Justin Schopick
Kushagra Singh
Chris Sultzbaugh
CS 172: IR
June 8, 2018

Twitter Searcher Report

Collaboration:

We all got together to divide the labor and research the coding process of this portion of the project. The information provided made the indexing and search algorithm much easier. Once we finished researching together, Kushagra coded the json parsing and indexing, Chris coded the query search function, and Justin coded the webpage to conduct the search and connected it to the Spring server. We all worked together to complete the report by adding the information about our specific coding work.

Overview of System:

Our system uses Lucene, JSON Simple, Spring Boot, and AngularJS.

In order to parse the json file, we use JSON Simple. We go line by line, and file by file (in which we store all the tweets, one line per tweet), and then use the JSON parser in order to parse a tweet and create a JSON object out of it. Once the JSON object is created, we take the relevant information stored. These were as follows: tweet id, coordinates, location (both coordinates and location as a string), tweet body, timestamp (both in epochs in ms, and standard), hashtags, username, title of link (if there is any), and username of the tweeter.

Once we parse a tweet, we create an inverted index using Lucene in order to index it. We index it based on all of the features mentioned above. We also use lucene and it's Standard

Analyzer in order to search a query. A query is parsed using a simple string scanner, where we differentiate between hashtags (if it starts with '_'), user mentions (if it starts with '@'), location (if it starts with 'loc:'), link titles (if it starts with 'title:'), and text (not having any of the tags mentioned above). In addition to this we use the time (in ms) to increase the weight of the newest tweets. The power of these weights are as follows:

- Text : 1.0, Location: 2.0, Hashtags: 1.5, Username: 1.75, Link Title: 1.25, Time: 2.5

These weights were given based on the specificity needed of a query(i.e. a location is pretty specific, versus text could mean anything).

Once we were able to analyze a query, we used Spring-Boot to create a REST service for our backend. This waited for an input from our front end, which then sent it in to our function to create an index (if it wasn't already created), and then to query parse. We used AngularJS for our front-end.

Limitations of system:

Our system had a couple limitations, the main ones were involving the query. If a user puts in a '#' the word succeeding it is not read (i.e. if someone has the query: 'UCR #Highlander' , then #Highlander would not be parsed). We got around this, by using '_' as the tag to represent hashtags.

Another limitation with the query is that if the user uses the 'loc:' and 'title:' tag it needs to be handled towards the end (in either order). This is because locations and titles in can have multiple words (i.e. 'Los Angeles' , or 'Sheraton Hotel'). This limitation is done to ensure that text for the location/ title is not confused for text in the tweet.

Another limitation was the tweets. Since they are all stored, they are stale. In addition to this, it is hard to compare fresh tweets (as our current time is very different from the time of the tweets), in order to overcome this we used the time of our newest tweet as our 'current time'. If we were to change to a system that indexed tweets as they came, we would remove this and use the current time.

Deployment:

Software Requirements:

- Java Development Kit must be installed in order to compile and run the program
- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Maven, Node.js, npm, and Angular.js - Install via terminal. Angular.js requires Node.js version 8.x or greater and npm version 5.x or greater to run without errors.
    - Maven: sudo apt install maven
        - To check version: mvn -v
    - Node.js: sudo apt install nodejs
        - To check version: node -v
    - npm: sudo apt install npm
        - To check version: npm -v
    - npm install -g @angular/cli
        - To check version: npm @angular/cli -v

External Packages Needed(these are all inside of the submission folder):

