Final Report: Ranking of Academic Papers

Dec 2018

1 BACKGROUND

Paper publication is one of the most important factors in measuring scholars' academic ability. The way to set up the ranking metric could affect the evaluation of a researcher. The goal of this project is to construct a ranking metric to evaluate academic papers and researchers. Although we have some influential and widely used metrics such as h-index, which means that a scholar with an index of h has published h papers each of which has been cited by other papers at least h times[1]. H-index has its own limitations and drawbacks. In addition, there are also several improved ranking metrics based on h-index. M-index is defined as h/n, where n is the number of years since the first published paper of the scientist[1]. Moreover, G-index is an alternative for h-index. It averages the numbers of citations, which means it allows citations from higher-cited papers to be used to bolster lower-cited papers in meeting a threshold[2]. In this project, we aim to build an all-encompassing personalized metric for both academic papers and researchers. Our inspirations are based on the limitation of h-index and other metrics.

2 DATASET

The data we collected is provided by Microsoft Academic Graph(MAG) including 9 datasets with 166,192,182 papers and with the size of 104G in total. The datasets contains different fields such as computer science, physics, social science, history, chemistry, biology etc. The earliest paper is from 1800 and the latest paper is from 2018. Each file is in text format which stores all the information of papers in JSON objects. Each record contains id, $title$, $authors.name$, $author.org$, $venue$, $year$, $keywords$, fos, $n\_citation$, $references$, $page\_stat$, $page\_end$, doc_type, $lang$, $publisher$, $volume$, $issue$, $isbn$, doi, pdf, url, $abstract$ as features. One feature to be noted is that $n\_citation$ is the number of citation that was referred to this paper instead of the number of references of this paper. There are many features that are not helpful for our project, such as $page\_stat$, $page\_end$, and as the datasets are so big, we discarded some features in the preprocessing step. We analyzed all the information based on these datasets.

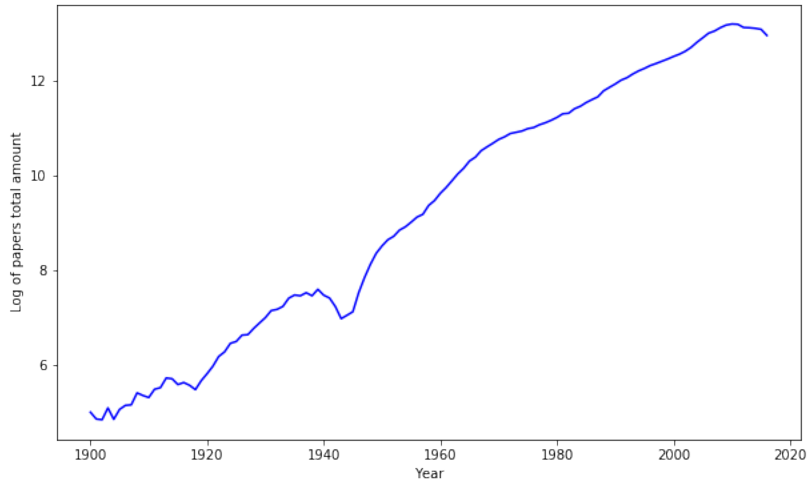


Figure 2.1: The logarithm amount of papers in Computer Science and Physics (1900-2017)

3 Methodology

3.1 Introduction

To realize our metric, we created a new author-indexed dataset based on the original dataset and each record in it contained an author's name, id of papers that the author participated in and the orders of the author in these papers. This dataset clearly represents how much work each author has done so far. This dataset is also used for identifying the top 100 ranked researchers and comparing our metrics with h-index respectively.

We also built a citation network graph, which explicitly shows the citation relationship among papers. We removed some self-citations whose purposes are just to increase the citations' numbers of itself. Thus, we reduced self-citation impact on the score. Then, we applied the personalized PageRank algorithm on the citation network graph to obtain a score for each paper. By combining the score with the author-indexed dataset, we were able to identify the top 100 ranked authors.

In order to do the time analysis, we extracted the number of citation for each paper along the year to see the trend of $n\_citation$. Moreover, after getting the 49,773 keywords from fos attribute, we built a table that contains $keyword$ and the frequency of this keyword in each year. This could tell us which keyword is popular in specific year which is also regarded as a factor for our PageRank algorithm. Also, we analyzed PageRank in each year to see how the time has influence on the PageRank.

