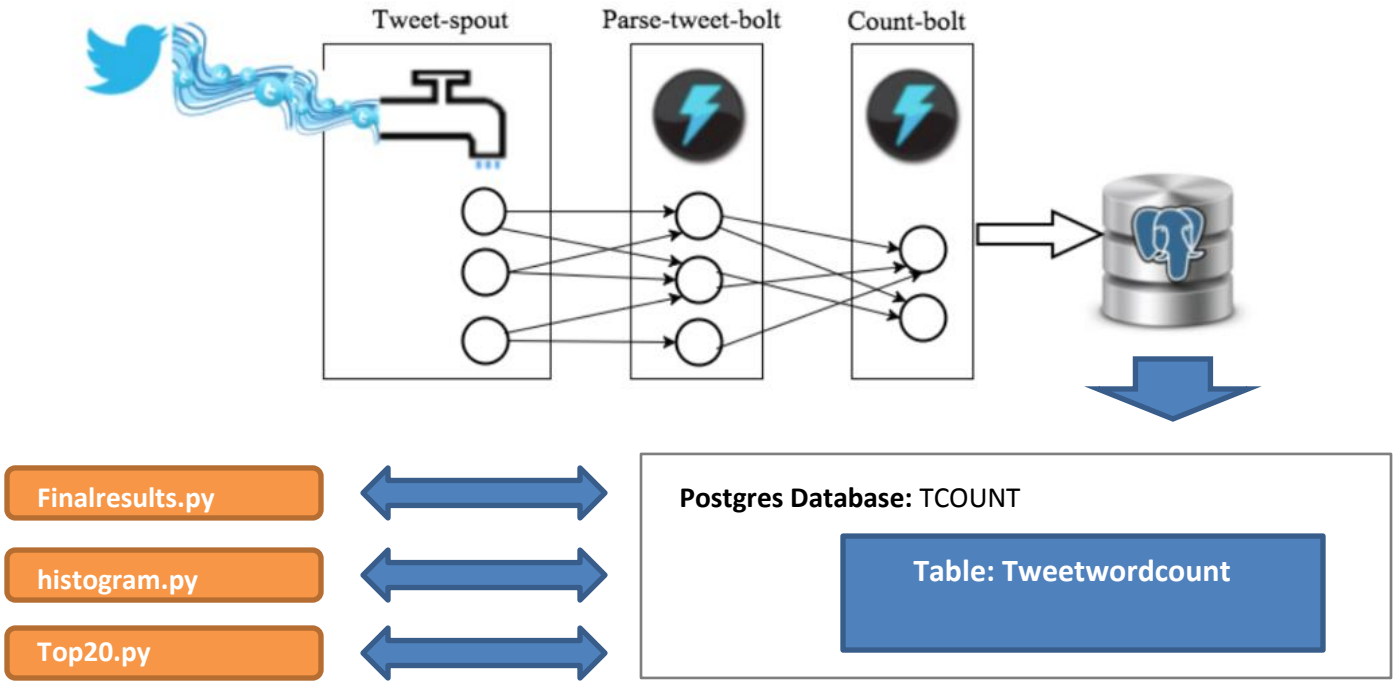
Live Twitter Data Processer: Fusing Streamparse, Word Parsing, and Relational Database Management

Introduction

The figure below gives a high-level overview of how this applications functions. We utilize a stream of tweets accessible through a custom Twitter application to feed our Storm streamparse with live data. Our network of spouts and bolts dissects incoming tweets into individual words. As the system counts the encountered words, it leverages the power of PsycoPG to populate a database in Postgres that can we queried later for analysis with python scripts.



Three default python queries can be run to: i) determine the final results for all counts or a specific string, ii) generate a list of all words with counts within a specified range, iii) find the top 20 words by count.

Overview of Directories:

Our application can be accessed by cloning our github repository **w205_Exercise_2** .

Our repository contains the following in the main directory:

- **All three default python scripts**
- **Instructions in Readme.txt**
- **The extweenwordcount directory.**
  - **This contains our storm topography, spouts, bolts, and all supplemental steamparse files.**

Instructions for running the twitter stream parse application:

1) Inside the extweetwordcount directory, run the storm streamparse:

$ sparse run

2) Enter Control + C to end the stream.

3) Run the finalresult.py script.

To return the count for a specific word, enter it as an argument after the script.

ex: $ python finalresults.py Berkeley

To return a list of all counts (limited to 1,000 results), you can omit this argument.

ex: $ python finalresults.py

3) Run the histogram.py script. To find all words with a count between two specific values, enter the lower and upper bounds of the ranges as a second and third argument after the script, respectively. Use spaces to separate the limits.

ex: $ python histogram.py 10 12 (this returns all words with a count between 10 and 12 and specifies their count)