“finalVector.pickle” and “title.pickle” are from part 2. With the result of PageRank(recorded in finalVector.pickle) imported, “part3.py” can skip the time-consuming step of calculating PageRank for all users again. We can focus on the core idea of integrating cosine similarity(part 1) and PageRank(part 2). The program here is primarily based on “part1.py”

It takes about an average 4 minutes for returning of result.

1. Cosine Similarity vs. PageRank:
   1. Cosine Similarity:
2. Characteristic for tf-idf:

TF-IDF is used to give a document a score based upon some query. The score changes based upon the query, and without a query there is no score.

1. Problem for tf-idf:

Many spammers try to manipulate their page. Earlier attempt is to modify a page to repeat the terms many many times (trying to increase the TF/IDF score). The evolution of Page rank has mitigate this to some degree because page rank in based on "out-of-page" information that the site owner is much harder to manipulate.

* 1. PageRank:

1. Characteristics for PageRank:

PageRank assigns a score to a document based upon the documents it links to, and the documents which link to it. The score does not vary depending on the query used (i.e. it is a global ranking scheme).

1. Problem for PageRank:

But people use Link-farms to game the page rank algorithms. The idea is to trade links between different domains. There is active research in this area about how to catch these patterns and discount their ranks.

1. My method to integrate Cosine Similarity with PageRank:

My method to integrate PageRank (per user) and cosine similarity (per tweet) is to add them together with a weight allocated to each part. In my program, I assigned the weight of each part equally as 1.That means final\_score( tweet ) = cosine\_similarity\_score( query, tweet ) + PageRank( tweet author ).

The reason for this integration comes from my intuition that

1. The more similar a tweet is to a query, the more valuable the tweet is.
2. Since our tweet corpus is all about the same topic (Mars), the more a user is mentioned by others, it is more likely that he/she knows well about the topic and has a good reputation in this field.

Therefore, I choose my integration method as above.

1. Summary:

In general, ranking with the new method is superior to ranking with cosine similarity or PageRank separately. It is a more objective ranking method. Even if a webpage with certain term injected everywhere in the content, it will not be ranked very high because few users want to add links to this spam webpage (low PageRank Score). Although my new method can still face some spam problem, it is trickier for spammers to achieve their goal.

1. Program Examples:

Take Cheng’s sample test as an example,

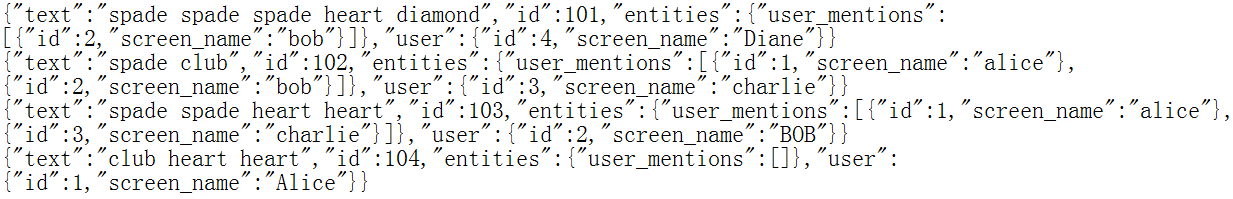


Figure 1 Cheng's Sample Test

The following screenshot is result of my part 1 program, which only applies tf-idf for ranking:

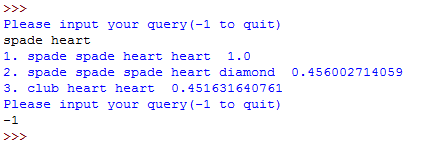


Figure 2 tf-idf result

“spade heart” is my query. The score at the end of each returned result is the cosine value between query vector and tweet vector. Tweet 1 is from Bob. Tweet 2 is from Diane. Tweet 3 is from Alice.

The following screenshot displays the PageRank Scores for each user:

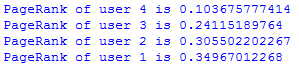


Figure 3 PageRank of users

Alice’s score is 0.34967012268

Bob’s score is 0.305502202267

Charlie’s score is 0.24115189764

Diane’s score is 0.103675777414

The following screenshot is result of my part 3 program, which integrates tf-idf and PageRank for ranking:

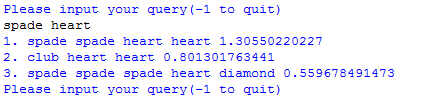


Figure 4 tf-idf + PageRank result

In comparison with figure 2, I found the second and third result changed their position. According to my method:

final\_score( tweet ) = cosine\_similarity\_score( query, tweet ) + PageRank( tweet author )

The final score of the second tweet from Alice (her PageRank Score 0.34967012268 plus her tweet’s tf-idf Score 0.451631640716) overtakes that of Diana (PageRank Score 0.103675777414 plus her tweet’s tf-idf Score 0.456002714059). So the tweet of Alice climbed to the second place with my new ranking system.

This result is meaningful. For query “spade heart”, Diana’s tweet has a slightly higher tf-idf score than Alice’s. They are almost the same. In other words, the two tweets are almost equivalently relevant for the query. However, Alice’s PageRank is much higher than Diane’s. That means Alice is more frequently mentioned by other users and is more likely to be an expert in the topic corpus. Alice’s words should be deemed more influential than Diane’s. Therefore, Alice’s tweet should rank higher, just like the return of my new ranking system.