Searching Similar Books Based on Book-Tag Network

Mingxi Zhang*, Yating Wang
College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai, China

Email address:
WAXL7461@aliyun.com(Mingxi Zhang), 1612807336@qq.com (Yating Wang)
*Corresponding author

Acknowledgements

This work was supported in part by the National Science Foundation of China under Grant 62002225, and in part by the Natural Science Foundation of Shanghai grant 21ZR1445400.

Abstract: In the era of information explosion, there are countless book resources, accompanied by the increasingly serious problem of information overload. How to effectively and accurately find the books that users need from massive book resources has become a hot spot. In this paper, we propose a Sim Rank-based system to search similar books in a "book-tag" network of bipartite type. First, we build a "book-tag" network by "description" relationships between books and tags, in which books and tags are regarded as nodes, and relationships between the books and the tags are regarded as edges. Second, we adopt Sim Rank to calculate the similarities over the "book-tag" network, and then return the top k most similar books according to the similarity scores between the given book and each candidate one. Empirical studies on real dataset demonstrate the effectiveness and accuracy of the system.

Keywords: Sim Rank; Bipartite network; Similarity score

1. Introduction

With the rapid development of information technology and internet technology, the scale of data is expanding at an unprecedented speed. “Big data” brings great opportunities and challenges to all walks of life. Especially in the field of book publishing, book resources are increasing day by day, with more than 500,000 books published every year, which leads to an increasingly serious problem of information overload. Therefore, how to effectively obtain the books that users need from the massive book database has become an urgent problem to be solved.

Similarity search technology refers to finding the most similar object in the set of candidate objects for a given sample object, which has attracted the attention of many scholars and is widely used in many fields, such as social network analysis, recommendation systems, data mining, and information retrieval. Considering the actual situation, what people care about is not the similarity of all objects, but how to obtain objects that are particularly similar to a given query object. For example, in the similarity search of books, when the user enters a book, it is more hoped that the system will return a list of the most similar books.

The existing similarity search methods can be roughly divided into two categories: one is content-based measures for specific domain, such as text similarity matching [1]; the other is link-based measures between two objects, such as Sim Rank [2] and P-Rank [3]. Sim Rank is a classical similarity measure based on clear human intuition and solid theoretical background, which has been widely studied and applied in recommendation systems [4], pattern matching [5], spam detection [6], and many other network applications [7, 8, 9]. The main idea of Sim Rank is that a node is most similar to itself, and two nodes are similar if they are related to similar nodes. Different from the similarity measure which needs to build a hierarchy manually, Sim Rank is domain independent and suitable for any domain with object-to-object relationship. Therefore, we adopt Sim Rank to calculate the similarities of books and realize the similarity search of massive book resources.

In this work, we propose a Sim Rank-based system to search similar books in a “book-tag” bipartite network. Figure 1 shows the system process framework to search similar books. First, we preprocess the original data of the book to remove redundant information and complex indexes in the text. At the end of the preprocessing step, stop words should be removed because they have nothing to do with the key theme of the book, and stemming should also be extracted because they add redundant text indexes. Second, we build a “book-tag” network based on “description” relationships between books and tags, in which books and tags are regarded as nodes, and relationships between the books and the tags are regarded as edges. And then we remove the noisy links based on TF-IDF model [10], in which the lower informative tags are removed. Last, we adopt Sim Rank to calculate the similarities over the “book-tag” network, and return the top k most similar books according to the similarity score between the given book and each candidate one. Our main contributions are as follows:

1) Based on Sim Rank model, we propose a system to search similar books. In our method, it is very important to consider the tag information corresponding to the books when searching similar books.

2) We build a "book-tag" network based on "description" relationships between books and tags. For the similarity search of books, we only need to find out the tags that have similar relationships in the "book-tag" bipartite network, and then we can find similar books.

3) We conducted a comprehensive experiment to test the system proposed in this paper. These results show that the book searched by Sim Rank has a strong similarity with the original input book, which proves that our system accurately and effectively realizes the similarity search of books.

2. Related Work

In the link-based similarity search methods, book structure semantics are combined with metadata and on tologies. These approaches comprehensively select book topics, browsing, purchase, registration, scoring, demographic information, social networks and other multi-dimensional information, and use weighting, switching, mixed presentation, feature combination, series, meta-level mixing, etc., to combine two or more information to obtain better search results.