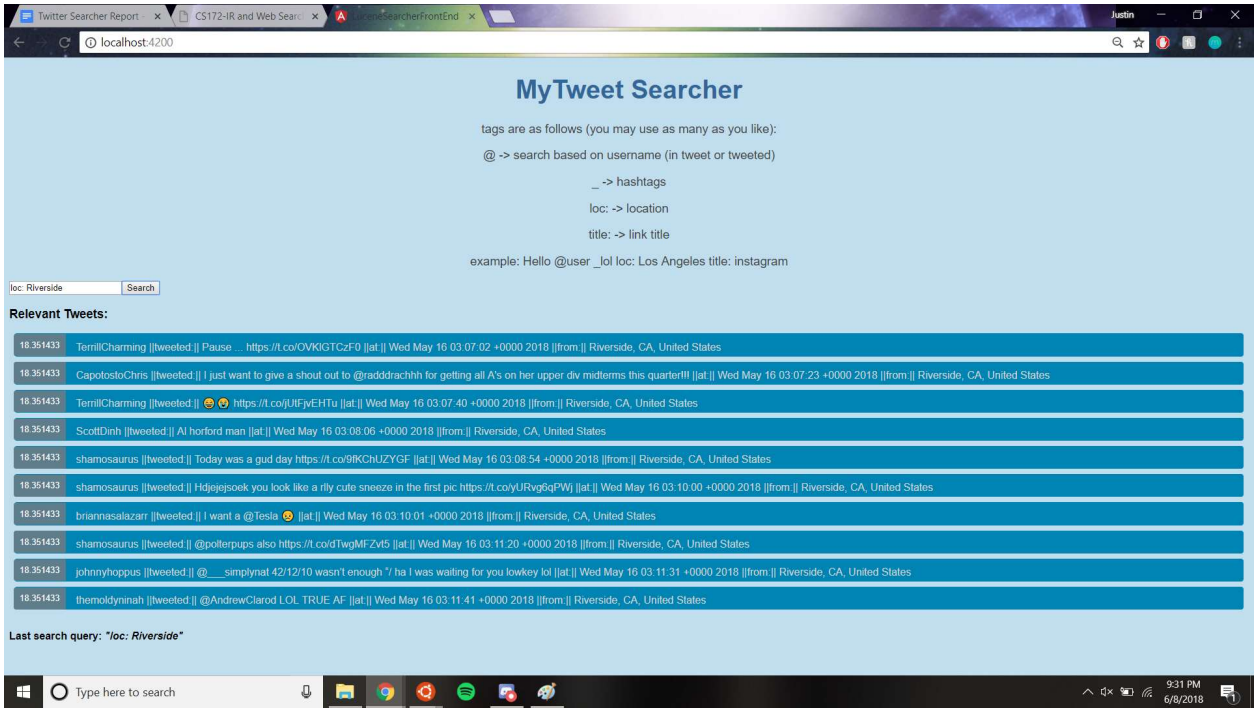
- json-simple-1.1.1.jar
- lucene-core-7.3.1.jar

