Tweet Collector and Search Engine

CS 172 Project Report: Part B

Armando Gutierrez SID: 861213968 Allison Nguyen SID: 861204602 Tianyu Liu SID:862122385

6/10/2020

Collaboration Details

Armando and Anthony worked on the lucene program which parses and indexes json files from Part A of the project. This was done in its own program before working on the web app development. Once this was finished, Armando and Allison worked together on developing the web application. This step integrated the previous lucene program with the creation of the search engine using html and a java servlet class.

Overview

In Part A of the project, we collected the json objects of the tweets and wrote the data to an output.txt file. We updated this code to output only the fields of the json object that we cared about. This includes the username, tweet contents, webpage title, date, location, and hashtags.

As briefly mentioned in the Collaboration Details section, our group implemented the lucene searcher before the web application. The lucene program reads the json objects from the output.txt file created in Part A. It then parses the input to store these text fields into a document. Once all the documents are created, it indexes them. Since our lucene program was based on the provided demo code, our ranking algorithm uses the boost Hashmap to rank the above mentioned fields in the user's search query. Currently, our program ranks the fields as follows:

- Username (0.25)
- Tweet Contents (0.35)
- Webpage/Link Title (0.20)
- Date (0.10)
- Location (0.05)
- Hashtags (0.05)

Once the lucene program was finished, we pasted it into a java servlet program that would allow us to create a web based interface. For this, we used Eclipse for Java EE and the web server Resin 4.0. We created a basic search engine using the SearchEngine.html file. We then created a Servlet Class which reads the user's search query, executes the integrated lucene program, and then displays the ranked results in a new page. This page displays the results from highest to lowest rank, displaying the username, tweet, etc.

System Limitations

One limitation in our system is the ranking algorithm. We used the MultiFieldQueryParser used in the demo code provided to us, so this particularly affects the date/time and location fields. Unless the user specifies these fields in the query, then the ranking algorithm cannot boost these fields. For example, if we specify tweets made on June 5, it will not prioritize June 4th tweets over June 3rd tweets, and so on. Another limitation is that the user must press the back button on the web page in order to enter another search query.

Execution Instructions

In order to execute Part B, you must have Eclipse for Java EE, Lucene version 7.3.1, and Resin Pro version 4.0.64. The steps are as follows:

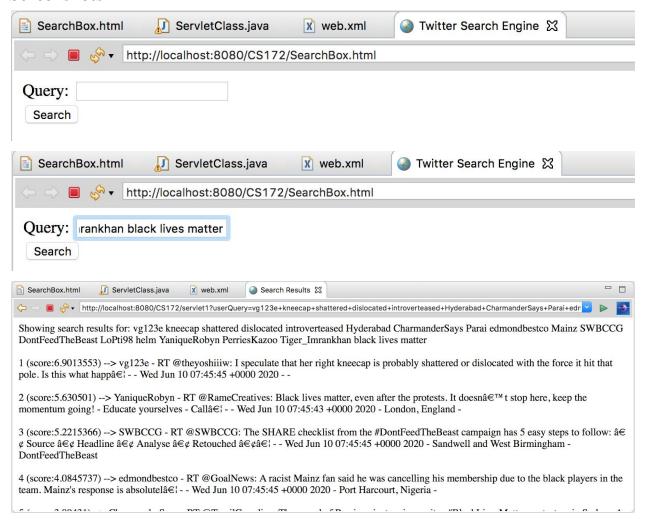
- 1. Download the CS172 Project zip file
- 2. Move the twitter.py and crawler.sh files to your desktop
- 3. Execute crawler.sh and make sure that an output.txt file is created in desktop (See execution instructions from Part A for more information)
- 4. Move output.txt to the CS172 Project directory
- 5. Import the CS172 Project into Eclipse (Java EE)
- 6. Click on the "Servers" tab and start the resin web server
- 7. Open a tab on your web browser and go to http://localhost:8080/CS172 Project/SearchEngine.html
- 8. Enter search query and view the results
- 9. To enter another search query, press the back button to be taken to the search bar

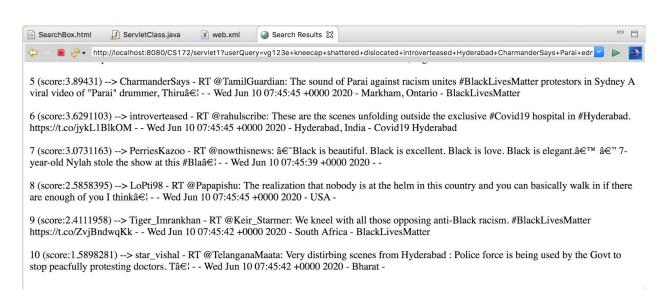
Link to Demo Video

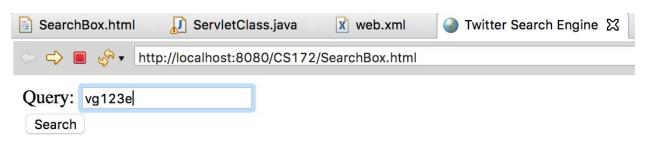
https://www.loom.com/share/07f50e7f39f14833987f24b6911bc3be

NOTE: I talk slowly, so I recommend watching it at **1.2x or 1.5x speed** to get it around 5 minutes. Thank you!

Screenshots







Showing search results for: vg123e

1 (score:1.3367769) --> vg123e - RT @theyoshiiiw: I speculate that her right kneecap is probably shattered or dislocated with the force it hit that pole. Is this what happâ \in - - Wed Jun 10 07:45:45 +0000 2020 - -