Twitter Keywords Search

Tianyi Tang

http://www.tty8128.com
https://github.com/8128/TwitterKeywordSearch

Products

Command line tool

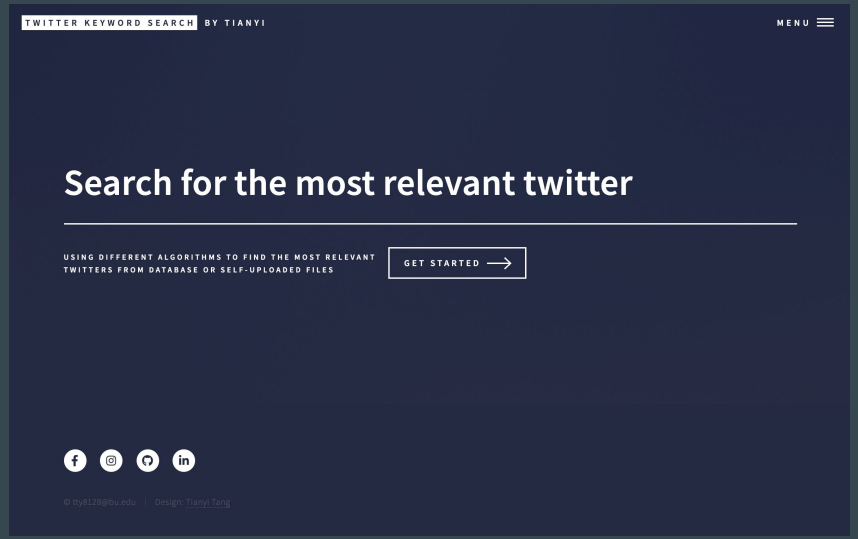
```
--------------------------------------------------------------------------------
|                        Twitter Keyword Searcher                               |
|                        Developed by Tianyi Tang                               |
|                                                                               |
|             return the top relevant twitter with your keywords                |
--------------------------------------------------------------------------------

Enter 1 to enter your own CSV file with absolute path,
enter anything else to use default CSV
2
Loading Default CSV File
1041776 twitters loaded

Start to create reversed index map
Reverse Index time: 4744ms

Please Input Your Keywords.
Split them using spaces, and end with enter
```

Spring Boot Website http://www.tty8128.com



Algorithm - Matching

The keywords will be stored into HashSet, each for O(1) time complexity

Split the twitter into words, and iterate through every single word, use the contains function to check whether the word match any word inside of the hashset, each word for O(1) time complexity

An relevance value will be generated using the number of matching.

Algorithm

Heap Sort

Initialize a new PriorityQueue(which implements the heap in Java), set the size to the needed size, and override the Comparator to compare them with the relevance value, and make it a min heap

Iterate through all the twitters and compute their relevance values. When the heap size reaches the needed size, and after that the new twitter's relevance value is larger than the min value in the heap, then poll out the min twitter and add the new twitter

Bucket Sort

Because the twitter's length is smaller then 140, the largest relevance value will not be larger than 140, so buckets of 140 will be initialized.

Iterate through all the twitter and calculate the relevance between the keywords and twitters, then use it to store the twitter to the

bucket[relevance value]

Finally, iterate from the last index of the bucket, add twitters until the list reaches the needed size

Improvement

- If users want to search several times ?
- Use reverse index

After load all data to list, iterate through all the twitters and store them to a HashMap

HashMap<String, HashMap<Integer, Integer>>

First Map: Key - Word / Value - Second HashMap

Second Map: Key - index of twitter in the data list / Value - this word's frequency

When keywords come in, new a HashMap to store twitter and overall frequency, and compare frequency using Java's Collections.sort and BucketSort

Website development

http://www.tty8128.com

- Using Spring Boot Framework, MyBatis to connect to MySQL
- MySQL (t_twitter / t_twitter_temp)
- HTML5/CSS/thymeleaf

Con:

Cannot use MySQL database on SCC, so it is deployed on Amazon EC2

Thank You