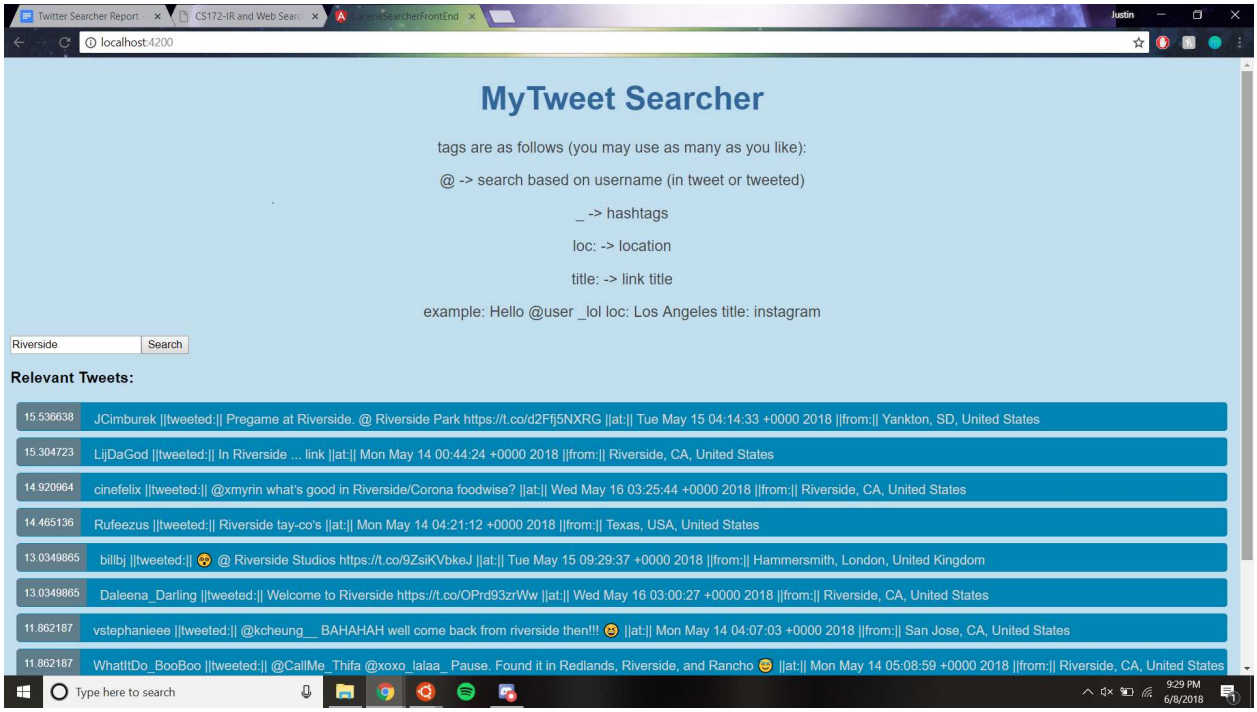
- lucene-queryparser-6.6.0.jar

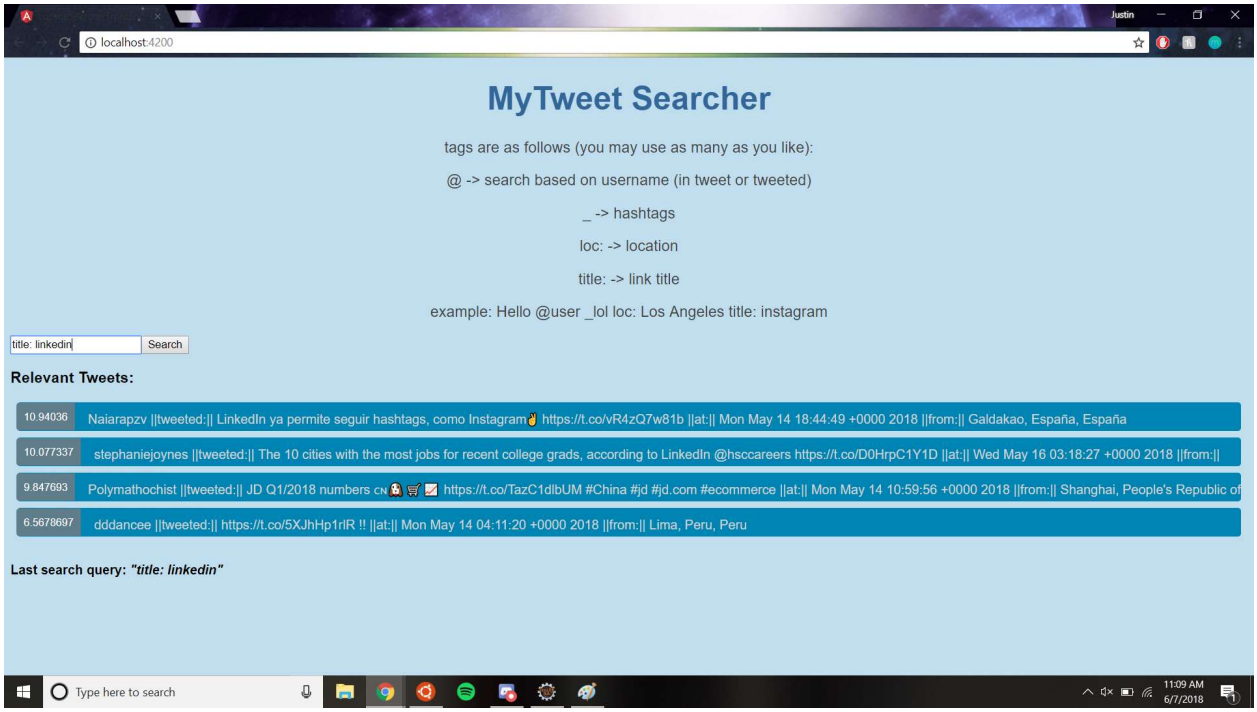
To Run(must be on unix/linux):

