Megan Pera

Twitter Word Count

Application Architecture

MIDS Summer 2016
W205: Data Storage and Retrieval

Application Idea

As of summer 2016, about 500 million Tweets are sent per day. It is one of the biggest social media applications available to the public, and is widely used by people all over the world. Being able to capture, store, and glean information from Tweets gives the ability to track general trends in social dialogue at a specific point in time.

This application captures English-language tweets as they appear in the Twittersphere, parses them into individual words, and stores the words in a database along with a count of how frequently the word appears over the streaming time frame. The application can be launched to capture and store the most salient words of the hour.

Description of the Architecture

This application captures Tweets, parses them, counts the words, and stores individual words and word counts in a database.

To capture Tweets, this application uses the Python tweepy library to interact with a Twitter application created on https://apps.twitter.com. Using the Twitter app and tweepy, Tweets are captured and streamed into a Tweet-spout. This Tweet-spout is the entrance to a streaming pipeline that is executed using Apache Storm.

The Tweet-spout filters incoming Tweets and sends all English Tweets into the Parse-bolt of the Storm architecture. The Parse-bolt splits tweets into individual words, and sends these words to the Count-bolt part of the Storm architecture. The Count-bolt updates the word count for each incoming word in a Postgres database called “Tcount.” Words and counts are stored in a table called “Tweetwordcount,” which accepts the output of the Count-bolt. Outputs are read from the database using the Python library psycopg2.

Architecture Summary	Technology
Tweets are streamed from Twitter	Twitter App & Python: tweepy
Tweets are fed into the Tweet-spout	Apache Storm
English Tweets are sent to the Parse-bolt	Apache Storm
Individual words are sent to the Count-bolt	Apache Storm
Words and counts are updated in the Tcount database	Apache Storm, Postgres
Outputs are read from the database (summaries or whole)	Python: psycopg2

Directory and File Structure

All files are stored in a directory called “exercise_2” in Github. To obtain the complete files for this application, use the command “git clone https://github.com/mpera/exercise_2”.

The Apache Storm structure is stored in the tweetwordcount directory. The file structure for this directory is summarized below.

Sub-directory	File contents	Purpose
topologies	tweetwordcount.clj	This file is the Storm “map”; dictates execution of application
src/spouts	tweets.py	This file is the Tweet-spout script
src/bolts	parse.py, wordcount.py	These files are the Parse-bolt and Count-bolt scripts
virtualenvs	tweetwordcount.txt	Contains information about dependencies

The finalresults.py and histogram.py files in the main exercise_2 folder contain scripts that can be run after the application has collected data into the Postgres database. These scripts read the database and return outputs that summarize the words stored therein.

Necessary Information to Run the Application

The application requires that the machine be equipped with the following:

- Python 2.7
    - psycopg2
    - tweepy
- Streamparse
- Lein
- Postgres

The machine also must have the above outlined tweetwordcount files. Additionally, port 5432 needs to be open and Postgres needs to be running. To set up the Postgres database and data table, run the db_setup.py.

Detailed information on dependencies and required installations can be found in the README.md file in the exercise_2 directory.