3.2 PREPROCESSING

In this part, we aimed to clean the data and obtained necessary information which was not provided by MAG dataset. Three steps were performed in preprocessing part: author names filtering, keywords extraction and paper field classification. Moreover, we mainly focused on papers written in English since papers in other languages would introduce lots of synonyms among different languages which were hard to deal with.

In normal case, person name written in English always in the following format: (First name) (Last name) or (First name) (Middle name) (Last name) where middle name and first name were sometimes represented by its capital letter. However, we realized that there are some invalid names, like some special characters before the real name parts. We dealt with those names to guarantee that the names were in the correct format. Besides, there were several articles whose authors contained non-English characters. We changed this kind of author names to proper English form by referring web pages from Google Scholar and their personal websites. Also, we took name repetition into consideration. For example, we had Benjamin J. Arnold, Benjamin N. Arnold, and Benjamin Arnold. We would like to know if the one without the middle name is Benjamin J. Arnold or Benjamin N. Arnold or a totally different person. Unfortunately, we thought it would take a long time to determine matching relations among authors, so we decided to leave those as they were. Some types of invalid English author names and corresponding correct ones were listed in the following table 3.1.

Invalid name format	**Format that should be**
-Jr. Paulo Drews	Paulo Drews
§ and Aaron R. Dinner	Aaron R. Dinner
和彰 村上	Kazuaki Murakami

Table 3.1: Special Examples of Names

The fos feature in the dataset is the abbreviation of "field of study", which indicates related keywords of articles. However, there were about one third of articles which is lack of fos field. Thus, we needed to extract keywords from their titles and abstracts. In this part, we regarded title and abstract of one paper as several independent sentences and applied rake-nltk algorithm to obtain keywords list where related keywords were with high rank scores[3].

The MAG dataset contains a variety of subjects containing computer science, biology, etc. In our project, we focused on the articles in computer sciences and physics, the amount of which were 484,8237(2.7%) and 691,5527(3.9%) respectively. We had to summarize the subject of each article since this information was not provided in original dataset directly. To achieve this goal, we built a relation graph where nodes were articles, edges meant there are shared keywords between two articles and weights for edge showed how strong the relation was. In other words, if two articles shared more keywords, the edge between them would get a higher weight. Then, we used this relation graph to search articles which has relatively strong relations with keywords computer science and physics respectively. If selected articles did not belong to other main subjects such as biology, chemistry, etc, the subject of these articles could be specified. All keywords related to computer science and physics, also the main academic subjects are obtained from Wikipedia[4][5][6].

3.3 RankScore Metric

3.3.1 PageRank Algorithm as Base

PageRank is an algorithm used by Google Search to rank websites in their search engine results. It is a way of measuring the importance of website pages. 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[7].

The simplified PageRank algorithm is defined as follow: Suppose there are four web pages A, B, C, D as shown in the figure 3.1. The page B had a link to pages A and C, page C had a link to page A, and page D had links to all three pages. First, we give initial value for each page, which is $\frac{1}{number of pages}$. In this example, the initial value for each page would be 0.25. In the first iteration, page B would transfer half of its existing value to page A, and transfer the other half to page C. Page C would transfer all of its existing value to page A, which is the only page links to C. Since D had links to all three pages, it would transfer one third of its existing value to A, and one third to B, and one third to C. Thus, the formula for $PR(A)$ would be $PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$. Same for $PR(B)$, $PR(C)$ and $PR(D)$.

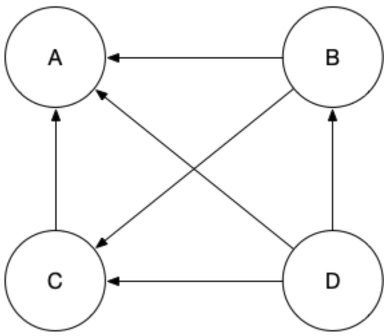


Figure 3.1: Author Citation Graph Sample

The basic formula for our ranking algorithm is:

$$RS(A) = (1-d) + d * (\frac{RS(T_1)}{C(T_1)} + \frac{RS(T_2)}{C(T_2)} + ... + \frac{RS(T_n)}{C(T_n)})$$

where:

- A: a paper in a citation graph
- $RS(A)$: the ranking score of A
- $T_1 \dots T_n$: papers that cited A

- $C(T_i)$: the total number of references of T_i
- d: considered to stop the other papers having too much influence, this total vote is "damped down" by multiplying it by some fraction (usually 0.85)
- $(1-d)$: if a paper has no citation, it will still get a small score of 1-d which can balance the scores.

