PageRank-based Link Analysis on Books

Linh Chi HOANG
Data Science for Economics
33141A

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work, and including any code produced using generative AI systems. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

1 INTRODUCTION

This project was carried out within the course *Algorithms for Massive Data* taught by Professor Dario Malchiodi. Based on the publicly available Amazon Book Review dataset on Kaggle, the objective was to develop a ranking system using PageRank-based Link Analysis to identify the most influential books. Specifically, the books were linked if they were rated with high scores by at least two different users. In other words, the importance of a book depended on how many other books that shared a significant number of high-rating users with it.

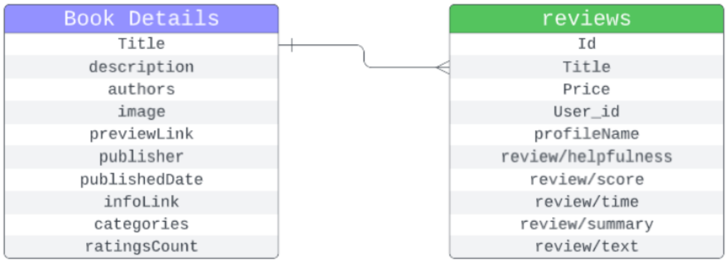
Since inconsistencies in book titles could influence the link structure, the project also investigated a second approach, in which similar titles were identified and merged using a Jaccard-based similarity measure before applying the PageRank algorithm. Finally, the project compared the results obtained with and without merging duplicated book titles.

2 DATASET

The original dataset consists of two files: *Books-rating.csv* and *books-data.csv*, as illustrated in Figure 1. These datasets refer to the same collection of books and can be linked via the *Title* attribute. The *books-data.csv* file provides metadata for each book, such as title, authors, description, and image, while *Books-rating.csv* contains user IDs, review scores, ratings' helpfulness, and timestamps. Together, these datasets allow us to combine user rating behavior with book-level descriptive attributes.

2.1 Data preprocessing

For the purpose of the project, *Books-rating.csv* dataset was mainly used to extract three attributes *Title*, *User-id* and *review/score*. The file contains in total 3 million rating entries, corresponding to 212,404 unique book titles, and 3 million users giving reviews.



Book Details
Title
description
authors
image
previewLink
publisher
publishedDate
infoLink
categories
ratingsCount

reviews
Id
Title
Price
User_id
profileName
review/helpfulness
review/score
review/time
review/summary
review/text

Figure 1. Columns of Books-data.csv and books-rating.csv and their relationship (source: Kaggle).

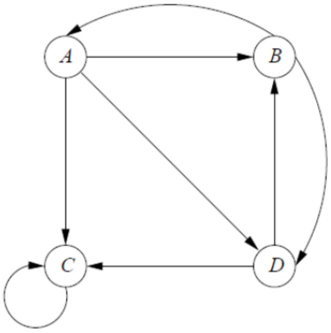


Figure 2. Node *C* is a dead end (left) and a spider trap (right) (source: Textbook *Mining of Massive Datasets*)

To prepare the data for PageRank-based link analysis, several preprocessing steps were applied. First, *Books-rating.csv* dataset was filtered to retain only reviews with a score of at least 4, which contributed approximately 2.5 million entries of the entire file. Checking for NAN values was the next step. A total of 208 missing entries were found in the *Title* attribute and 561,787 in *User-id*. All records containing missing values in either column were then removed.

For the second approach, a further preprocessing step was also applied, where book titles were first normalized by converting them to lowercase and removing spaces, punctuation, and special characters before a Jaccard similarity measure was computed to detect near-duplicate titles. As a result, the number of unique titles decreased from 380 to 378.

Besides, because of RAM limitations, the analysis was further restricted to books with at least 81 unique users who rated them. The filtered dataset, consequently, consisted of a total of 75,786 rows with 60,081 unique users and 380 unique book titles.

3 METHODOLOGY

3.1 PageRank Algorithm

The PageRank algorithm computes an important score for each node in a graph by analyzing how these nodes are connected to each other.