Sim Rank [2] is one of the most influential link-based measures because it relies on the simple intuition that similar objects point to similar objects. In addition, it captures structural similarity well for the reason that it no longer only considers the direct in-link among nodes but also considers the indirect in-link. Moreover, Sim Rank is a general model in that as long as objects and their relationships are built as a network, the similarity measure can be defined based on the structural context of objects.

There are also other measures of link-based. P-Rank [3] is similar to Sim Rank, but it improves Sim Rank by incorporating both in-links and out-links between objects and controls the relative importance of in/out-links by means of a parameter. Path Sim [11] calculates the similarity between nodes by the numbers of meta-paths on a heterogeneous network. Sim Cat [12] defines the similarities between nodes by incorporating category information and aggregating relationship network structures. SLING [13] uses an index-based Sim Rank algorithm to calculate the similarity scores, which requires its index structure to be rebuilt from scratch whenever the input network is updated. TSF [14] also adopts an index-based approach and allows for efficient updates, which uses a special index structure to calculate top k queries. Probe Sim [15] assesses similarity without pre computing index structure, so it can support real-time computation of top k queries on a dynamic network. Sim Rank* [16] remedies the problem of "zero-similarity" in SimRank, which enriches semantics without suffering from increased computational overhead. PR Sim [17] uses the structure of networks to efficiently answer single-source Sim Rank queries. Uni Walk[18] calculates the similarities between nodes based on Monte Carlo, which could directly locate the top k similar vertices for any single source by sampling paths originating from the single source. Sim Push [19] is state-of-art index-free method and provides a theoretical guarantee.

3. Network Construction

3. 1 Data preprocessing

In the book data, the text content generally contains a lot of unnecessary information, such as stop words and punctuation, which can affect the accuracy of extracting keywords of the text. It is generally considered that the stop words and punctuation are of no importance for extracting the theme of text. What's more, removing the redundant information from the text, which can improve the accuracy and credibility of keyword extraction in real dataset. In this work, we remove stop words according to the public stop word list XPO6, which can be found on the website http://xpo6.com/download-stop-word-list.As we all know, a word may have different grammatical expressions. Specifically, we can extract word stems from words with different tenses, such as extracting "put" from "putting"; we can also extract word stems from words with plural forms, such as extracting "cat" from "cats". Extracting word stems can reduce the text index and reduce the calculation time of the algorithm. And, it can ensure that the effectiveness of keyword extraction is improved. Porter Stemmer [20] is one of the most commonly used to kenizers, which has the advantage of accurately extracting word stems. In this paper, we use Porter Stemmer to extract word stems.

3.2 Building "book-tag" bipartite network

A "book-tag" network is defined as a bipartite network $G = (V, E)$, where$V = V_b \cup V_t$, $V_b$and $V_t$represent the sets of books and tags respectively; E denotes the set of edges of "description" relationship between books and tags, and an edge$e(b_i, t_j) \in E$ denotes a book$b_i \in V_b$is described as a tag $t_j \in V_t$.

For a given book, we get some keywords after preprocessing. Then we use these keywords as tags and the ASIN of each book as books to construct a "book-tag" bipartite network. In the book dataset, each book can be represented by multiple tags, and each tag can also describe multiple books. All books and tags are regarded as nodes, and all the "description" relationships between books and tags construct edges. Figure 2 shows a "book-tag" network. For the similarity search of books, we only need to find out the tags that have similar relationships in the "book-tag" bipartite network, and then we can find similar books.

3.3 Removing Noisy Links

Noisy links are the tags that cannot effectively discriminate books while computing the similarities. During searching similar books, noisy links not only affect search results but also incur the expensive time and space overhead, so it is necessary to be removed. Term Frequency-Inverse Document Frequency (TF-IDF) [10] could be seen as a promising method to find noisy links. It is a statistical method to assess whether a tag is important for a book. In other words, if a tag describes a book, and this tag rarely appears in the description of other books. It shows that the tag has a better discrimination ability. Term frequency(TF) indicates how often a tag appears in a book, it is defined as:

$$tf_{t,b} = \frac{n_{t,b}}{|O(b)|} (1)$$

where $n_{t,b}$is the number of a tag t is associated with a book b, and we set $n_{t,b}$roughly as 1. There is no duplicate tags t will appear in the description of the book b, because the tag is different from the text; and $|O(b)|$ is the number of out-neighbors of book b. The inverse document frequency(IDF) is a measure of the general importance of a tag, it is defined as:

$$idf_t = \frac{\log |n_b|}{|I(t)|} (2)$$

where $|n_b|$ is the total number of books in the data, $|I(t)|$ is the number ofin-neighbors of tag t. Based on TF and IDF, the TF-IDF value for tag t and book b is defined as:

$$tfidf_{t,b} = tf_{t,b} * idf_t (3)$$

Intuitively, a tag with a higher TF-IDF value indicates it is more discriminative for the books. And the tags with lower TF-IDF value should be removed to avoid affecting search results. Then, we remove noisy links with low TF-IDF values according to a threshold δ, defined as $\delta = (max - min) * h + min$where$h \in (0,1)$. In the "book-tag" network, the links correspond to TF-IDF values lower than δ are removed before similarity computation.

