

General Overview

The script that we created connects to a Mongo database that we run locally. The main operations in our script are: search for tweets, search for users, list tweets, list users, and compose a tweet. The function `search_tweets` retrieves tweets that contain all the specific keywords. It allows the user to view the basic details of the tweet found and to select a specific tweet to view its full details. The function `search_users` lets users search for other users by a keyword present in the user's display name. It returns a list of users matching the criteria, and similar to tweets, allows viewing full details of a selected user. `List_top_tweets` will list a certain amount of tweets that is determined by the user. When the function finds a matching tweet it will display the id, date, content, and username of the person who posted it. `List_top_users` will display an amount of users determined by the user, once the function finds a matching user it will display the username, displayname, and followersCount with no duplicates. If a user chooses to view more information they can choose a specific user to view all their details.

Details

This script is designed to load tweet data from a JSON file into a MongoDB database. The script connects to a local MongoDB instance on a specified port. It checks if a collection named "tweets" exists; if it does, it drops the existing collection and creates a new one. The script reads each line of the file, which is expected to contain one JSON object per line. Each line is converted into a Python dictionary using `json.loads()`. These dictionaries are then accumulated into a batch list. Once the batch reaches a size of 10,000, it is inserted into the "tweets" collection in the database using `insert_many`. This process reduces the number of write operations to the database, enhancing performance especially when dealing with large datasets. After looping through the entire file, if there are any remaining JSON objects in the batch (less than 10,000), they are also inserted into the database.

Testing

By comparing the function outputs to the expected variables detailed in the question, we can assure that the functions are performing correctly and are ready for testing. Our methodology ensures that no errors will occur with any tests that pertain to the assignment, as our script is built to account for all errors we may encounter.

Break-Down Strategy

Every member of our team put in their best effort on the project, agreeing on how to divide the work fairly. We kept each other posted about our progress and what needed to be done next through regular messages on Discord. Everyone jumped in to handle their parts when they could. Initially, James and Eason tackled phase one, while Rayan and Luke took on phase two. After finishing with phase one, James and Eason also pitched in to help with phase two. Trust was key to our approach; we counted on each other to get things done on time and correctly. This trust meant we didn't need to check in on each other too often, which made us more efficient.

User Guide

1. Running the script
 - a. Execute the script
 - i. Ensure pymongo is installed beforehand and if it is not typed in pip3 install pymongo. Go into the terminal and type mongod --port "port number" like mongod --port 27012. If that doesn't work, then do mongod --port 27012 --dbpath ~/ (make a folder) so like mongod --port 27012 --dbpath ~/folder. Then do python3 load-json.py "json file name" "port number" like (python3 load-json.py 10.json 27012). After the JSON file is loaded into Mongo DB, type in python3 operations.py "port number" like (python3 operations.py 27012) to interact with the database.
 - b. Choose which function to run

2. Main Menu

- a. Search for tweets
 - i. Input keywords to retrieve relevant tweets, the user may enter more than one keyword separated by a space
- b. Search for users
 - i. User must input a relevant name or location, in order to search for users
- c. List top tweets
 - i. Users must choose how many tweets will be displayed, then the function will display as many tweets as the user pleases. Tweets may be selected by the user if they want to view more details for specific tweets.
- d. List top users
 - i. Users must enter how many users they want to view at a time, and similarly to list top tweets the user can select specific users to view more details about the selected user.
- e. Compose tweet
 - i. Users may choose to enter anything as a tweet, the user may not enter nothing as a tweet.
- f. Logout
 - i. The user may log out if they do not wish to continue.

Users that input anything other than a digit from 1 - 6 will be hit with an error message, the user must only select a number from 1 - 6 based on which operation they want to use.