Book Recommender System Using Linked Data for Improving Serendipity

著者（英）	WENG Renlou , Masao TAKAKU
journal or publication title	Journal of Japan Society of Information and Knowledge
volume	29
number	2
page range	164-169
year	2019-05
URL	http://hdl.handle.net/2241/00157919

doi: 10.2964/jsik_2019_031

第 27 回年次大会予稿

Book Recommender System Using Linked Data for Improving Serendipity

Renlou WENG[1]*, Masao TAKAKU[2]

1 Graduate School of Library, Information and Media Studies, University of Tsukuba
E-mail: s1821641@s.tsukuba.ac.jp
2 Faculty of Library, Information and Media Science, University of Tsukuba
E-mail: masao@slis.tsukuba.ac.jp
*Corresponding Author

Recent years, to overcome the flood of information, recommender systems (RSs) are being used in many scenarios, such as online shopping stores, movie website and so on. However, many recommender algorithms focus on accuracy based on a user profile, which may lead to reducing user's satisfaction. As high-accuracy based RSs suggest similar items that the user may have known before. As a result, the recommendation leads to hurt user's satisfaction. And there is a concept called serendipity which is a way to address this problem. This paper focuses on improving the serendipity of the RSs. We use two approaches, the variety of resources in Linked Data and author similarity. We extract author information from DBpedia dataset, which is one of the datasets in Linked Data. Then we calculate author similarity based on this information. This paper describes methods of book recommendations from the Book-Crossing dataset, reports recommendation results and discusses our recommender system according to our results.

Keywords: Book recommendation, Serendipity, Linked Data, Author similarity, DBpedia

1 INTRODUCTION

Recommender systems (RSs) have become increasingly popular in recent years, and have been utilized in many fields, such as movies, books, songs and other fields [1]. And this paper focuses on book recommender system.

Traditionally, book RSs tend to offer books that similar to the book which has indicated as interesting for users. Suppose that a user likes 1Q84 and has bought 1Q84 BOOK 1 in Amazon, and Amazon book RS suggests 1Q84 BOOK 2 or BOOK 3 to the user. As a result, the user tends to be bored with offered books which the user has already known before. This problem may lead to reducing users' satisfaction from book RSs. And one of the ways to address this problem is to increase the serendipity of a RSs.

This paper uses two approaches to improve serendipity in book RSs. Firstly, as book RSs always recommend the same author's books to users, it is difficult to find other authors' books for users. As traditional RSs recommend same author's books, we do not directly calculate the similarity of books in this paper. We calculate author's similarity to recommend different authors' books. It leads to improve serendipity in recommendations. Secondly, we use Linked Data [2] which is a method of publishing structured data. We can easily obtain various kinds of author's information, such as author's genre, award and so on. Kotkov [3] indicated that a variety of information may improve serendipity in RSs.

The goal of this paper is to improve the serendipity of book RSs. And we use two approaches (author similarity and Linked Data resource) to improve the serendipity.

2 RELATED WORK

In this section, we review related works in two aspects, which are serendipity and Linked Data in a recommender system.

2.1 Serendipity in RSs

Oku et al. [4] presented a system that increased possibly serendipitous recommendations by selecting books whose content is a mixture of the content features of two books from the user's preference. We do not use book features but focusing on book author's features to improve serendipitous recommendations. Yamaba et al. [5] proposed a serendipity-oriented recommender system based on Folksonomy. Yamaba used Folksonomy as an indicator to recommend serendipitous items. In this paper, we focus on using Linked Data resource other than Folksonomy and recommending serendipitous items.

2.2 Linked Data in RSs

Di Noia et al. [6] proposed a recommendation method that used Linked Data to calculate the similarity between movies based on movies' properties (e.g., the director, the genre, the starring, etc.) in DBpedia [7]. Noia used a direct relationship to give a recommendation. We do not calculate the direct relationship of items but consider a recommendation from an indirect perspective on improving serendipity.

Ichise et al. [8] tried to use Linked Data to recommend similar author based on the author's two properties (influenced, influencedBy). For instance, an author's similar authors are extracted from the author's influenced people and influencedBy people. In this paper, we do not calculate a direct relationship.

3 SYSTEM DESIGN

We suppose that a user is interested in a book and he or she is also interested in the book author's related authors and their books. Considering of related authors, it is possible to think that the user may ask a question like "who is the most similar author according to the author's genre?" and "who is the most similar author according to the author's influenced people". Although, it may be difficult for the user to find similarity authors by his/her own self, our system can calculate related authors based on these questions and recommend the books.

