Author Recommendation Based on Cooperation Relationship

An useful application of page-rank algorithm

YuQi Yi
Centre for Cognitive Machines and Computational Health
CMACH
ShangHai, China
awonderfullife@sjtu.edu.cn
Student number: 515030910596

***Abstract*—This report is reporting a interesting engineering attempt in scholar search field - finding related authors and do some recommendation for it. In this project, we defined some formulas about the influence identification between authors and then we construct a directed graph between them, using page rank algorithm to get the hidden relations in author.**

***Index Terms*—data mining, recommendation, page-rank, scholar searching**

I. Introduction

In this project we did the following contributions. First, we defined a rule that help us collecting the author's co-operation information from papers. Then we proposed some reasonable conclusion based on different author weight and their cooperation sequence. To specify the influence value, we also proposed some formulas that help us to calculate based on the information we collected. Finally, we run the page-rank algorithm on the directed graph that we used to represent influential between authors. And thus we could get the final graph that help us to do some recommendation about related authors.
To illustrate this project in detail, in this report, we would focus on the following four different aspects.

II. Background

A. Authors of paper

All persons designated as authors must meet the criteria for authorship detailed in the following statement: We [or substitute "I"] certify that we have participated substantially in the conception and design of this work and the analysis of the data [when applicable] as well as the writing of the manuscript. We have reviewed the final version of the manuscript, approve it for publication, and take public responsibility for its content. Neither this manuscript nor one with substantially similar content under our authorship has been published or is being considered for publication elsewhere, except as described in an attachment.

Identify applicable funding agency here. If none, delete this.

The co-authors of a paper should be all those persons who have made significant scientific contributions to the work reported and who share responsibility and accountability for the results. Other contributions should be indicated in a footnote or an acknowledgments section. An administrative relationship to the investigation does not of itself qualify a person for co-authorship (but occasionally it may be appropriate to acknowledge major administrative assistance). The author who submits a manuscript for publication accepts the responsibility of having included as co-authors all persons appropriate and none inappropriate. The submitting author should have sent each living co-author a draft copy of the manuscript and have obtained the co-authors assent to co-authorship of it.

B. Application value

Nowadays, in data mining field. If we wanted to retrieve the valuable and useful information we wanted, we always need to find the relationship under different data. For a retrieve or recommend system in paper search. As we know, almost all researches is not finished by a single person and an remarkable idea also might comes from the idea of different people. It is a common phenomenon that different researches have a close-knit cooperation on a single paper. For example a young researcher might publish a paper instruct by his or her mentor, in this situation, it is a common situation that the the mentor has a great influence on the young researcher so the idea of this paper might proposed by mentor and developed by young researcher. Besides this, there are also exist some other situation such as two high weight author have cooperation on single paper or mentor act as the first author, young researchers do some other related contribution on it.

III. Content introduction

In this part we will introduce the details about constructing the whole project in parts. Each parts execute a certain function in this project and also it is relatively individual.

A. Get the Author Information

For a typical paper, we could see it's format in picture 1 (where Mask R-CNN is a very classical paper in computer

Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick

(Submitted on 20 Mar 2017 (v1), last revised 24 Jan 2018 (this version, v3))

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at: this https URL

Comments: open source; appendix on more results
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: **arXiv:1703.06870 [cs.CV]**
(or **arXiv:1703.06870v3 [cs.CV]** for this version)

Fig. 1. A Normal Paper Infomation

vision field). under it's title and above it's abstract, it contains the information about the authors. In most situations, a paper is written by several authors in which it contains one fist author and some other cooperate authors. The way we record author information in paper is stored the author as the node in the graph and the edge is the cooperation relationship between them. Take the Mask-RCNN paper as an example we could get the following information after first step:

$$Nodes : (K, G, P, R)$$

$$Edges : [(G, K), (P, K), (R, K)]$$

in which K stands for $KaimingHe$, G stands for $GeorgiaGkioxari$, P stands for $PiotrDollar$, R stands for $RossGirshick$ and edge (G, K) means the directed edges starts from G point to K.
so after this step, we could get a directed graph without weight about authors. Then in step2 and step3 we try to calculate the weight for each directed edge.