4. Sim Rank Model Applied to the Network

Compared with content-based approaches, link-based approaches are able to mine the semantic information of books, which can effectively search semantic similar books. Among existing link-based similarity computations, Sim Rank can be regarded as one of the most attractive methods for computing similarity scores. The reason is as follows. First, the main idea of the Sim Rank is "two objects are similar if they are related to similar objects". It no longer only considers direct in-link among nodes but also considers indirect in-link. Second, Sim Rank is a general model that can be applied in all similarity search fields. Last, Sim Rank is suitable for bipartite networks, while the "book-tag" network happens to be a bipartite network.

We adopt Sim Rank to calculate similarity in a "book-tag" network. Our key observation is that "similar books contain similar tags, and similar tags describe similar books", which is consistent with the intuition of Sim Rank. Let $S(b_1, b_2)$ denotes the similarity between books $b_1, b_2 \in V_b$, and let $S(t_1, t_2)$ denotes the similarity between tags $t_1,\ t_2 \in V_t$. If $b_1 = b_2$, $S(b_1, b_2) = 1$, and similarly, $S(t_1, t_2) = 1$ if $t_1 = t_2$. For $b_1 \neq b_2$, $S(b_1, b_2)$ is defined as:

$$S(b_1, b_2) = \frac{C}{|O(b_1)||O(b_2)|} \sum_{i=1}^{|O(b_1)|} \sum_{j=1}^{|O(b_2)|} S(O_i(b_1), O_j(b_2)) \quad (4)$$

and for $t_1 \neq t_2$, $S(t_1, t_2)$ is defined as:

$$S(t_1, t_2) = \frac{C}{|I(t_1)||I(t_2)|} \sum_{i=1}^{|I(t_1)|} \sum_{j=1}^{|I(t_2)|} S(I_i(t_1), I_j(t_2)) \quad (5)$$

where C is a constant between 0 and 1, which is typically set as 0.8 according to [2]; $|O(b_1)|$ is the number of elements of the set $O(b_1)$, $|I(t_1)|$ is the number of elements of the set $I(t_1)$, $O(b_1)$ is the number of out-neighbors of book b_1 and $I(t_1)$ is the number of in-neighbors of tag t_1. $O_i(b_1)$ denotes the *i*-th out-neighbor of book b_1, and $I_i(t_1)$ denotes the *i*-th in-neighbor of tag t_1, where $1 \leq i \leq |O(b_1)|$ and $1 \leq i \leq |I(t_1)|$. If $O(b_1) = \emptyset$ or $O(b_2) = \emptyset$, $S(b_1, b_2) = 0$, and similarly $S(t_1, t_2) = 0$.

The similarity scores are calculated iteratively. At the *l*-th iteration, $R_l(b_1, b_2)$ denotes the similarity scores between book b_1 and book b_2, $R_l(t_1, t_2)$ denotes the similarity scores between tag t_1 and tag t_2. If $b_1 = b_2$, $R_0(b_1, b_2) = 1$ at $l = 0$, otherwise $R_0(b_1, b_2) = 0$, and the same for $R_0(t_1, t_2)$. When $l = 2,3,4, \ldots, R_{l+1}(b_1, b_2)$ is defined as $R_{l+1}(b_1, b_2) = 1$ if $b_1 = b_2$, otherwise:

$$R_{l+1}(b_1, b_2) = \frac{C}{|O(b_1)||O(b_2)|} \sum_{i=1}^{|O(b_1)|} \sum_{j=1}^{|O(b_2)|} R_l(O_i(b_1), O_j(b_2)) \quad (6)$$

and similarly, $R_{l+1}(t_1, t_2)$ is defined as: $R_{l+1}(t_1, t_2) = 1$ if $t_1 = t_2$, otherwise:

$$R_{l+1}(t_1, t_2) = \frac{C}{|I(t_1)||I(t_2)|} \sum_{i=1}^{|I(t_1)|} \sum_{j=1}^{|I(t_2)|} R_l(I_i(t_1), I_j(t_2)) \quad (7)$$

5. Results

The experimental machine is configured with Intel(R) Xeon(R) Bronze3106 CPU@1.70GHzand 128GB RAM, under Windows 2012R, and development environment is VS C++ 2019. The experiment was conducted on the real set of Amazon data (http://snap.stanford.edu/data/amazon-meta.html). We select 468,658 representative objects from the dataset to construct a "book-tag" bipartite network, include76,272 tags, 392,386 books, and 1,924,792 "description" relationships.