3.1 System Overview and Method

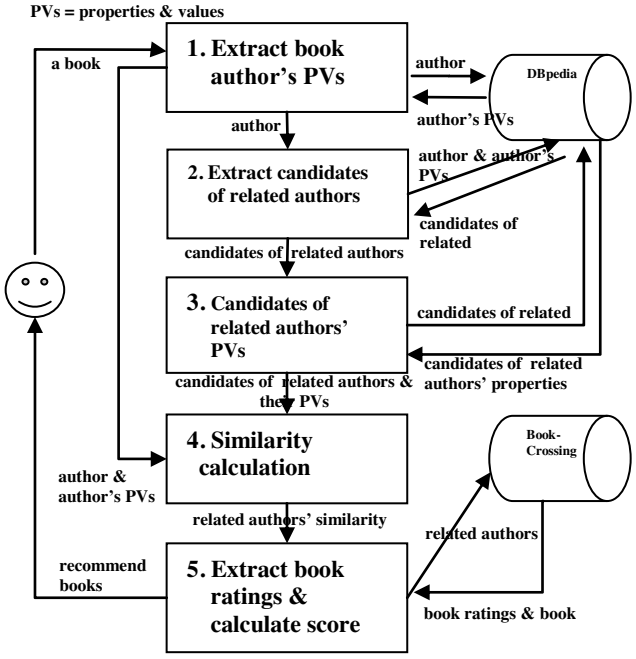

Figure 1: Overview of our system

We can imagine that a user is using an online book store, it is easy to access the user's profile, such as the pages the user is focusing on, the retrieval history the user has made and so on. The online book store can know the book that the user is interested in based on his/her profile and recommend other books.

Figure 1 is our system overview and we explain it in detail in this section. As we do not have any such user profile at all, thus, our system requires a user to type a book title which the user is interested in into our recommender system.

In the step 1, we extract the book author's properties and values from DBpedia [7]. Using DBpedia SPARQL endpoints [9], we can easily extract authors' structured information from diverse properties. The authors' information contains many properties, such as author's genre, influenced people and so on. We use three properties (genre, influencedBy, subject) in this paper. Here, we take Hermann Hesse as an example. If we want to extract Hermann Hesse's three properties (genre,

influencedBy, subject) and their values, we use a SPARQL [9] query shown in Figure 2.

```
prefix dbpedia: <http://dbpedia.org/resource/>
prefix dbpedia-owl: <http://dbpedia.org/ontology/>
prefix dcterms: <http://purl.org/dc/terms/>
select ?genre ?influencedBy ?subject
where
{
 {dbpedia:Hermann_Hesse dbpedia-owl:genre ?genre }
 UNION { dbpedia:Hermann_Hesse dcterms:subject ?subject}
 UNION {dbpedia:Hermann_Hesse dbpedia-owl:influencedBy
 ?influencedBy }
}
```

Figure 2: SPARQL query for extracting Hermann Hesse's three properties and their values

In the step 2, we get the candidates of related authors based on the book author's properties and values from DBpedia. We use three properties (genre, influencedBy, subject) in this paper. If two authors have some common property values, we consider them as similar authors. In general, the more property values the two authors have in common, the more similar they will be. We use a SPARQL query shown in Figure 3.

In the step 3, we get the candidates of related authors' properties and their values from DBpedia. The SPARQL query in this step is the same as Figure 2 except the author's name.

In the step 4, we calculate author similarity based on the author's properties and their values. We calculate author's similarity by using Jaccard Similarity (1). We can find related authors in this process.

$$Sim\_score(A,\ B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

In the step 5, we extract related authors' book ratings from Book-Crossing [9] dataset and calculate recommendation score (2) which combines book rating value and author similarity score (1). We consider that if our recommendation score is only consisted by author similarity score, the recommendation may be full of the same author's books. Thus, we try to add another score to avoid this situation.

$$Score(book\ 1,\ A) = z(Sim\_score(A,\ B)) + z(book\ 1\ rating) \tag{2}$$

Where B is book 1's author. A is an author who has a relationship with B.

Since the scales of author similarity score (0~1) and book rating value (1~10) are different, we calculate their z-scores (3) for normalization.

$$z = \frac{x - \bar{x}}{\sigma} \tag{3}$$

where $\bar{x}$ is the mean of the sample values, σ is the standard deviation of the sample values.

Finally, we recommend books to the user based on the recommendation score.

3.2 Datasets

3.2.1 DBpedia

DBpedia [7] is one of the datasets in Linked Data, which is aiming to extract structured data from Wikipedia. As of 2018 release version, there are 64,239 books, 32,512 authors in DBpedia.