B. Define author influence relationship

To calculate the specific weight value for each edge, first we need to define the influence relationship between the authors for a single paper. Suppose we have get the rank for a single author from database (the higher rank a author is means the author has higher influence in the scholar field, this is always calculate from the paper number he or she publish and the citation number about his or her paper). Normally we think there exist four different situation for the cooperation relationship so we defined their corresponding influence relationship respectively.

- Low rank author - High rank author: in this situation, the normal situation is the high rank author have a great influence on the low rank author. as it might just the situation that the low rank author ask some modify advices from the high rank author, but the main part for paper is done by low rank author.
- High rank author - High rank author: in this situation, the normal situation is that, two author discuss some idea and the first author will reference the co-author's view point, besides the co-author will also reference the first author's idea in the project he does. Because both two author admit the other author and they have correlation, so the influence between them is bidirectional.
- Low rank author - Low rank author: this situation is somehow like the situation above, because two author might be co-workers(they might be the researchers in same institution). so the influence between them is also bidirectional however the degree might not as large as the situation above. (To simplify in this project, we regard the influence degree is same as the situation above)
- High rank - low rank: in this situation, the High rank will have a middle influence on low rank. Normally, the low rank author act as an assistant, he might be just a temp co-researcher. so the influence from high rank to low rank is limited.

We should mention you that on the above item, the first term describe the rank of first author for a paper, the second term describe the co-author of that paper. High rank means the author has a high status on the scholar field he focus on while low rank means the author has a limited influence in his research field.

C. Calculate the specific influence weight

In the above step we have defined the influence relationship between first author and co-author and in this step we define the specific calculate formula about the influence value(which would affect the edge weight of the directed graph we construct on step 1) the following is the formula we calculate:

$$Ref = \frac{L_i - L_0}{|i - 0|}$$

if $Ref \geq C$:

$$E_{weight} = 2 * Ref$$

if $-C < Ref < C$:

$$E_{weight1} = \frac{Ref}{2}$$

$$E_{weight2} = \frac{Ref}{2}$$

if $Ref \leq C$:

$$E_{weight} = Ref$$

in which the i stand the i_{th} co-author while C stands a constant that defined by us which used to control the classify edge of influence relationship.

After we calculate and added all the edge weight, the next task is how to normalized them. Fist we normalied then into range $[0, 3]$ then to smooth it, we used the activation function used in neural networks. we take the $tanh$ as the function we used to smooth value.

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$Fig2$ shows the shape of $tanh$ function. Then we get the final weight of all directed edge.

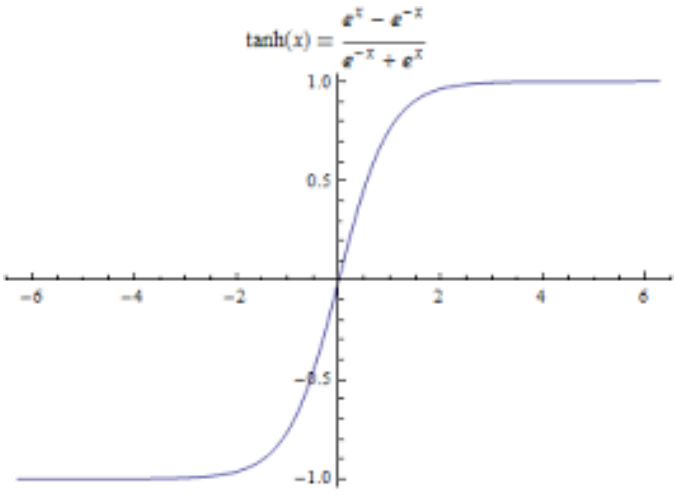

Fig. 2. Graph of tanh function

D. Page Rank

In order to measure the relative importance of web pages, Google propose PageRank algorithm, a method for computing a ranking for every web page based on the graph of the web. PageRank algorithm has a wide applications in search, browsing, and traffic estimation area.