Although PageRank is a great and widely used algorithm, it has some limitations. First, the PageRank algorithm does not reflect current popularity, which is an important aspect of search to user. People likely to search the most recent information. Then, older pages may have higher link than the newer ones, even if a new page have some good contents but it may not have many links in the early state.

In terms of our project, we improved some disadvantages of the original PageRank algorithm. First, some subjects are very popular during a period of time, for example, nowadays, Artificial Intelligence is considered as a very hot word, so the paper related to Artificial Intelligence may be cited more times than other subjects that are not popular currently. This popular limitation is what we wanted to improve in our metric. In addition, for the drawback that PageRank favors old pages, we did keyword analysis. By analyzing the keywords, we could know which keyword is popular at which time period, and we adjusted our formula based on this analysis. Moreover, we also took the order of an author into consideration. A paper usually has more than one authors and the order of each author reflect the author's contribution, i.e. each author did not make contribution equally. So, we gave authors score based on their order in a paper. Also, the total number of publication is a factor as well. Suppose an author has many publications, but just a few of them has been cited, which means his papers do not have high quality overall. We penalized this kind of situation.

The following parts present how we improved the original pagerank algorithm in three ways to finally build our own ranking metric model.

3.3.2 Topic Popularity (Time Analysis) in RankScore

To tell whether a paper's popularity is relevant to the topic of it covers, we introduced the topic popularity coefficient into the time analysis part. Admittedly, a paper probably becomes too popular because it meets the current technology interest. The following content shows our basic thoughts and method to smooth the influence of topic popularity.

A research topic commonly derived from some discovers or other researches. During the lifetime of one topic, there are some early research papers which can be seen as milestones and basis of later related researches. These papers should be given higher scores compared to the later works in this field. Figure 3.2 shows the number of papers within the four topics. We can give the summary from the figure that the basis of one topic usually appears at its early stage and followed by a highly popular period. These excellent articles and their authors were worth a high rank score.

Conversely, some papers in popular period only introduced small improvement to a problem or performed technical summary which may receive relative high citation count.

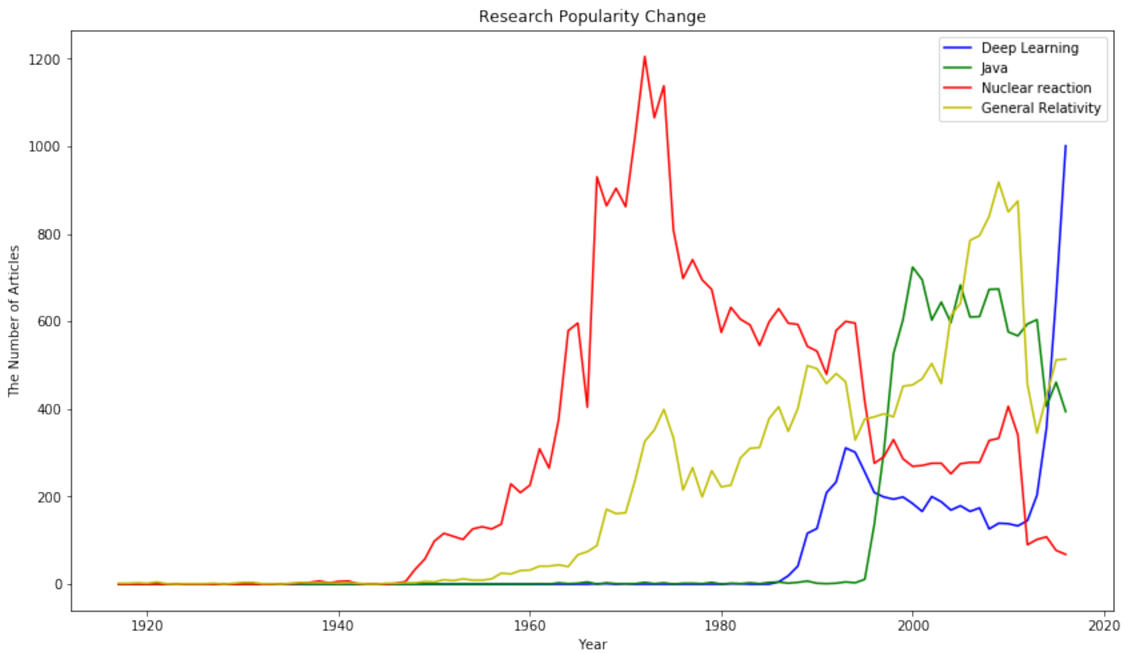


Figure 3.2: Popularity Change among Topics

This situation, in our opinion, should be given a low rank score. Thus, we used a coefficient related to the keyword popularity.