In this project, books were modeled as nodes, and an edge between $book_i$ and $book_j$ was created if at least two users rated both books with a high score (*review/score* ≥ 4). The expected outcome of PageRank algorithm is a column vector v_k, whose j^{th} component is the probability representing how likely a user is to land on that book by following edges in the graph.

The PageRank vector v_k is computed iteratively by repeatedly multiplying the transition matrix $\mathcal{M}$ with the distribution vector from the previous iteration. After k iterations, the updated vector is

$$v_k = \mathcal{M} v_{k-1} \tag{1}$$

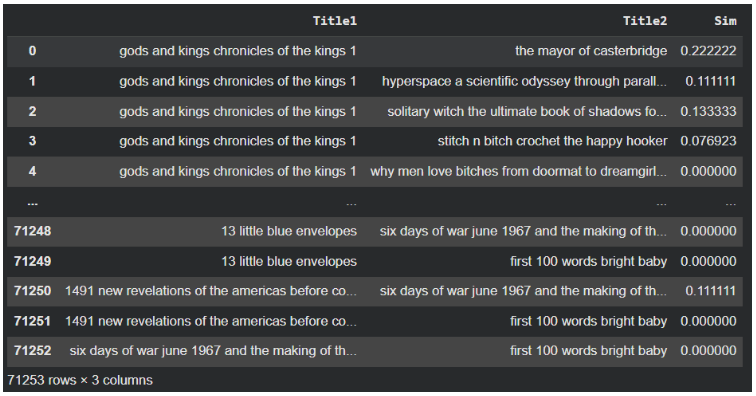
The convergence is reached when applying the transition matrix no longer changes the distribution vector.

However, in most real-world graphs, two issues commonly arise in link analysis: *dead ends* and *spider traps*. A dead end is a node with no outgoing edges, while a spider trap is a group of nodes that link only to each other and have no edges leading outside the group, as shown in Figure 2. In other words, a spider trap is a set of nodes with no dead ends but no arcs out. To avoid these problems, the calculation of PageRank is modified by introducing a probability $(1-\beta)$, which allows a user to randomly teleport to a random node. The damping factor β was set 0.85 in this project. Therefore, the update rule becomes

$$v_k = \beta \mathcal{M} v_{k-1} + (1-\beta)\frac{1}{n}\mathbf{1} \tag{2}$$

where:

- n is the number of nodes in the graph



	Title1	Title2	Sim
0	gods and kings chronicles of the kings 1	the mayor of casterbridge	0.222222
1	gods and kings chronicles of the kings 1	hyperspace a scientific odyssey through parall...	0.111111
2	gods and kings chronicles of the kings 1	solitary witch the ultimate book of shadows fo...	0.133333
3	gods and kings chronicles of the kings 1	stitch n bitch crochet the happy hooker	0.076923
4	gods and kings chronicles of the kings 1	why men love bitches from doormat to dreamgirl...	0.000000
...	...	...	...
71248	13 little blue envelopes	six days of war june 1967 and the making of th...	0.000000
71249	13 little blue envelopes	first 100 words bright baby	0.000000
71250	1491 new revelations of the americas before co...	six days of war june 1967 and the making of th...	0.111111
71251	1491 new revelations of the americas before co...	first 100 words bright baby	0.000000
71252	six days of war june 1967 and the making of th...	first 100 words bright baby	0.000000

71253 rows × 3 columns

Figure 3. Pairs of book titles with corresponding Jaccard similarity values (source: *Source code*).

- $\mathbf{1}$ is a column vector of all ones (the number of the entries is equal to the number of the nodes)
- $\frac{1}{n}\mathbf{1}$ is the uniform probability distribution.

3.1.1 Transition matrix $\mathcal{M}$

The transition matrix $\mathcal{M}$ describes how the probability distribution of the random surfer evolves after one iteration. In this project, $\mathcal{M}$ represented the book-book transition probabilities. Each entry m_{ij} denoted the probability of moving from book j to book i.

To construct this matrix, we first computed, for every pair of books, the number of users who rated both books with a high score. These counts formed a weighted adjacency matrix. Each column was then normalized by its column sum so that

$$\sum_i m_{ij} = 1,$$

ensuring that $\mathcal{M}$ was column-stochastic. The resulting transition matrix has shape $(380, 380)$, corresponding to the 380 unique book titles included in the final dataset.