Table 1-3 show the top-5 returned results of our method for 3 different queries on the Amazon dataset. Among them, the query "The New York Times Cook Book" is a book about Cooking, Food and Wine. In the ranking of the returned results of this query, the results corresponding to the top-5 ranking positions are obviously related to Cooking. For example, the result of sort 1 is the query itself, "The New York Times 60-Minute Gourmet" of sort 2 is a book about fast food, "The Silver Palate Cookbook" of sort 3 is a flavored cookbook, "The Fannie Farmer Cookbook: Anniversary" of sort 4is a cookbook related to farmers, and "An American Feast: A Celebration of Cooking on Public Television" ranked 5 is also a book related to cooking. The query "SQL Server 2000 Developer's Guide" is a book about SQL server development. Obviously, sorts 1, 2, 3, 4, and 5 are all books related to SQL server. In addition, the results of query "The Complete Idiot's Guide to Communicating with Spirits" can also be similarly analyzed, so we do not repeat them here. Empirical studies on real dataset demonstrate the effectiveness and accuracy of the system.

6. Conclusion

This paper proposes a Sim Rank-based system to search similar books in a "book-tag" network of bipartite type, which is divided into two stages. The first stage is to build a "book-tag" network by "description" relationships between books and tags, in which books and tags are regarded as nodes, and relationships between the books and the tags are regarded as edges. The second stage is to adopt Sim Rank to calculate the similarities over the "book-tag" network, and then returned the top k most similar books according to the similarity scores between the given book and each candidate one. Empirical studies on real dataset demonstrate the effectiveness of this system. In future work, we hope to improve the computational efficiency of adapting to large networks. Although the search process does not require pre-calculation, its running time is exponential, which is expensive especially for large networks. We believe that the time cost can be significantly reduced by referring to existing studies.

References

[1] Wang, L., Li, Y., Huang, J., & Lazebnik, S. (2017). Learning two-branch neural networks for image-text matching tasks. IEEE Transactions on Pattern Analysis & Machine Intelligence, PP(99), 1-1.

[2] Jeh, G., & J Widom. (2002). Sim Rank: a measure of structural-context similarity. the eighth ACM SIGKDD international conference. ACM.

[3] Zhao, P., Han, J., & Sun, Y. (2009). P-Rank: A comprehensive structural similarity measure over information networks. Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009. ACM.

[4] Huynh, H. X., Phan, N. Q., Pham, N. M., Pham, V. H., & Ismail, M. (2020). Context-similarity collaborative filtering recommendation. IEEE Access, PP(99), 1-1.

[5] Spiliopoulos, & Leonidas. (2013). Beyond fictitious play beliefs: incorporating pattern recognition and similarity matching. Games & Economic Behavior, 81, 69-85.

[6] Tseng, C. Y., Sung, P. C., & Chen, M. S. (2011). Cosdes: a collaborative spam detection system with a novel e-mail abstraction scheme. IEEE Transactions on Knowledge &Data Engineering, 23(5), 669-682.

[7] Wu, D., Shi, J., & Mamoulis, N. (2018). Density-based place clustering using geo-social network data. IEEE Transactions on Knowledge & Data Engineering, 1-1.

[8] Zou, Y., X Yao, Chen, Z., & Zhao, M. (2018). Verifiable keyword-based semantic similarity search on social data outsourcing. IEEE Access, PP(99), 1-1.

[9] Li, Y., Gu, C., Dullien, T., Vinyals, O., & Kohli, P. (2019). Graph matching networks for learning the similarity of graph structured objects.

[10] Martineau, J., & Finin, T. (2009). Delta TFIDF: An Improved Feature Space for Sentiment Analysis. Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009. DBLP.

[11] Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). Pathsim: meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment, 4(11), 992-1003.

[12] Choudhury, A., Sharma, S., Mitra, P., Sebastian, C., Naidu, S. S., &Chelliah, M. (2015). SimCat: an entity similarity measure for heterogeneous knowledge graph with categories. ACM.

