Intelligent Web Assignment

team name: unkonwn
member: Pengyuan Zhao, Likang Cao, Shangshu Lu

1.

Introduction:

Question 1 is about tracking public discussions, if user input some keywords or hot topics, the website will output a list of messages containing the required keywords. And all the output should follow the standard.

Issues:

Using lab example code is a good solution for this question, but to find a proper twitter api and select query parameters is the most challenge thing. After getting data from Twitter server, push back all the useful tweets to the web client is also a challenge thing, the data sent back to the server is JSON format and client should transfer it to a readable format first. And the most difficult thing is to display who retweets the tweets and provide a hyperlink for the message. We could get the tweets id and retweets id, but there are still some work to do to solve this question.

Design choices:

This part we use Search Tweets api to get useful information according to user's input. According to the requirement, we should use keyword and geographic region(like latitude and longitude) to get the discussions. Search Tweets api provides two useful parameters which are query and geocode that can specify the requirement.

requirements:

The design nearly comply with the requirement, it outputs a standard information that covers the hot topic, but the further function like display who has retweeted each message is not solved in this design.

Limitations:

In my opinion, the extension of this part is that the web may return a graph that shows the same venues these user both visited.

2a:

Introduction:

This part of function will extract tweets form specific users and analyze the high frequent keywords from it in the last few days. The keywords will be sorted depending on the total amount. Moreover, the most frequent Y keywords used in the last Z days must be displayed (with associated total count, as well as count for each users).

Issues:

1.The tweets from users will contain a lot of punctuations and blank. This part must be considered otherwise there would be a plenty of illegal keywords come out.
2.The high frequency keyword may not exist in a specific user's tweets, which should also be considered.

Design choices:

The design of this part is not complex, but the algorithm of implementation is quite sophisticated.

First of all, we need extract tweets from a specific users. The days and users' screen_name can be controlled by the search/tweets API. Following by token the tweets and eliminate the punctuations and blanks, we store them into arrays.
After we repeat this step for a few times(which are limited by the number of users entered), we can get a series of arrays which are contained all text of tweets we get from the API. We can combine them all, and divide it into individual words. Next, we could calculate the frequency of each words and sort them in the order. According from the keywords number we entered, we can put these specified high frequency keywords into an array. At last, we compare this array with the arrays of each users tweets, and get the date of each keywords for specific users.

Advantages:

we can decide the details of data we need by written algorithm, which can make our system stronger and more flexible.

Disadvantages:

The process of getting and handling data is quite complex. All parts of implement are designed in the same function, there is lacking in several interface and encapsulation to control these functions. So it is quite difficult to debug the code and identify which part comes out the problem in a short time.

Requirements:

we can return the correct information to the frontpage, including the screen_name and the keywords we entered. (which are based on the number and days). However, the data is not shown into a perfectly correct format. We can send the correct array which store correct data to the frontpage, but failed to show it into a table. The reason is that the frontpage concern the other part of this project and we have limited time to fix it.

Limitations:

1.The principle of token tweets is quite loosen due to there is not specific requirement in assignment. If it does, it is quite easy to add a stop-list to do the token work.

2. The way of extract and cope with data is complex. We should divide the whole requirement into smaller parts and make it happened into separated functions. So that we can debug our program more effective and more readable.

2b:

The aim of this question is to find venues the user visited in last several days, the main strategy to solve this problem is shown below: Firstly, I used the search/tweets API to get the user’s tweets with Swarmapp check-in messages; then, find use split() to find the check-in id from user’s tweets; after that,use foursquare checkins resolve API(https://api.foursquare.com/v2/checkins/resolve) to get venue name through the check-in id we got.

Issues:

There are so many Issues I met when I was doing this problem; firstly, I need to look for a correctly API from both twitter and foursquare API documentations to get desire date. Secondly，I had to analysis data I got to retrieve the desire information(ie. Check-in Id),,after that, the largest issue is that some users cannot find through Twitter and foursquare API, so that it makes a large challenge during testing the system.

Design choice:

For overcoming the problem of searching user's tweets with check-in information, I used search/ tweets API swarmapp/com/c/ as query and set parameter "from "be a given user, and I used setDate() to get the days we need.

For finding out all the user's tweets with check-in message, I split 'expanded_url' properties of tweets to get check-in ids because it is easy to find that all the check in messages are shown by the format swarmapp/com/c/check-in ID, After that, I used foursquare API 'https://api.foursquare.com/v2/checkins/resolve' to get venues name.

For storing the information, I built two tables in the database which are named "users Information"(which used to store "user name", "user id", "user location" and "user's description") and "locationVisited"(which used to store user id, venue id and venue name) ,I design them with the concept of relational database, I used "user id" in the table "usersInformation" as the foreign key in the table "locationVisited".

Advantage:

The advantage of this system is that it can basically implement all the functionalities of assignment demand.

Disadvantage:

Firstly the system is not so robust and always suffers from all kinds of issues, such as some users cannot find through this system.
Secondly, the system is not so efficiency and always cannot retrieve some user when processing.
Finally, the database is not so efficiency when accessing data.

2c:

This part is to get a list of user by querying a venue's name and check who have visited this venue in a limit days. And the more work is to provide a hype link that can get more information about the user.

Issues:

There are so many Issues I met when I was doing this problem; firstly, I need to look for a correctly API from both twitter and foursquare API documentations to get desire date. Secondly， I had to analysis data I got to retrieve the desire information(ie. Check-in Id),,after that, the largest issue is that some users cannot find through Twitter and foursquare API, so that it makes a large challenge during testing the system.

challenges:

To get the list of user is quite difficult and provide the user's detail.

Design choice:

The design of this part use two apis to get the correct output, first is using foursquare venue search api to get the venue's coordinate and then use tweeter search api to get the users by querying the checkin information.

requirements:

This solution partly fulfilled the requirement. It can return a list of users who have visited the venue but can not return the detail information of the user.
limitations: In my opinion, the extension of this part is that the web may return a graph that shows the same venues these user both visited.

3.web interface:

introduction:

The web interface uses a index html file to provide four part of the project. They are searching history part, tracking discussions part, queries about specific users part and result output part. This web interfaces using a web open source frame work - skeleton-2.4.0 to make the web interface more clear and beautiful.

challenges:

To make the web interface more clear, there are a lot of work to find a proper framework and redesign the css style for each element. And create a beautiful web interface is very challenging.

design choices:

The web interface should provide clear forms and guidance for users to query the different data.