3.2 Jaccard Algorithm

Considering that duplicated or highly similar book titles could affect the outcome of the link analysis, the Jaccard algorithm computation was applied to detect such cases. Mathematically, for two sets Set_1 and Set_2, Jaccard similarity is defined as

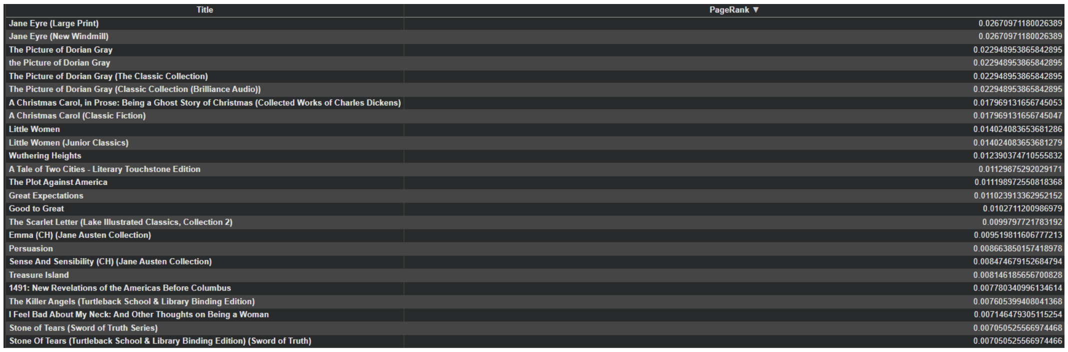
$$Jaccard(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2} \tag{3}$$

In this project, each book title was represented as a set of the words it contains. Thus, two titles were compared by computing the Jaccard similarity between their corresponding word sets, allowing to identify titles that are nearly identical or refer to the same book. As a result, a total of 71,253 title pairs were generated together with their Jaccard similarity values, as shown in Figure 3.

A threshold of 0.4 was applied to the Jaccard similarity values, meaning that all title pairs with a similarity of at least 40% were retained for merging. Titles identified as similar were grouped into clusters, and each cluster was assigned a canonical representative title. These canonical titles were then used to replace all titles within their respective clusters, ensuring that duplicated or highly similar books were represented consistently. The resulting cleaned set of titles was subsequently used as input to the PageRank algorithm for further analysis.

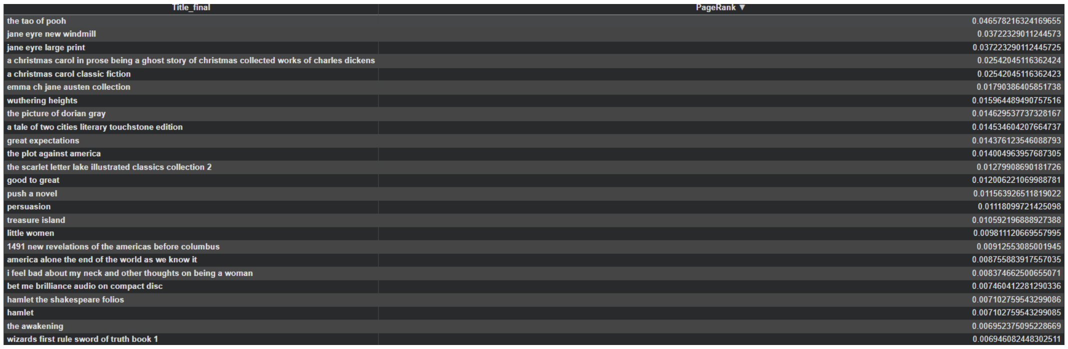
4 DISCUSSION AND CONCLUSION

Figure 4 and Figure 5 present the final book rankings obtained without and with title merging, respectively. In the case without merging, several books with nearly identical titles appeared as separate entries and received identical PageRank values, which was not informative for link analysis and artificially inflated their perceived importance. After applying the clustering procedure, notable



