W205 Exercise–2   Ram Balasubramanian 11/19/2017

Contents

Application Description

This twitter application does the following

- It connects to twitter and monitors the stream of tweets until the application interrupted by the user
- It parses each tweet – and splits the tweet into words (stripping out any urls, retweet tags etc.)
- It updates a database table in postgres with the word and the number of times the word has occurred so far in the tweet stream

Directory and File Structure

Directory Path	File Name	Description
/exercise_2	Plot.png	Bar chart file
	README.txt	File explaining how to run the app
	Twittercredentials.py	Twitter Application tokens/creds
	Twittercredentials.pyc	Have no idea what this file is or does
	barplot.py	Python program to create Plot.png file
	extweetwordcount	Directory (see below for files here)
	finalresults.py	Python program to create finalresults
	finalresults.txt	Results of the finalresults.py program
	hello-stream-twitter.py	Test program for twitter creds
	histogram.py	Python program for word count
	psycopg-sample.py	Sample python program to connect to postgres
	screenshots	Directory (see below)
/exercise2/extweetwordcount	README.md	
	_build	System files
	_resources	System files
	config.json	Not sure what ths file is
	fabfile.py	Not sure
	Logs	Logs from streamparse
	project.clj	Not sure
	src	See below
	tasks.py	Not sure
	topologies	See below
	virtualenvs	System file
/…extweetwordcount/src	bolts	Bolts directory
	spouts	Spouts directory
/../src/bolts	__init__.py	Initializer
	parse.py	Parse bolt file
	wordcount.py	Wordcount bolt file
/../src/spouts	__init__.py	Initializer
	tweets.py	Tweet spout program
../extweetword…/topologies	tweetwordcount.clj	Topology clojure file
/exercise_2/screenshots	TweetStreamRunning.png	Twitter stream
	topology_compnents.png	File showing topology components
	word count table.png	File showing wordcount

Application Idea

The Twitter Word Count application listen to twitter via the twitter API. It parses each tweet and breaks it out by word. It stores each word in a postgres table in the tcount database and updates the count of that word.

Other Information about the submission:

The screenshots directory contains three files as stated above in the Directory/Files Structure section. Here is what each screenshot shows

1. TwitterStreamRunning.png: Shows screenshot of when the application is running. This shows the messages emitted by the wordcount bolt – there are two processes (one for each bolt) that emits the word and the count of the word (number of times the word has been seen in this run)
2. topology_compnents.png: Shows the screenshot of the files in the logs directory. This shows 2 count-bolt files, 3 parse-tweet-bolt files and 3 tweet-spout files. This corresponds to the topology that was required for the application.
3. word count table.png: This file shows the execution of the "finalresults.py" program. The first execution looks for the number of occurrences of the word "trump". The second shows what the program returns when the given word is not present. Screenshot does not show what happens when no argument is provided to the file – it simply lists the word counts in alphabetical order as required (this is stored in the finalresults_output.txt file)