Name : Sambit Mishra
UIN : 720002013
Assignment : Homework 2

The zip file contains three files , **part1.py** , **part2.py** and **part3.py** corresponding to 3 parts of the question along with this Readme file.

part1.py:

In order to compile and run the program, we need to have a jason file which contains the tweets and will be input to this file. Input the follwoing command at the terminal:

python part1.py mars_tweets_medium.json

In the above case the jason file is the same directory as part1.py. If the jason file is not in the same directory , give the relative path accordingly. The below example contains the jason file in one directory above , where the code is stored.

The above will run the program and will ask for user query input :

Input Query

Once you input the query, eg. Mars. The top 50 results will be diplayed in the format:

user: tweet: tweet_id: score:

Example:

python part1.py ../tweets_vector.json

Input Query
spade

user:bob tweet:spade spade heart heart tweet_id:5 score:0.707106781187

user:Diane tweet:spade spade spade heart diamond tweet_id:3 score:0.464999039074

user:Alice tweet:spade club tweet_id:4 score:0.383332888988

part2.py

This code will take a jason file as input and print the top 50 user based upon the page rank. The results will be displayed in the format:

id: screen_name:

In order to compile and run the program we need to type in the follwoing command at the terminal:

python part2.py ../mars_tweets_medium.json

The top 50 results are displayed after the execution. If there are lesser results then all the results are displayed.

Example:

python part2.py ../tweets.json

Users Ranked according to Page Rank Score are:

id:1 screen_name:Alice
id:2 screen_name:BOB
id:3 screen_name:charlie
id:4 screen_name:Diane

part3.py

In this part of the homework we were required to build an "Integrated Page Rank System".

Approach:

I first calculate the tf-idf score for a given query using the part1. This will give me a list containing tuple of (tweet_id,score,user_id,screen_name,tweet).

By using the second part I will calcuate the user and their corresponding page rank score.

The above two are input to my integrated page rank system. I iterate through every tweet in my tweet_list from part 1, find the correspoding user from part two and multiply as **[0.6 * tf-idf score * 0.4 *page rank score]** to get a new score for the corresponding tweet. This tweet along with the user id ,new score , screen_name is stored in a new list. I iterate till my new list has 50 tweets since we will display the first 50 tweets only.

Once my iteration is complete I will have a new list containing the tweets , user id ,new score , screen_name. I will sort this list in descending order of new score such that my sorted list will contain the tweet with highest score on the top. I will then display the result.

Reasoning behind the approach:

Since we are bascially trying to answer a Query of a user , I have given more weightage to my query followed by relevance of user [page rank]. In the above formula you will see that I have used a multiplying factor of 0.6 for query and 0.4 for user. If we change these values then we can change the order of relevance. By multiplying and finding a new score we ensure that we take into consideration both the tf-idf and page rank of user to rank our final results in this integrated page rank system.

Comparison of All 3 parts:

Part1 of the Assignment deals with pure vector retrieval where we return results purely based on the tf-idf score. This does not take into consideration the user relevance(page_rank). In normal search scenario for example,Wikipedia might have a better tf-idf score for a query say "Gangnam Style", but the YouTube Video featuring the video is more relevant. Thus the You Tube video should feature first rather than Wikipedia. Moreover , it is most likely that the user who searched for Gangnam Style is most interested in video rather than reading about the Singer Psy and Gangnam Style which Wikipedia will provide

Part2 of the assignment just ranks the users. Only using this is also not relevant. All our user relevance should be tied to a query. Just take the example of Twitter data corpus. There could be some very hit users. But suppose we are quering for something which these hit users have not tweeted about much , then we should not display 1 off the odd tweets by them. Similarly the page rank of www.cnn.com will be more than www.cricinfo.com. But if we query for cricket we should display www.cricinfo.com as our best result rather than www.cnn.com which has a better page rank. So ideally page rank should be tied or build on top of query result.

In Part3 we have integrated both the above systems and is prefferd.

Execution of part3:

In order to compile and run the part3 of the code , use the follwing command:

python part3.py ../mars_tweets_medium.json

On executing above we will be asked to input an query:

Input Query

We then input a query and it dislays the top 50 results where the page rank along which tf-idf socre have been considered to deliver the result

Example:

python part3.py ../mars_tweets_medium.json

Input Query

mars

name:HOT_ROB_ tweets:['mars']
name:JordanFehr tweets:['mars']
name:jameslano tweets:['mars']
name:WatchJeff tweets:['mars!! mars!! mars!! mars!!']
name:outdoortype tweets:['mars!']
name:sonerdlike tweets:['mars mars mars mars mars!!!']
name:returnofthe_mac tweets:['#mars']
name:katiewicks tweets:['mars']
name:Sean_LookAlike tweets:['mars']
name:btumpak tweets:['mars'] .....
.....

[Note all the 50 results are displayed once you run in terminal. I have just pasted the first few results].

Observation:

The results of part 3 are more smoothed and are not biased to only 1 factor like page rank or tf-idf score. If we run the same query e.g. “mars” on part 1 and part3 we can observe the difference in the ranking through the order in which the results are displayed. Part1 simply displays the tweet with highest tf-idf score where as part3 takes both tf-idf and page rank into consideration to rank and display the result.

To compare run the same query e.g. “mars” on part1 and part3. We can clearly see the difference in results.

Preferance:

I will prefer a ranking system which will consider both the tf-idf score and page rank to deliver the result, much like the “Integrated Page Rank System” which is build in the 3rd part of the assignment.

4th Part:

The URL to my page is :

http://sambiturjnaswxkfjjkn.wordpress.com/