Finding Most Influential Papers Based on DBLP Citation Network

By

Saurabh Pathak

Sonali Priya

INTRODUCTION

PageRank is an algorithm which is used by Google Search for ranking the web pages. It works on a link analysis algorithm. A page that is linked to by many pages with high PageRank receives a high rank itself. It requires few iterations to find the approximate PageRank values.

PageRank was developed at Stanford University by Larry Page and Sergey Brin in 1996 as part of a research project about a new kind of search engine. Sergey Brin had the idea that information on the web could be ordered in a hierarchy by "link popularity".

A page is ranked higher as there are more links to it. It was co-authored by Rajeev Motwani and Terry Winograd. The first paper about the project, describing PageRank and the initial prototype of the Google search engine, was published in 1998: shortly after, Page and Brin founded Google Inc., the company behind the Google search engine.

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

In this project we have tried implemented this algorithm on the DBLP citation network dataset. We have performed the ten iteration using the Hive Script to find the top ten ranked pages and used the map reduce to find the page index and the citations.

METHODOLOGY

Map Reduce Implementation:

1) **DBLP Input Format :** We have used DBLPInputFormat.java class where we parse the given DBLP-Citation-network V7 data. We have used several overridden methods of InputFormat like createRecordReader(), nextKeyValue(), getCurrentKey(), etc. to achieve this.

2) **Mapper:** We used PageRankMapper.java class to parse the output from the DBLP input format. We get the input in mapper in the format like #*…….\n\n\n!#. We parse this input in mapper to emit key as index of page and the value as reference index.

3) **Reducer:** We use PageRankReducer.java class to emit all the pages with its corresponding page references as a link graph.

Hive Script Implementation:

STEPS :

1) Loaded the link graph data into mpRedOp table.

2) Created a table IncOnly and selected only the indices which had incoming links.

3) Created a table which will have indices, all of it's incoming links, no. of outgoing links of all incoming links, page rank of incoming links as four columns and loaded initial data into it before first iteration.

4) Written a script to take summation of page ranks of incoming links and eventually calculating the page rank of indices and updated all new page ranks.

5) Repeated the step 4 10 times to get the page ranks after 10 iterations.

6) Written the script to find the top ten page ranks and store it in a text file which is the required paper_rank.txt.

CONCLUSION

With page rank algorithm implementation using map reduce and hive scripts, we could find the top ten papers with highest page rank.

REFERENCES

1) https://hadoop.apache.org/docs/r1.2.1/api/
2) https://cwiki.apache.org/confluence/display/Hive/LanguageManual
3) http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm
4) http://pr.efactory.de/e-pagerank-algorithm.shtml