1. Download the myTweetSearcher zip file and unzip it to your desired location
2. Open your terminal and find the location where you stored the myTweetSearch directory
3. Open the myTweetSearch directory(e.g. cd myTweetSearch)
4. While in the myTweetSearch directory, type "./runSearch.sh" into the terminal without quotation marks to start the Spring server for the Back End code
5. Repeat steps 2-3 in a new terminal window. When in the myTweetSearch directory, type "./runFrontEnd.sh" without quotation marks to start the Web Server to use the program.
6. Note: Since we did not include our index folder, in order for our back-end code to create the index, you must search once before it starts. Since we are submitting a small fraction of our data set, some recommended queries are: "hello", "los angeles", "loc: new york"

Screenshots:




Twitter Searcher Report
CS172-IR and Web Searc
localhost:4200
Justin
MyTweet Searcher
tags are as follows (you may use as many as you like):
@ -> search based on username (in tweet or tweeted)
_ -> hashtags
loc: -> location
title: -> link title
example: Hello @user _lol loc: Los Angeles title: instagram
Riverside
Search
Relevant Tweets:
15.536638 JCimburek ||tweeted:|| Pregame at Riverside. @ Riverside Park https://t.co/d2Ffj5NXRG ||at:|| Tue May 15 04:14:33 +0000 2018 ||from:|| Yankton, SD, United States
15.304723 LijDaGod ||tweeted:|| In Riverside ... link ||at:|| Mon May 14 00:44:24 +0000 2018 ||from:|| Riverside, CA, United States
14.920964 cinefelix ||tweeted:|| @xmyrin what's good in Riverside/Corona foodwise? ||at:|| Wed May 16 03:25:44 +0000 2018 ||from:|| Riverside, CA, United States
14.465136 Rufeezus ||tweeted:|| Riverside tay-co's ||at:|| Mon May 14 04:21:12 +0000 2018 ||from:|| Texas, USA, United States
13.0349865 billbj ||tweeted:|| @ Riverside Studios https://t.co/9ZsiKVbkeJ ||at:|| Tue May 15 09:29:37 +0000 2018 ||from:|| Hammersmith, London, United Kingdom
13.0349865 Daleena_Darling ||tweeted:|| Welcome to Riverside https://t.co/OPrd93zrWw ||at:|| Wed May 16 03:00:27 +0000 2018 ||from:|| Riverside, CA, United States
11.862187 vstephanieee ||tweeted:|| @kcheung__ BAHAHAH well come back from riverside then!!! ||at:|| Mon May 14 04:07:03 +0000 2018 ||from:|| San Jose, CA, United States
11.862187 WhatItDo_BooBoo ||tweeted:|| @CallMe_Thifa @xoxo_lalaa_ Pause. Found it in Redlands, Riverside, and Rancho ||at:|| Mon May 14 05:08:59 +0000 2018 ||from:|| Riverside, CA, United States
Type here to search
9:29 PM
6/8/2018


localhost:4200
Justin
MyTweet Searcher
tags are as follows (you may use as many as you like):
@ -> search based on username (in tweet or tweeted)
_ -> hashtags
loc: -> location
title: -> link title
example: Hello @user _lol loc: Los Angeles title: instagram
title: linkedin
Search
Relevant Tweets:
10.94036 Naiarapzv ||tweeted:|| LinkedIn ya permite seguir hashtags, como Instagram https://t.co/vR4zQ7w81b ||at:|| Mon May 14 18:44:49 +0000 2018 ||from:|| Galdakao, España, España
10.077337 stephaniejoynes ||tweeted:|| The 10 cities with the most jobs for recent college grads, according to LinkedIn @hsccareers https://t.co/D0HrpC1Y1D ||at:|| Wed May 16 03:18:27 +0000 2018 ||from:||
9.847693 Polymathochist ||tweeted:|| JD Q1/2018 numbers cn https://t.co/TazC1dlbUM #China #jd #jd.com #ecommerce ||at:|| Mon May 14 10:59:56 +0000 2018 ||from:|| Shanghai, People's Republic of
6.5678697 dddancee ||tweeted:|| https://t.co/5XJhHp1rIR !! ||at:|| Mon May 14 04:11:20 +0000 2018 ||from:|| Lima, Peru, Peru
Last search query: "title: linkedin"
Type here to search
11:09 AM
6/7/2018