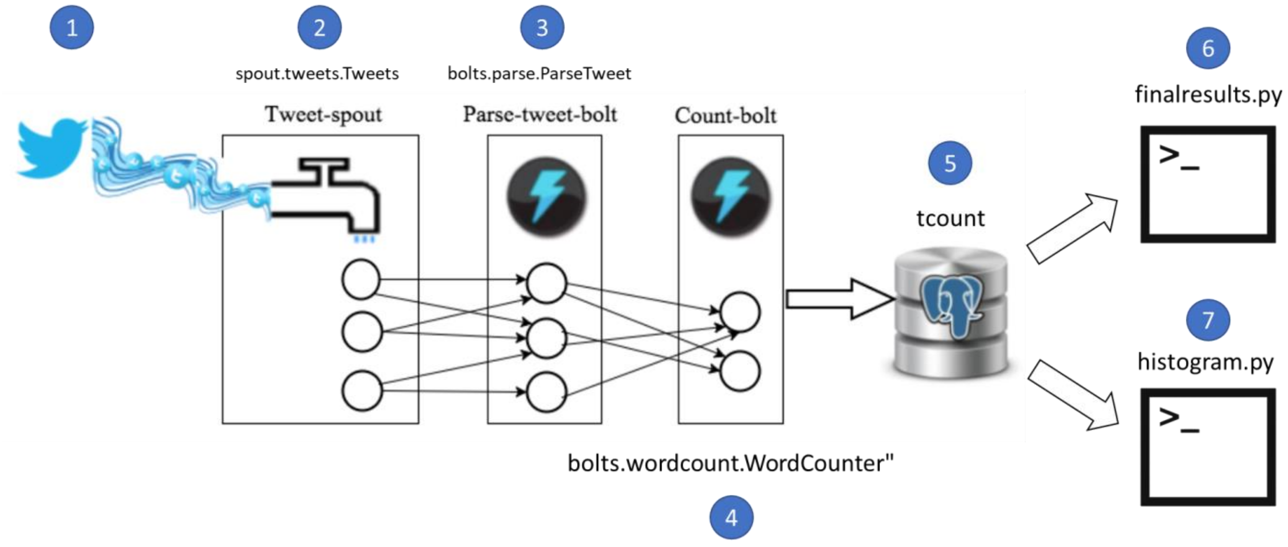
Directory and File Structure

The following is the directory and file structure under the /exercise_2/extweetwordcount/ folder

Filename	Location	Description
tweetwordcount.clj	topologies/tweetwordcount.clj	The clojure file detailing the topology of the stream application.
setup.sql	db_setup/setup.sql	A SQL file that sets up the tcount database and tweetwordcount table. Running this SQL file drops the previous tcount database and tweetwordcount table.
tweets.py	src/spouts/tweets.py	Tweet-spout
parse.py	src/bolts/parse.py	Parse-tweet-bolt
wordcount.py	src/bolts/wordcount.py	Count-bolt
finalresults.py	scripts/finalresults.py	Python script. When passed a single word as an argument, finalresults.py returns the total number of word occurrences in the stream. **Dependency:** tcount_db.py
histogram.py	scripts/histogram.py	Python script. The script gets two integers k1,k2 and returns all the words with a total number of occurrences greater than or equal to k1, and less than or equal to k2. **Dependency:** tcount_db.py
tcount_db.py	scripts/tcount_db.py	Custom python module that contains the details of the database and returns the database connection. Used from finalresults.py and histogram.py.

screenshot-twitterStream.png	Screenshots/screenshot-twitterStream.png	Screenshot of the Twitter stream running
screenshot-finalResults.png	Screenshots/screenshot-finalResults.png	Screenshot of finalresults.py returning data.
screenshot-histogram.png	Screenshots/screenshot-histogram.png	Screenshot of histogram.py returning data.
screenshot-database.png	Screenshots/screenshot-database.png	Screenshot of the database with the data.
Readme.txt	Readme.txt	Shows the step-by-step instructions on how to run the application.
Plot.png	Plot.png	A bar chart that shows the top 20 words in your Twitter stream.
Architecture.pdf	Architecture.pdf	This architecture PDF document detailing my Twitter application, including directory and file structure, application idea, description of the architecture, file dependencies, any necessary information to run the application, etc.

Architecture Details

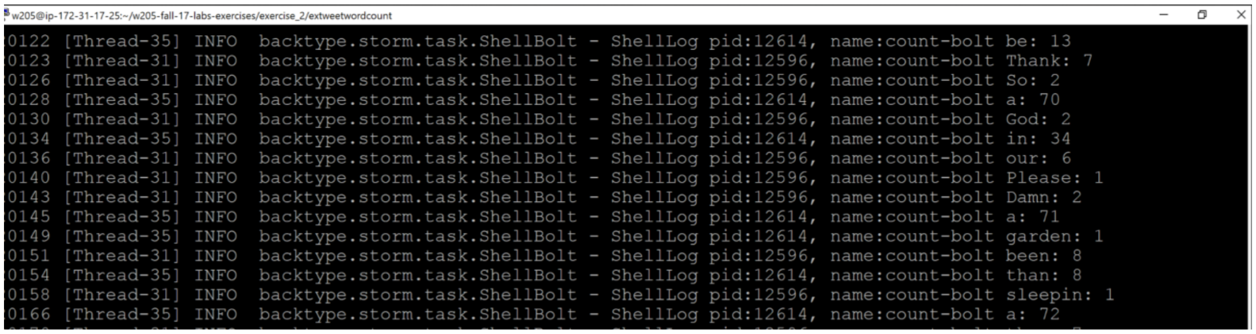


#	Component	Details
1	Twitter API	The Twitter API.
2	Tweet-spout	The tweet-spout accesses the Twitter API using the Tweepy library, creates the stream and listen for English tweets. It passes the tweets to the parse-tweet-bolt component.
3	Parse-tweet-bolt	The parse-tweet-bolt parses the tweets, extracts the words from each parsed tweet, and emits the words to the next bolt component (count-bolt) in the topology.