Here we gives a mathematical description of PageRank and some illustrate how to apply PageRank algorithm on how to build the author influence system.

First we compare the backlinks of web page and author relationship, figure 3 shows the backlinks in web page and we could see that, it is the same shape as the directed graph of author relationship we build(in which the direction of backlinks means the influence direction, besides we have another information about the edge weight which provide

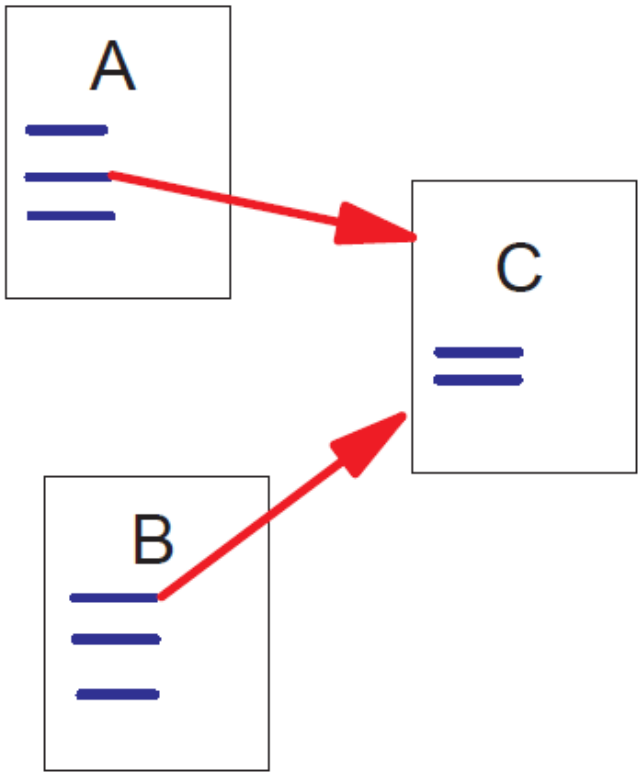

Fig. 3. A and B are Backlinks of C

more information about links) Web pages vary greatly in terms of the number of backlinks they have. For example, the Netscape home page has 62,804 backlinks in our current database compared to most pages which have just a few backlinks. Generally, highly linked pages are more important than pages with few links. Simple citation counting has been used to speculate on the future winners of the Nobel Prize. PageRank provides a more sophisticated method for doing citation counting.

The reason that PageRank is interesting is that there are many cases where simple citation counting does not correspond to our common sense notion of importance. For example, if a web page has a link o the Yahoo home page, it may be just one link but it is a very important one. This page should be ranked higher than many pages with more links but from obscure places. PageRank is an attempt to see how good an approximation to importance" can be obtained just from the link structure.

We think the cooperation between authors is the similar situation as the situation of web backlinks. so it is valuable idea that we use PageRank algorithm in our project. A author has high rank of influence weight if the sum of the ranks of its backlinks is high. This covers both the case when a author has many backlinks and when a page has a few highly ranked backlinks.

1) Definition of PageRank: Let μ be a web page. Then let F_μ be the set of pages μ points to and B_μ be the set of pages

that point to μ. Let $N_\mu = |F_\mu|$ be the number of links from μ and let c be a factor used for normalization (so that the total rank of all web pages is constant).
We begin by defining a simple ranking, R which is a slightly simplified version of PageRank:

$$R(\mu) = c \sum_{\nu \in B_\mu} \frac{R(\nu)}{N_\nu}$$

Note that the rank of a page is divided among its forward links

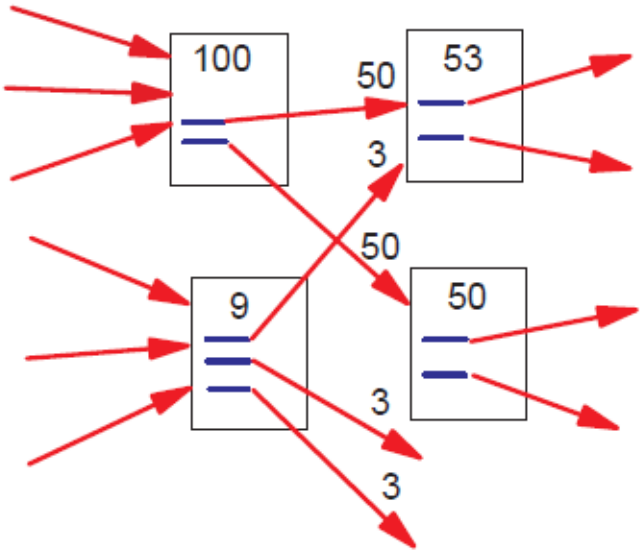

Fig. 4. Simplified PageRank Calculation

evenly to contribute to the ranks of the author they point to. Note that $c < 1$ because there are a number of authors with no forward links and their weight is lost from the system. The equation is recursive but it may be computed by starting with any set of ranks and iterating the computation until it converges. Figure 4 demonstrates the propagation of rank from one pair of pages to another. Figure 5 shows a consistent steady state solution for a set of pages. Stated another way, let A be a square matrix with the rows and column corresponding to web pages. Let

$$A_{\mu,x} = \frac{1}{N_u}$$

if there is an edge from μ to ν and $A_{\mu,\nu} = 0$ if not. If we treat R as a vector over web pages, then we have $R = cAR$. So R is an eigenvector of A with eigenvalue c. In fact, we want the dominant eigenvector of A. It may be computed by repeatedly applying A to any nondegenerate start vector.
There is a small problem with this simplified ranking function. Consider two authors that point to each other but to no other author. And suppose there is some author which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank(since there are no outedges). The loop forms a sort of trap. To overcome this, PageRank algorithm defined a rank source to deal with

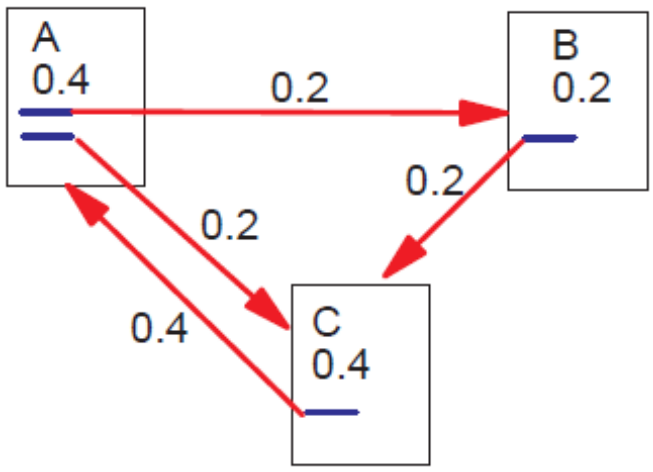

Fig. 5. Simplified PageRank Calculation

this problem, Here we won't discuss it in details.

2) Random Surfer Model: The definition of PageRank above has another intuitive basis in random walks on graphs. The simplified version corresponds to the standing probability distribution of a random walk on the graph of the author relationship. Intuitively, this can be thought of as modeling the behavior of a "random surfer". The "random surfer" simply keeps walking on successive links at random. However, if a real author surfer ever gets into a small loop of authors, it is unlikely that the surfer will continue in the loop forever. Instead, the surfer will jump to some other author. The additional factor E can be viewed as a way of modeling this behavior: the surfer periodically "gets bored" and jumps to a random author chosen based on the distribution in E. (where E as a user defined parameter).