3.2.2 Book-Crossing

Book-Crossing [10] is a dataset collected from the Book-Crossing community and contains 278,858 users (anonymized but with some demographic information) providing 1,149,780 ratings about 271,379 books. There are some foreign language (non-English) books. We delete these books since our recommendation focuses on English books. We use this dataset's book ratings as a part of our recommendation score and link it with DBpedia due to there are a relatively small number of book titles available in DBpedia. In this paper, we take book ratings' average score. Furthermore, book ratings may improve the quality of our recommendation.

4 RESULTS

Table 1 and Table 2 show top results from the related authors for W. Somerset Maugham and Hermann Hesse, respectively. The related author's calculation is described in formula (1), as Step 4 in Figure 1. The recommendation results of W. Somerset Maugham can be seen in Table 3. And the

```
prefix dbpedia: <http://dbpedia.org/resource/>
prefix dbpedia-owl: <http://dbpedia.org/ontology/>
prefix dcterms: <http://purl.org/dc/terms/>
select ?author ((count(?subject)+count(?influencedBy)+count(?genre)) as ?count)
where
{
      { ?author a dbpedia-owl:Writer.
        <http://dbpedia.org/resource/Hermann_Hesse> dcterms:subject ?subject.
        ?author dcterms:subject ?subject.
        filter (?author != (<http://dbpedia.org/resource/Hermann_Hesse>))
       }
 union
       { ?author a dbpedia-owl:Writer.
         <http://dbpedia.org/resource/Hermann_Hesse> dbpedia-owl:influencedBy
         ?influencedBy.
         ?author dbpedia-owl:influencedBy ?influencedBy.
         filter (?author != (<http://dbpedia.org/resource/Hermann_Hesse>))
        }
 union
        { ?author a dbpedia-owl:Writer.
          <http://dbpedia.org/resource/Hermann_Hesse> dbpedia-owl:genre ?genre.
          ?author dbpedia-owl:genre ?genre.
          filter (?author != (<http://dbpedia.org/resource/Hermann_Hesse>))
        }
}
group by (?author)
order by desc (?count)
```

Figure 3: SPARQL query for extracting candidates of related authors for Hermann Hesse

recommendation results of Hermann Hesse can be seen in Table 4. Recommendation results consist of Book ISBN, Book Title, Book Author, Author Score, Book Rating, and Final score.

Table 1: Related authors for W. Somerset Maugham

No.	Author	Score
1	William Corlett	0.263
2	Craig Hinton	0.184
2	William Matthew Scott	0.184
4	John Roman Baker	0.159
5	P-P Hartnett	0.154
6	S. Fowler Wright	0.150
7	Patrick White	0.145
8	Frederick Rolfe	0.143
8	N. J. Crisp	0.143
8	Robert Nichols	0.143
11	Hanif Kureishi	0.140
11	Henry Treece	0.140
13	George Gissing	0.139
13	Jonathan Kemp	0.139
15	Graham Greene	0.132
15	Brigid Brophy	0.132
15	Patrick Hamilton	0.132
15	Jeremy Brooks	0.132

5 DISCUSSION

We give W. Somerset Maugham and Hermann Hesse as examples in the previous section. In this section, we discuss the results of the recommendation as follows.

Table 2: Related authors for Hermann Hesse

No.	Author	Score
1	Gerhart Hauptmann	0.160
2	Wilhelm Hauff	0.154
3	Horst Bienek	0.128
4	Friedrich Christian Delius	0.125
5	Paul Heyse	0.122
6	Helmut Heienbuttel	0.116
7	Georg Heym	0.114
8	Gustav zu Putlitz	0.109
9	Albert Knapp	0.108
9	Gunter Grass	0.108
11	Friedrich Holderlin	0.104
12	Thomas Mann	0.103
13	Alexander Kaufmann	0.100
14	Michael Kruger	0.098
14	Ludwig Pfau	0.098
14	Reinhold Schneider	0.098
17	W. G. Sebald	0.095
17	Gottfried Kinkel	0.095

Our calculation is based on the author's genre, subject, and influencedBy properties. W. Somerset Maugham was an English and playwright writer. He was also known as a short story writer. The results from Table 1 show that there are many related authors who are also English, short story and playwright writers, such as William Corlett, Patrick White, etc. In addition, Hermann Hesse was a German writer and poet, who had also received the Nobel Prize in Literature as well. In Table 2, we can find that there are many German writers and poets, who have also been awarded the Nobel Prize in Literature, such as Gerhart Haupt-