$$Coef_i = \frac{\sum_{w \in KeywordList_{article}} count_{year}(w) * \log_{10} keywordcount_{year}}{2 * keywordcount_{year}} \tag{3.1}$$

$$PopCoef_i = 2^{Coef_i} \tag{3.2}$$

where $KeywordList_{article}$ recorded all keywords from the list of field of study provided by the dataset and/or keyword extraction in $article$, w is a specific keyword in $article$, $count_{year}(w)$ is the number of keyword w appearing in $year$, $keywordcount_{year}$ represented the total amount of appearing keywords in $year$ which can be considered as the academic activity in this year. Also, $year$ was decided by $article$. Normally, this popularity coefficient is in the range $[1, 2]$. The value near 1.0 means that the article is not influenced by popularities of related topics significantly, while the value near 2.0 means the article received an excessive score than its true value. In common cases, the paper regarded as basis of a topic would have a coefficient really near to 1.0.

3.3.3 Total Number of Publications in RankScore

Traditional PageRank algorithm regarded each page as an independent and equal node, however, this is not reasonable among all authors. In some ranking metrics, the total amount of published papers would in a relatively high position, which lets researchers focus on the papers amount instead of papers quality. Thus, we would like to introduce a personalized coefficient which describes the situation of an author's research. In this part, the total number of publications was used to represent the an author's research "activity". This kind of activity as personalized coefficient was defined as the formula 3.3.

$$PersCoef_{author} = \frac{\sum[a \in ArticleList_{author}]}{active\_year_{author}} \tag{3.3}$$

$$active\_year_{author} = \min(current\_year, last\_active\_year) - first\_publication\_year \tag{3.4}$$

In formula 3.3 and 3.4,

- $PersCoef_{author}$ is the personalized coefficient of $author$;
- $author$ is the author name in authors list;
- $active\_year_{author}$ represents the academic active period of $author$ which is the difference of the year of author's first publication and the current year or the year if the author does not publish any articles in continuous three years;
- a is one of an article in $ArticleList_{author}$;

This kind of personalized coefficient would give penalty to articles the author of which publishes a lot of papers with low quality.

3.3.4 RankScore Metric with Improvements

The total rank score we used is described in the formula 3.5.

$$RS_a = \frac{1-d}{\sum_{au \in AL_a} PersCoef_{au} * 2^{-order} * C(papers) * PopCoef_a} + d * \sum_{i=1}^{n} \frac{RS(T_i)}{C(T_i)} \tag{3.5}$$

- a: a paper in the citation graph
- RS_a: the ranking score of a
- AL_a: authors' list of paper a
- au: an author for paper a
- $PersCoef_{au}$: personalized coefficient of author au
- $order$: the order of author au, starting from zero
- $C(papers)$: the total number of papers in citation graph, this term is used to reduce the minimal of rank score
- $PopCoef_{au}$: popularity coefficient of paper a
- $T_1 \ldots T_n$: papers that cited a

- $C(T_i)$: the total number of references of T_i
- d: considered to stop the other papers having too much influence, this total vote is "damped down" by multiplying it by some fraction, here 0.55 was used

Our raw results get directly from formula 3.5 are a list of small decimals. Then we perform standard normalization to these results, which lets us compare authors' scores among different disciplines.

3.4 Author Score Based on RankScore

The baseline model here was the summation of rank score among all articles belonged to an author. From the side of an article, there were usually more than one author listed in the authors' list. These authors may not contribute to the paper and project with equally and we noticed that first two authors usually were responsible for paper writing and core research parts while other authors played the role of assistant. Thus, we decided to apply a coefficient that measured author's contribution to the article. For each author, we calculated the author score by rank score of all papers, together with a contribution ratio respectively. The formula we used is:

$$AS_i = \sum_{p \in P_i} 2^{-order} * RS(p) \tag{3.6}$$

where AS_i represents the score of author i, P_i is the list of papers written by author i, 2^{-order} gives author reasonable weight based on the order of author. For example, if this author is the first order, its order value is 0, and the weight of this author is $2^{-0} = 1$. The order value is 2 if the author is at the third position in the author list and the weight of this author is $2^{-2} = 0.25$. Obviously, the first author contributed more than the third author. This is how the order of author affect the weight. Thus, it is reasonable to get the author score by ranking score among one's papers and the corresponding weights.