3) Computing PageRank: The computation of PageRank is fairly straightforward if we ignore the issues of scale. Let S be almost any vector over Web pages (for example E). Then PageRank may be computed as follows:

Algorithm Start:

$R_0 \leftarrow S$
repeat
  $R_{i+1} \leftarrow AR_i$
  $d \leftarrow ||R_i||_1 - ||R_{i+1}||_1$
  $R_{i+1} \leftarrow R_{i+1} + dE$
  $\delta \leftarrow ||R_{i+1} - R_i||$
until $\delta < \varepsilon$

Algorithm 1: Compute-PageRank

Note that the d factor increases the rate of convergence and maintains $||R||_1$. An alternative normalization is to multiply R by the appropriate factor. The use of d may have a small impact on the in influence of E.

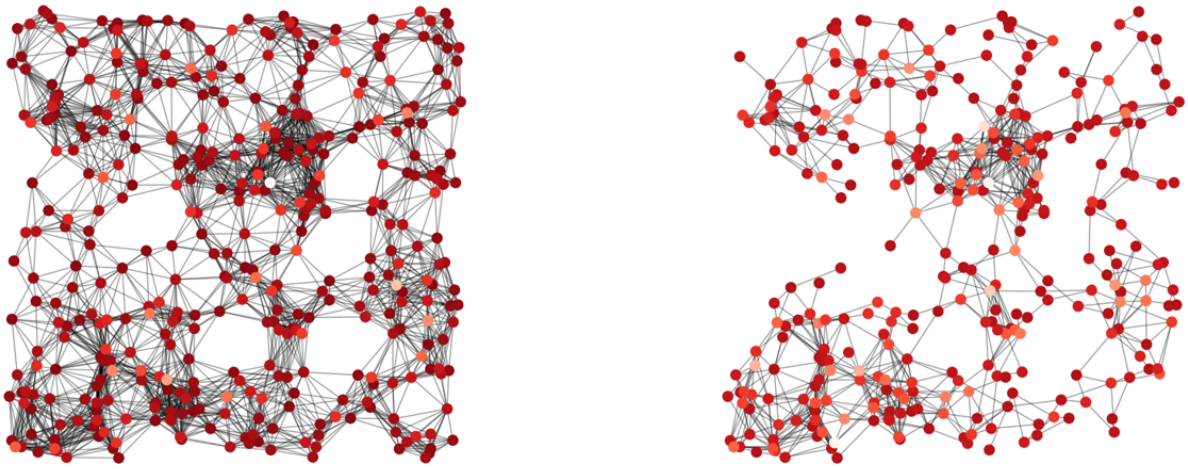

Fig. 6. Author Influential graph

To simplify the representation, we conclude the PageRank algorithm with following formula representation:

$$PageRank(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PageRank(p_j)}{L(p_j)}$$

in which p stands for a author, p_i stands for the influence rank of author i, N is the number of total authors. $L(p)$ stands for the edge that start from node p, d is the reciprocal of weight. we should mention that for traditional PageRank algorithm, d is an constant value because we backlinks doesn't contains the information about link's weight. so it's a big change in our project, which would leading to better results.
The final result of running PageRank algorithm on the directed graph we get in the above steps is that we get an new influence graph, which shows the author influence on other related authors and this graph will help us to do the author recommendation.

