Web Mining Programming Assignment 2

R01944027 林廷舟

PageRank

Implementation

My PageRank implementation is simple. The concept is described as following. At the beginning, each node has a prestige of 1.0. In each iteration, every node equally gives it's current prestige to the nodes that it links to with a damping factor D=0.85. As for those without outlink, we add links to every node for it.

I think the most important thing is not adding edge to every node for those without outlinks explicitly. The graph is usually sparse, adding them one by one is too time-consuming. To avoid this. I record down nodes without outlinks when building up the adjacency matrix. At the beginning of each round, I sum their prestige up and distribute them at once with the random surfing prestige 1-D. This avoids adding them one by one for each node. Every iteration looks like this.

```
1. s = SUM(prestige of nodes without outlinks)
2. Initial all prestige with 1 - d + d * s / maxnodes
3. For every node i with outlinks:
        Prestige of the node i reaches += Prestige of i in last iterations / # outlinks of i
```

And iteratively compute it until converging.

Findings

1. It seems that the document graph are usually sparse and dealing with sparse graphs are very important in information retrieval. In this task, if I don't precompute the prestige without outlink, 1 iteration takes about 2 min, which is totally untolerable.
2. It's significantly converging as iteration goes
3. Larry Page was so great that he founds Google and I can only sit here and write this assignment

LexRank

Implementation

I reused my code of Page Rank. First, I build up vectors for each sentence. Next, I compute all-pair similiarity for these sentence. Then I build up a graph G(V, E) where V are the sentences and (i, j) and (j, i)is in E iff similarity of sentence i and sentence j is greater than threshold T. I express such graph as an adjacency matrix and compute Page Rank according to this graph.

Output

The sentence ranked highest is 6. I think it's reasonable. The corpus is mainly about the Warrior - Spurs series of NBA playoff. Sentence 6 talked about "Curry" and it's performance on "shots"; "final" is also mentioned a lot to talk about his closing shot. I think is a nice summary.

Findings

If the threshold it set to 0.2. Sentence 37 comes to be the best one from the second place. If the threshold is set higher, I think the summary is not so good. Maybe it's because that it's over smoothed by 1-d. The resulting graph also became sparser as threshould arises and the computing time lowers.