3.5 Prediction for Paper Popularity

There are several ways for measuring popularity of a paper, for example the prize it wins, the citation amount it acquires, and the number of its downloads on the internet, etc. We selected the second one also the most intuitive one to represent the popularity of a paper.

Based on the ranking metric we built above, we decided to include three kinds of parameters into our model to predict the citation number of a paper in future years: citation count (C), mean of keyword popularity(kw_mean), summation of rank scores of the paper's authors(AS_sum). For some year y and a paper P, the citation count of P is how many times the paper was (or will be) cited in year y; mean of keyword popularity is the average frequency of all of the keywords in P appear in the whole dataset of year y; summation of rank scores of the paper's authors calculates the sum of ranking score for all authors of P in y.

We decided to use data in three consecutive years to predict the following next year, e.g., in order to predict the citation number of some paper P in 2017, we would extract C, kw_mean and AS_sum respectively in year 2014, 2015 and 2016.

Figure 3.3 shows the heatmap of correlations among these features. It is obviously that citation counts among years are highly correlated. Therefore, we chose linear regression model to predict citation counts for papers in future years.

We took the whole dataset as the training dataset to bulid up our linear regression model. Set of X included all features in figure 3.3 other than year 2017 and y was set to be the citation number in 2017. Then we randomly extracted 10k records from the training dataset and performed 5-fold cross-test on it. The results and analysis will be explicitly shown in the next section.

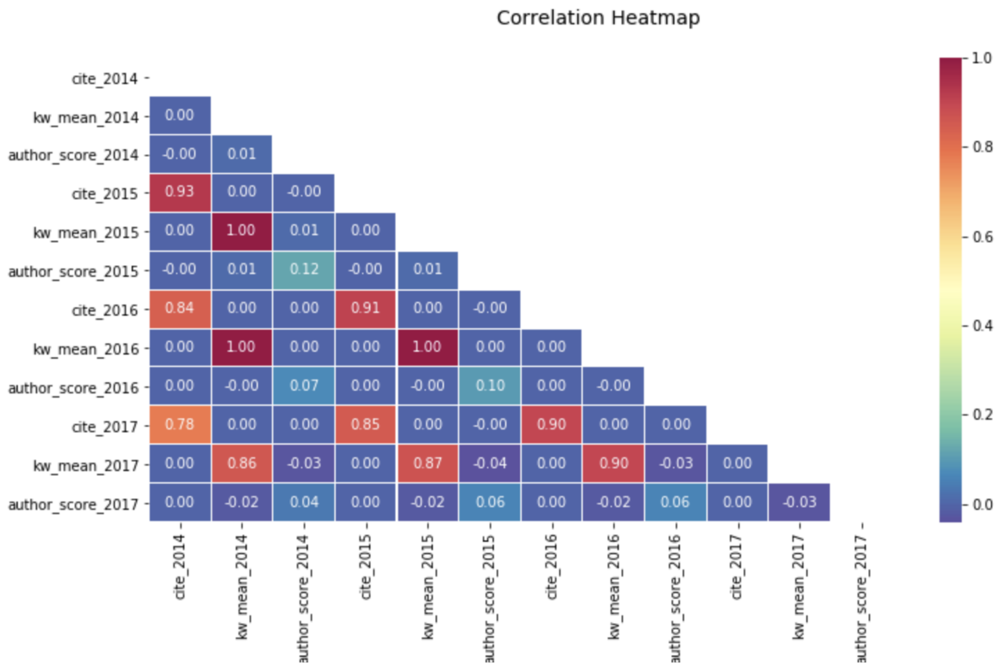


Figure 3.3: Heatmap

4 Result and Validation

Our source codes and some of results are available from GitHub: https://github.com/lchloride/paper_review. All raw and preprocessed data are not uploaded.

4.1 RankScore Metric Results

We performed our rank score metric among all authors in computer science and physics and normalized results with standard normal distribution. The distribution of authors' scores in computer science and physics are shown in figure 4.1. The shape of distributions among computer science and physics are really similar which indicates most of authors have low rank scores.