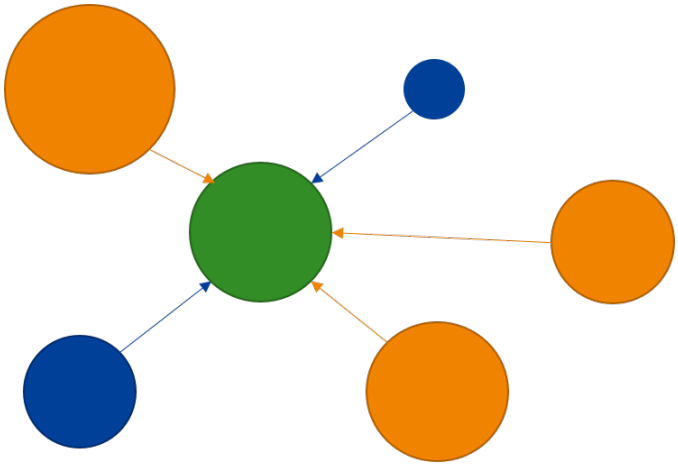

Fig. 7. Author Recommendation in Graph

E. Author Recommendation

In this step, we use the information we get in the formal step to achieve the goal of author recommendation. Here we show how the author recommendation system works. When the user search for a specific authorwe find the author location on the directed graph we get. Then we could find the related authors who have the backlinks to the author we find. As the directed graph about author influence have the information about the influential weight for single author, so we could fliter out the author with low influential weight and only return the recommend authors whose influential weight greater or equal than the threshold influential weight value we set. Figure 7 shows the operation we done in this step.

- Green Circle means the author user searched
- Blue Circle means the author have backlinks point to searched author but with low influential weight
- Origin Circle means the author have backlinks point to searched author and with high influential weigh

In the final recommendation system, we return green author as well as the origin author acts for recommendation authors.

IV. Result Display

As a single person project, due to the limit of time and lack of database information. In this project I generate several node and edges to act as author and the cooperation ship between them. Also I generate the initial rank weight for each author. So the data are similar to the real data collected from paper(by our method defined above)
we(though I did all this project by myself, but I prefer say we instead of I) write the PageRank algorithm in Python and run it on laptop, finally we get the influential weight directed graph showed in Figure 6. (Fig 6 contains two result which corresponding to two different dataset we generate, for two different data set we mantain their author to be same but modified the cooperation relationship between then) In figure 6 the node with shallow color means the author with low influential rate and the node with deep color means the node

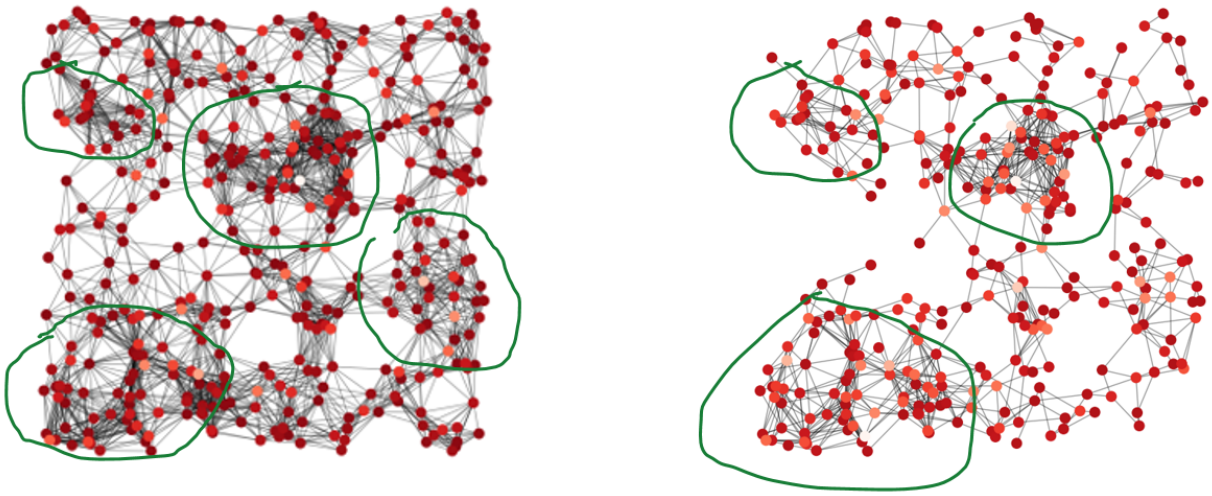

Fig. 8. Author Influential graph

with high influential rate. And the edge connected between two node means the two author have cooperation relationship which means they could influence each other.
In the above above section we get the author influential graph for author recommendation, however this graph contain more information and these information could used in some other applications(instead just used for author recommendation). For example, we could use the graph find the center fields for a research field, a center field for research could have many researchers and they would have many cooperation between each other, when we visulized the author influential graph, it becames easy for us to find such area. Figure 8 showed the area we marked out based on the two different author influential graph. Finding center field would help us to have a more specific idea about the research in that field.