Title	PageRank ▼
Jane Eyre (Large Print)	0.02670971180026389
Jane Eyre (New Windmill)	0.02670971180026389
The Picture of Dorian Gray	0.022948953865842895
the Picture of Dorian Gray	0.022948953865842895
The Picture of Dorian Gray (The Classic Collection)	0.022948953865842895
The Picture of Dorian Gray (Classic Collection (Brilliance Audio))	0.022948953865842895
A Christmas Carol, in Prose: Being a Ghost Story of Christmas (Collected Works of Charles Dickens)	0.017969131656745053
A Christmas Carol (Classic Fiction)	0.017969131656745047
Little Women	0.014024083653681286
Little Women (Junior Classics)	0.014024083653681279
Wuthering Heights	0.012390374710555832
A Tale of Two Cities - Literary Touchstone Edition	0.01129875292029171
The Plot Against America	0.011198972550818368
Great Expectations	0.011023913362952152
Good to Great	0.0102711200986979
The Scarlet Letter (Lake Illustrated Classics, Collection 2)	0.0099797721783192
Emma (CH) (Jane Austen Collection)	0.009519811606777213
Persuasion	0.008663850157418978
Sense And Sensibility (CH) (Jane Austen Collection)	0.008474679152684794
Treasure Island	0.008146185656700828
1491: New Revelations of the Americas Before Columbus	0.007780340996134614
The Killer Angels (Turtleback School & Library Binding Edition)	0.007605399408041368
I Feel Bad About My Neck: And Other Thoughts on Being a Woman	0.007146479305115254
Stone of Tears (Sword of Truth Series)	0.007050525566974468
Stone Of Tears (Turtleback School & Library Binding Edition) (Sword of Truth)	0.007050525566974466

Figure 4. Book rankings without applying Jaccard similarity computation (source: *Source code*).



Title_final	PageRank ▼
the tao of pooh	0.046578216324169655
jane eyre new windmill	0.03722329011244573
jane eyre large print	0.037223290112445725
a christmas carol in prose being a ghost story of christmas collected works of charles dickens	0.02542045116362424
a christmas carol classic fiction	0.02542045116362423
emma ch jane austen collection	0.01790386405851738
wuthering heights	0.015964489490757516
the picture of dorian gray	0.014629537737328167
a tale of two cities literary touchstone edition	0.014534604207664737
great expectations	0.014376123546088793
the plot against america	0.014004963957687305
the scarlet letter lake illustrated classics collection 2	0.01279908690181726
good to great	0.012006221069988781
push a novel	0.011563926511819022
persuasion	0.01118099721425098
treasure island	0.010592196888927388
little women	0.009811120669557995
1491 new revelations of the americas before columbus	0.00912553085001945
america alone the end of the world as we know it	0.008755883917557035
i feel bad about my neck and other thoughts on being a woman	0.008374662500655071
bet me brilliance audio on compact disc	0.007460412281290336
hamlet the shakespeare folios	0.007102759543299086
hamlet	0.007102759543299085
the awakening	0.006952375095228669
wizards first rule sword of truth book 1	0.006946082448302511

Figure 5. Book rankings applying Jaccard similarity computation (source: *Source code*).

differences in the rankings emerged. For example, *The Tao of Pooh* became the top-ranked book, even though it did not appear in the top 25 previously. Conversely, *The Picture of Dorian Gray* lost prominence after merging, with its PageRank decreasing from 0.0229 to 0.0146, indicating that its earlier importance was partly due to duplicated editions reinforcing one another.

Therefore, combining the PageRank algorithm with a similarity detection method like Jaccard could lead to more reliable rankings, as it helps consolidate duplicated or highly similar titles. However, Jaccard similarity based solely on raw text is not always an ideal solution. It cannot distinguish between books that differ in meaningful ways, such as editions, languages, formats, or publication conditions, which may influence readers' preferences and result in different rating opinions. Thus, more sophisticated similarity measures, for example applying machine-learning-based models, could be used to capture deeper semantic relationships between titles.

APPENDIX

The full implementation of this project is provided as a Jupyter notebook and can be executed on Google Colab. The source code is available at: `https://github.com/chihoang811/chihoang811/tree/feea3326c704e15b1abc1dd6bf9ee2c02feea674/PageRank_based_Link_analysis_on_Books`