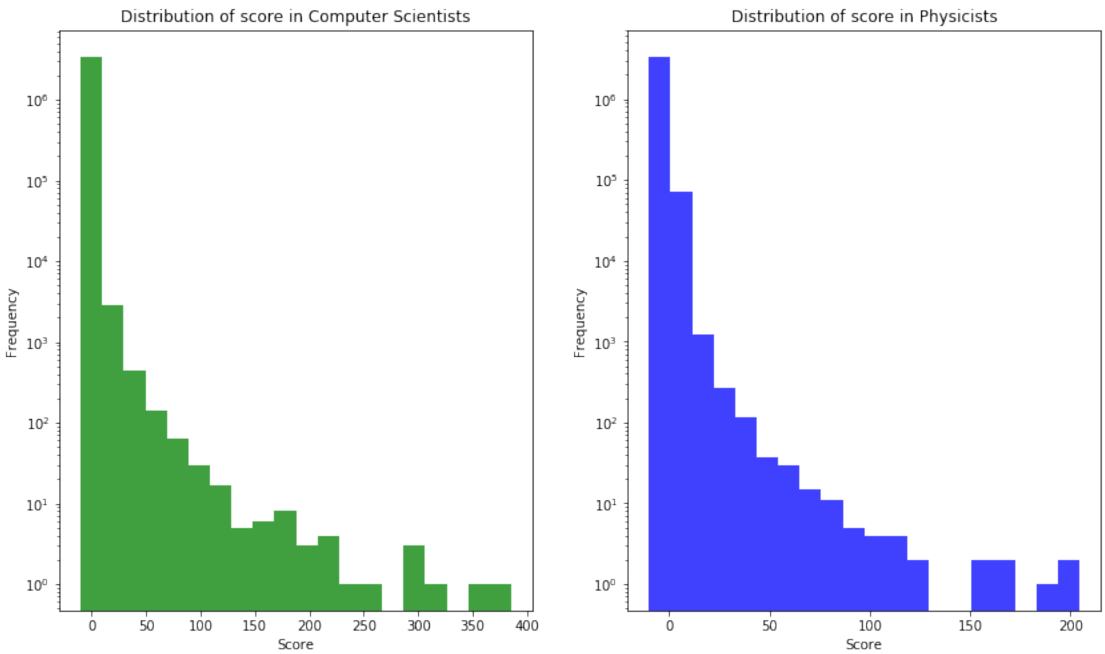


Figure 4.1: Distribution of authors' scores

Besides, we performed Pearson correlation tests in order to show the relations among rank score, citation numbers and year. The Pearson correlation coefficients are listed in the table 4.1. This result illustrates that our ranking score somehow has a relations with the number of citations for a paper. Moreover, year is not the decided factor to our ranking metric.

Correlation Coefficient	Rank Score
n_citation	0.591164
year	-0.056489

Table 4.1: Pearson correlation coefficient among rank score, citation number and year

4.2 Comparing with h-index

We summarized top 100 authors in computer science and physics by using our metric and also top 100 authors by using h-index. We collected the data from some famous and important rewards in both computer science field and physics field. In computer science domain, we have Turing Award, IEEE Medal of Honor, and Marconi Prize(in field of communication). In physics domain, we have the most famous reward Nobel Prize and Dirac Medal. Then, we wanted to check how many authors in the top 100 has won the influential rewards. we evaluated our metric by analyzing the coverage of our top 100 authors in the significant rewards versus the coverage of h-index top 100 authors in the rewards.

The results are shown in the table 4.2 and table 4.3.

CS	Turing Award	IEEE Medal of Honor	Marconi Prize
Our metric	21/100	2/100	5/100
h-index	2/100	0/100	0/100

Table 4.2: Awards Distribution in Computer Science

Physics	Nobel Prize	Dirac Medal
Our metric	5/100	0/100
h-index	0/100	0/100

Table 4.3: Awards Distribution in Physics

From table 4.2 and 4.3, we can see that our metric has higher coverage than the h-index metric, especially in Turing Award. In our metric, 21 of the top 100 authors are the recipients of Turing Award, whereas only 2 of the top 100 authors have won the Turing Award. Because Turing Award is recognized as the highest distinction in computer science, to some extent, our metric is more accurate then the h-index metric.

When we compared the top 100 authors in our result and h-index result, we noticed that h-index somehow underestimates some well-known scientists. Harry Nyquist who was known for sampling theorem published just six papers in his life with h-index of six, however, he should receive a high reputation since his work was the foundation of modern signal theorem. Our ranking metric gave Harry Nyguist a really high score which lets him at the fourth position among all computer scientists. Albert Einstein is another case. The amount of his works was fewer than current scientists but most of his papers have received more than one thousand citations.

4.3 Prediction Results and Analysis

As we mentioned in Section 3.5, we set up a linear regression model to predict citation counts for papers in some future year. For validation, we performed a 5-fold cross-test for the test dataset and compared the prediction results to the actual citation number in 2017. Figure 4.2 shows the statistics of the mean square error between predication and actual values of citation counts in 2017.