A. *Future development*

Until now, we have introduce our method for author recommendation. However, there still some aspects that needed modify in future, Here we introduce three proposal later research or develop direction.

- More accurate formulation: there must exist some better ways to calculate the weight for the edge. Besides we think we could also do some following operation on final direct graph to get more information about the authors.
- More area development: in this project we just focus on author relationship, but the citations for paper are also an important information for data mining. Though the situation might be different but our method also have certain reference value.
- More big data: In this project, page rank algorithm is run on a small number of data on an Intel I5 CPU core. However if we wanted to apply this algorithm on huge data, we need use the Hadoop method to help us.

Here we introduce the Hadoop as it will plays the key role if we wanted to apply this project work to the real scholar search engine.
The genesis of Hadoop came from the Google File System paper that was published in October 2003. This paper spawned another research paper from Google MapReduce: Simplified Data Processing on Large Clusters. Hadoop is an open source software platform for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Hadoop services provide for data storage, data processing, data access, data governance, security, and operations. Some of the reasons organizations use Hadoop is its ability to store, manage and analyze vast amounts of structured and unstructured data quickly, reliably, flexibly and at low-cost. **Benifits:**

- Scalability and Performance distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyze data at petabyte scale.
- Reliability large computing clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resilient when a node fails processing is redirected to the remaining nodes in the cluster and data is automatically re-replicated in preparation for future node failures.
- Flexibility unlike traditional relational database management systems, you dont have to created structured schemas before storing data. You can store data in any format, including semi-structured or unstructured formats, and then parse and apply schema to the data when read.

So it always a good choice method to realize algorithm if we need to handle huge data in real work system. (However not as easy as use Python on your own laptop)

V. Acknowledgment

Here I would show my greast respect to our teacher Xing-Bing Wang and our teacher assistant. In the whole classes.

Teacher gives us various of resources, invite some standful alumni to give speech and also let us access the front research of scholar search engine which inspired me a lot. Besides, the whole course contains various of knowledge that are really hot nowadays. I think I won't get such idea of author recommendation research without the knowledge I received from this course.
I would also like to thanks for my laboratory classmates, they really give me some good ideas on how to construct and implement the whole project.

REFERENCES

[1] Technical paper recommendation: A study in combining multiple information sources; C. Basu H. Hirsh, W. W. Cohen, C. Nevill-Manning; Journal artificial intelligence research; May 1, 2001
[2] The PageRank Citation Ranking: Bringing Order to the Web 1892; L Page, S Brin, R Motwani, T Winograd; ilpubs.stanford.edu; 1999
[3] The pagerank citation ranking: Bringing order to the web; L Page, S Brin, R Motwani, T Winograd; pdfs.semanticscholar.org; 1998
[4] The hadoop distributed file system; K Shvachko, H Kuang, S Radia - Mass storage systems , ; Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on
[5] The simultaneous evolution of author and paper networks; Katy Brner, Jeegar T. Maru, and Robert L. Goldstone; PNAS April 6, 2004. 101
[6] Combination of feature engineering and ranking models for paper-author identification in KDD Cup 2013; Chun-Liang Li, Yu-Chuan Su, Ting-Wei Lin, Cheng-Hao Tsai, Wei-Cheng Chang
[7] The weakening relationship between the impact factor and papers' citations in the digital age; George A. Lozano Vincent Larivire Yves Gingras; 2012 - Wiley Online Library