Exercise 2 Architecture

Spring 2017 W205 Section 1

Victoria Baker

A directory and file descriptions for the Twitter Wordcount application is outlined below. The files that were included with the quickstart command but were not edited for this project are highlighted in gray, and no description is included:

MIDS-W205_A2

exercise_2

- extweetwordcount
  - _build
    - classes
      - META-INF
        - maven
          - extweetwordcount
            - extweetwordcount
              - pom.properties
  - _resources/resources
    - bolts
      - __init__.py
      - parse.py
      - wordcount.py
    - spouts
      - __init__.py
      - tweets.py
  - logs
    - *this is folder contains all logs created during the development of this project
  - src
    - bolts
      - __init__.py
      - parse.py
        - The parse.py file extracts the tweet from the spout (tweets.py), splits it into words, and filters out hash tags, retweets, @ tags, and urls. It also strips words of leading and lagging punctuations and symbols, and converts words to all lowercase. If the word contains only ascii characters, then it is emitted.
      - wordcount.py
        - The wordcount.py file takes the valid words from the parse bolt (parse.py), connects to the postgres database tcount, checks if the word exists in the tweetwordcount table, and either

updates the count or inserts the word into the database. It increments the local count and prints the word and the local count to the screen.

        - spouts
            - __init__.py
            - tweets.py
                - The tweets.py file is the spout for this topology. It contains the authentication tokens to connect to Twitter. The file constructs a TweetStreamListener to listen and act on incoming tweets. The Spout creates the listener and listens for English tweets.
    - topologies
        - tweetwordcount.clj
            - The tweetwordcount file is a Clojure file that describes the topology. The tweet spout (tweets.py) and parse tweet bolt (parse.py) have a parallelism value of 3 and the count bolt (wordcount.py) has a parallelism value of 2.
    - virtualenvs
        - wordcount.txt
    - README.md
    - config.json
    - fabfile.py
    - project.clj
    - tasks.py
- README.txt
    - This file outlines instructions to run all of the functions of this application.
- Twittercredentials.py
    - This file contains the authentication keys used to access Twitter data.
- finalresults.py
    - This file contains a program to return the total number of word occurrences in the stream as outlined in the exercise description. When a word is passed, it will count the number of occurrences for that word. When no argument is passed, it will show a list of all words in the stream with the total count of occurrences, ordered alphabetically. If too many arguments are passed, the program will exit and an error message will be shown.
- hello-stream-twitter.py
    - This file was provided as an example in the exercise_2 assignment materials.
- histogram.py
    - This file contains a script that takes one string of two values separated by a comma and returns all the words with a total number of occurrences greater than or equal to the first value and less than or equal to the second value. When no argument is passed or when too many arguments are passed, the program will exit and an error message will be shown.
- histogram_chart.py

    - This file contains a script that builds a bar chart of the top twenty used words in the Twitter stream. The bars are built using asterisks based on the count for each word.
- psycopg-sample.py
    - This file was provided as an example in the exercise_2 assignment materials.