The prediction results are pretty ideal: according to the table above, most of the predictions (more than 75%) are very close to the actual citation counts of year 2017; Figure 4.2(b) tells us that MSE mostly distributes within 20 indicating that predictions with error less than 5 predominates the test dataset.

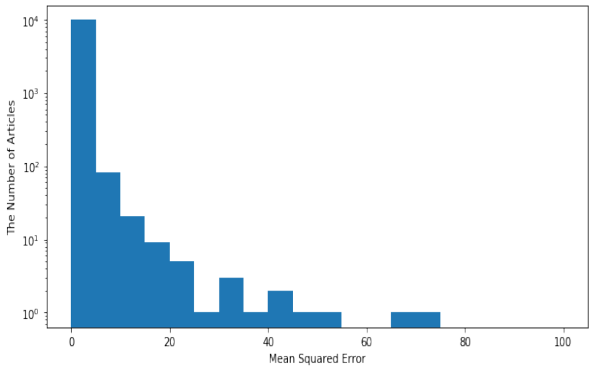
5 Future Work

There are still some problems and improvements remaining to be done in our work so far:

- In our baseline model for ranking metric, we found that many Chinese researchers had very high cumulative rank score due to the huge amount of papers they wrote, e.g.

Count	10000
mean	0.426080
std	8.088412
min	0.000003
25%	0.000948
50%	0.000958
75%	0.050580
max	679.515892

(a)



(b)

Figure 4.2: (a) MSE Statistics (b) MSE Distribution

"Wang Wei". That's obviously because we did not handle the case that different people with the same name, which is hard to separate however. Some solutions we could think of: identify them by venues or institutions, active period of releasing papers, co-authors, and field of study they focus on, etc.

- Since our preprocessing method is not delicate enough, some useful data are probably filtered out and hence the citation graph would not intact. As a result, some isolated points in the graph may negatively affect our ranking metric.

- For Physics, it is common that number of co-authors for a paper is very large (over 50). In our ranking metric, we gave the authors weight of 2^{-order} based on the authors' order in the list which is not fair in the above situation (30 out of 50 may made equally contribution) and the weight would be too small to calculate. Means of distributing weight of author in this case need improving.

- We tried to find various ways to validate our ranking metric and prediction model, but still not convincing enough for our perspective. Some more mathematical test and more methods of evaluation are needed in the future.

6 Reference

1. H-index. (2018, September 12). Retrieved from https://en.wikipedia.org/wiki/H-index

2. G-index. (2018, April 26). Retrieved from https://en.wikipedia.org/wiki/G-index

3. Rake-Nltk. (n.d.). Retrieved from: https://csurfer.github.io/rake-nltk/_build/html/index.html

4. List of Advanced Level subjects. (2018, November 23).
   Retrieved from https://en.wikipedia.org/wiki/List_of_Advanced_Level_subjects

5. Outline of computer science. (2018, November 24).
Retrieved from https://en.wikipedia.org/wiki/Outline_of_computer_science

6. Glossary of physics. (2018, Novermber 19).
Retrieved from :https://en.wikipedia.org/wiki/Glossary_of_physics

7. PageRank.(2018, December 04). Retrieved from: https://en.wikipedia.org/wiki/PageRank

7 APPENDIX

The following tables 7.1-7.3 show the top 100 researchers in computer science and physics.

order	Computer Science	Physics
1	Claude E. Shannon	M. H. Kalos
2	Azriel Rosenfeld	Brian P. Flannery
3	Ronald L. Rivest	John G. Kirkwood
4	H. Nyquist	Horatio Scott Carslaw
5	Alfred V. Aho	G. K. Batchelor
6	Lawrence R. Rabiner	Max Born
7	Gerard Salton	L.D. Landau
8	C. A. R. Hoare	Edward Witten
9	Rakesh Agrawal	Richard Bellman
10	Donald E. Knuth	Emmett N. Leith
11	Leonard Kleinrock	Saul A. Teukolsky
12	Whitfield Diffie	Lieut -Colonel George O Squier
13	Lotfi A. Zadeh	S.M. Sze
14	David E. Goldberg	R. D. Richtmyer
15	Adi Shamir	W. Shockley
16	Leslie Lamport	H. Kogelnik
17	E. F. Codd	J. C. Jaeger
18	Bernard Widrow	S. O. Rice
19	Anil K. Jain	Charles Kittel
20	Teuvo Kohonen	Richard P. Feynman
21	John E. Hopcroft	S. Chandrasekhar
22	David Marr	Stephen W. Hawking
23	John McCarthy	John A. Pople
24	M. R. Garey	John R. Carson
25	George A. Miller	Peter D. Lax
26	Charles E. Perkins	Benoit B. Mandelbrot
27	Edsger W. Dijkstra	Paula A. Whitlock
28	Ian T. Foster	Steven Weinberg
29	Jeffrey D. Ullman	Linus Pauling
30	Allen Newell	Zvi Hashin
31	Richard O. Duda	William W. Wood
32	Rodney A. Brooks	G. Herzberg
33	Geoffrey E. Hinton	Robert S. Mulliken
34	Henning Schulzrinne	Amnon Yariv
35	Lalit R. Bahl	Joseph B. Keller