4	Count-bolt	The count-bolt counts the number of each word in the received tuples, and updates the counts associated with each word in the tweetwordcount table inside the tcount Postgres database.
5	tcount database	A pre-setup Postgres database that has one table: tweetwordcount. This table stores the (word, count) pairs.
6	Finalresults script	A python script that queries the tcount database to get the total number of word occurrences in the stream.
7	Histogram script	A python script that queries the tcount database to number of occurrences greater than or equal to k1, and less than or equal to k2, with k1 and k2 provided as an argument.

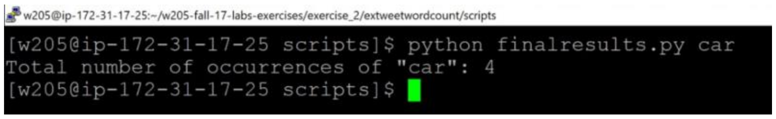
Screenshots of Application

Twitter stream



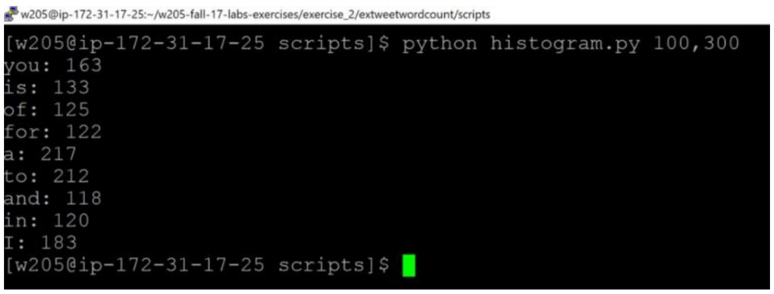
```
w205@ip-172-31-17-25:~/w205-fall-17-labs-exercises/exercise_2/extweetwordcount
0122 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt be: 13
0123 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt Thank: 7
0126 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt So: 2
0128 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt a: 70
0130 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt God: 2
0134 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt in: 34
0136 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt our: 6
0140 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt Please: 1
0143 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt Damn: 2
0145 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt a: 71
0149 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt garden: 1
0151 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt been: 8
0154 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt than: 8
0158 [Thread-31] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12596, name:count-bolt sleepin: 1
0166 [Thread-35] INFO  backtype.storm.task.ShellBolt - ShellLog pid:12614, name:count-bolt a: 72
```

Finalresults.py



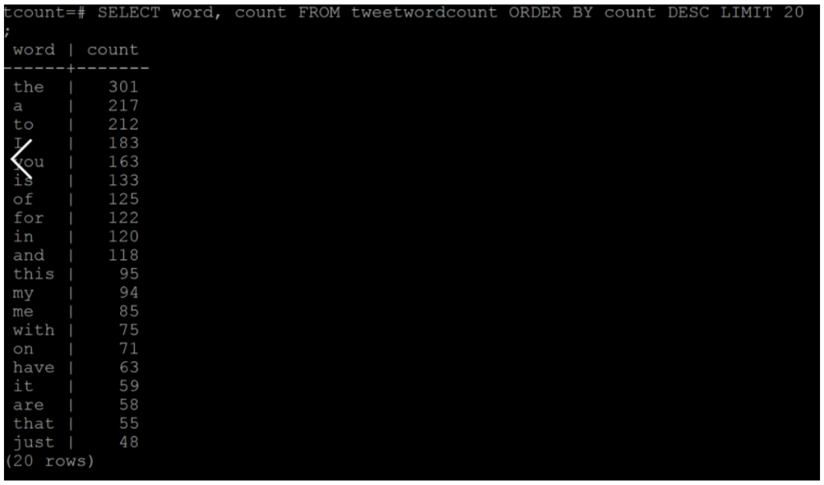
```
w205@ip-172-31-17-25:~/w205-fall-17-labs-exercises/exercise_2/extweetwordcount/scripts
[w205@ip-172-31-17-25 scripts]$ python finalresults.py car
Total number of occurrences of "car": 4
[w205@ip-172-31-17-25 scripts]$
```

Histogram.py



```
w205@ip-172-31-17-25:~/w205-fall-17-labs-exercises/exercise_2/extweetwordcount/scripts
[w205@ip-172-31-17-25 scripts]$ python histogram.py 100,300
you: 163
is: 133
of: 125
for: 122
a: 217
to: 212
and: 118
in: 120
I: 183
[w205@ip-172-31-17-25 scripts]$
```

Postgres database



```
tcount=# SELECT word, count FROM tweetwordcount ORDER BY count DESC LIMIT 20
;
 word | count
------+-------
 the  |   301
 a    |   217
 to   |   212
 I    |   183
 you  |   163
 is   |   133
 of   |   125
 for  |   122
 in   |   120
 and  |   118
 this |    95
 my   |    94
 me   |    85
 with |    75
 on   |    71
 have |    63
 it   |    59
 are  |    58
 that |    55
 just |    48
(20 rows)
```