[13] Tian B., & Xiao X. (2016). SLING: A near-optimal index structure for simrank. Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, PP, 1859-1874.

[14] Shao, Y., Cui, B., Chen, L., Liu, M., &Xie, X.(2015). An efficient similarity search framework for simrank over large dynamic graphs. Proceedings of the VLDB Endowment, 8, 838-849.

[15] Liu, Y., Zheng, B., He, X., Wei, Z., Xiao, X., & Zheng, K., et al. (2017). Probesim: scalable single-source and top-k simrank computations on dynamic graphs. Proceedings of the VLDB Endowment, 11(1).

[16] Yu, W., Lin, X., Zhang, W., Pei, J., & Mccann, J. A. (2019). Simrank*: effective and scalable pairwise similarity search based on graph topology. VLDB Journal, 28(3), 401-426.

[17] Wei, Z., He, X., Xiao, X., Wang, S., & Wen, J. R. (2019). Prsim: sublinear time simrank computation on large power-law graphs.

[18] Song, J., Luo, X., Gao, J., Chang, Z., & Hu, W. (2017). Uniwalk: unidirectional random walk based scalable simrank computation over large graph. IEEE Transactions on Knowledge & Data Engineering, PP(99), 1-1.

[19] Shi, J., Jin, T., Yang, R., Xiao, X., & Yang, Y.(2020). Real time index-free single source simrank processing on web-scale graphs. Proceedings of the VLDB Endowment.

[20] Karaa, W., & Griba, N.(2013). Information Retrieval with Porter Stemmer: A New Version for English.

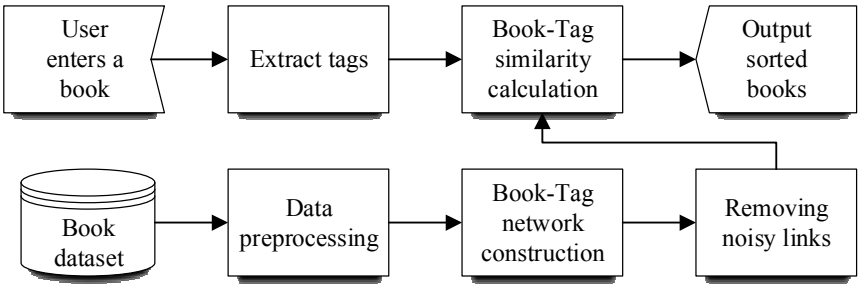

Figure 1. System process framework

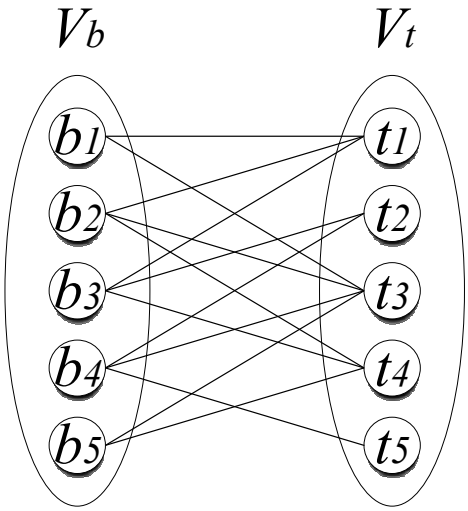

Figure 2. "book-tag"bipartite network

Table 1. The results of the query "The New York Times Cook Book"

k	*The New York Times Cook Book*
1	*The New York Times Cook Book*
2	*The New York Times 60-Minute Gourmet*
3	*The Silver Palate Cookbook*
4	*The Fannie Farmer Cookbook : Anniversary*
5	*An American Feast : A Celebration of Cooking on Public Television*

Table 2. The results of the query "SQL Server 2000 Developer's Guide"

k	*SQL Server 2000 Developer's Guide*
1	*SQL Server 2000 Developer's Guide*
2	*SQL Server 2000 Programming by Example*
3	*SQL Server 2000: A Beginner's Guide (Book/CD-ROM)*
4	*Microsoft SQL Server 2000 Database Development From Scratch*
5	*Microsoft SQL Server 2000 Bible with CD-ROM*

Table 3. The results of the query "The Complete Idiot's Guide to Communicating with Spirits"

k	*The Complete Idiot's Guide to Communicating with Spirits*
1	*The Complete Idiot's Guide to Communicating with Spirits*
2	*How to Meet & Work With Spirit Guides*
3	*How to Communicate With Spirits*
4	*The Complete Idiot's Guide to Being Psychic*
5	*The Complete Idiot's Guide to Leadership*