Table 7.1: Top 35 researchers in Computer Science and Physics

order	Computer Science	Physics
36	Judea Pearl	R. Courant
37	Robert M. Haralick	N. F. Mott
38	Michael Stonebraker	N. Bloembergen
39	Ben Shneiderman	J. C. Slater
40	Jim Gray	J. H. Van Vleck
41	Alex Pentland	E. Bright Wilson
42	Sally Floyd	F.H. Harlow
43	Andrew P. Witkin	John W. Cahn
44	Jakob Nielsen	P. B. Hirsch
45	J. Postel	P. A. M. Dirac
46	Frederick Jelinek	P. K. Tien
47	Berthold K. P. Horn	Albert Einstein
48	Martin E. Hellman	S.B. Cohn
49	Noam Chomsky	Berni J. Alder
50	David S. Johnson	G. B. Whitham
51	Daniel G. Bobrow	G. D. Boyd
52	Stuart K. Card	Sydney Chapman
53	John Ross Quinlan	John D. Ferry
54	Jiawei Han	Hans A. Bethe
55	Nils J. Nilsson	M. S. Longuet-Higgins
56	Peter E. Hart	E. M. Sparrow
57	Vern Paxson	M. A. Biot
58	David A. Patterson	Henry Eyring
59	Andrew J. Viterbi	A. Ashkin
60	Edmund M. Clarke	M. J. Lighthill
61	Bela Julesz	Emil Wolf
62	David G. Lowe	Clarence Zener
63	Erich Gamma	David Turnbull
64	Richard P. Lippmann	E. U. Condon
65	G.D. Forney	N. Metropolis
66	Claude Berrou	P. J. E. Peebles
67	Stephen Grossberg	A. E. H. Love
68	Butler W. Lampson	Irving Langmuir
69	Stephen Deering	E. Wolf
70	Mark Weiser	Charles H. Bennett

Table 7.2: Top 36-70 researchers in Computer Science and Physics

order	Computer Science	Physics
71	Demetri Terzopoulos	Joseph O. Hirschfelder
72	Peter J. Denning	Sara M. McMurry
73	Deborah Estrin	John R. Pierce
74	Peter F. Brown	E. Clementi
75	Richard S. Sutton	Eugene P. Wigner
76	Takeo Kanade	E. R. G. Eckert
77	Fouad A. Tobagi	B. E. Warren
78	Tomaso Poggio	John Bardeen
79	John K. Ousterhout	Geoffrey Taylor
80	Olivier D. Faugeras	J. P. Gordon
81	Richard M. Karp	Michael A. Nielsen
82	Van Jacobson	Louis J. Farrugia
83	Jon Louis Bentleyn	R. Ulrich
84	John L. Hennessy	B. E. Launder
85	David Chaum	James R. Wait
86	Ryszard S. Michalski	Juris Upatnieks
87	Stuart Geman	Julian Schwinger
88	Tim Berners-Lee	R. H. Stolen
89	Niklaus Wirth	J. VonNeumann
90	David Harel	Stephen Wolfram
91	Barbara Liskov	F. P. Bowden
92	Oded Goldreich	J. J. Hopfield
93	Hector Garcia-Molina	C. Truesdell
94	David A. Huffman	John W. Miles
95	David R. Cheriton	A. H. Cottrell
96	Kenneth Ward Church	J. G. Bednorz
97	Dorothy E. Denning	Simon Ramo
98	Yoav Freund	James R. Rice
99	Stephen A. Cook	Ilya Prigogine
100	Richard Fikes	J F Nye

Table 7.3: Top 71-100 researchers in Computer Science and Physics