Table 3: Recommendation results of W. Somerset Maugham

No.	ISBN	Book Title	Book Author	Author Score	Book Rating	Score
1	1555835279	Two Gentlemen Sharing	William Corlett	0.263	7.667	6.257
2	1555834248	Now and Then	William Corlett	0.263	7.000	5.826
2	0743410017	The Steps up the Chimney	William Corlett	0.263	7.000	5.826
2	0743410033	The Tunnel Behind the Waterfall	William Corlett	0.263	7.000	5.826
5	0947087133	Three Uneasy Pieces	Patrick White	0.145	10.000	3.960
6	0571177387	My Beautiful Laundrette & Other Writings	Hanif Kureishi	0.140	10.000	3.793
7	0861300181	The Golden Strangers	Henry Treece	0.140	10.000	3.778
7	0861300203	Red Queen, White Queen	Henry Treece	0.140	10.000	3.778
9	0486237753	The Dead Secret	Wilkie Collins	0.128	10.000	3.395
9	0486243338	The Haunted Hotel	Wilkie Collins	0.128	10.000	3.395
9	0192821954	Basil (World's Classics)	Wilkie Collins	0.128	10.000	3.395
9	0140434119	Armadale (Penguin Classics)	Wilkie Collins	0.128	10.000	3.395
13	0140016570	The Tree of Man (Modern Classics S.)	Patrick White	0.145	8.750	3.314
14	0393974987	The Mayor of Casterbridge (Norton Critical Edition)	Thomas Hardy	0.122	10.000	3.227
14	0553212699	Return of the Native (Bantam Classic)	Thomas Hardy	0.122	10.000	3.227

AS_Mean: 0.0713 **AS_SD:**0.0310 **BR_Mean:** 7.556 **BR_SD:** 1.548

Table 4: Recommendation results of Hermann Hesse

No.	ISBN	Book Title	Book Author	Author Score	Book Rating	Score
1	0449241807	Flounder	Gunter Grass	0.108	10.000	4.113
2	0679441832	The Magic Mountain	THOMAS MANN	0.103	10.000	3.950
2	0312120028	Death in Venice: A Case Study in Contemporary Criticism (Death in Venice)	Thomas Mann	0.103	10.000	3.950
2	0679752609	Buddenbrooks: The Decline of a Family (Vintage International)	Thomas Mann	0.103	10.000	3.950
5	0679772871	The Magic Mountain	THOMAS MANN	0.103	9.333	3.519
6	0375420576	The Tin Drum	Gunter Grass	0.108	9.000	3.467
6	0151007640	Crabwalk	Gunter Grass	0.108	9.000	3.467
8	0394700864	The Transposed Heads : A Legend of India	THOMAS MANN	0.103	9.000	3.304
9	067972575X	The Tin Drum (Vintage International)	Gunter Grass	0.108	9.000	3.225
10	0375504834	Austerlitz	W. G. Sebald	0.095	9.000	2.987
11	0140187243	Billiards at Half-Past Nine (Penguin Twentieth-Century Classics)	Heinrich Boll	0.078	10.000	2.985
12	0156014165	Too Far Afield	Gunter Grass	0.108	8.000	2.821
12	0156155516	Cat and Mouse	Gunter Grass	0.108	8.000	2.821
12	015675830X	The Rat	Gunter Grass	0.108	8.000	2.821
12	0749394854	The Flounder	Gunter Grass	0.108	8.000	2.821

AS_Mean: 0.0452 **AS_SD:** 0.0253 **BR_Mean:** 7.556 **BR_SD:** 1.548

mann, Gunter Grass. Thus, we can find that the author similarity is well expressed based on these results.

However, we aware that our research has the limitation that when we tried to link 32,512 authors of DBpedia to 90,485 authors of Book-Crossing, there are only 7,452 authors in common. There are top 18 related authors for W. Somerset Maugham in Table 1, there are only four related authors appeared in Table 3. In Table 4, there are only three related authors appeared. One of the reasons is that the Book-Crossing dataset do not completely cover the books which should be appeared in our recommendation results. And there may be a mismatch problem when we link DBpedia to Book-Crossing only by author name. Another reason is the deletion of non-English books. As we have deleted non-Enlish books, it may lead to some non-English authors' books can not be appeared.

6 CONCLUSION AND FUTURE WORK

In this paper, we describe the overview of our book recommender system and methodology. We illustrate the recommendation score in detail and give some examples.

We will try to use another book dataset (such as Amazon) to link DBpedia for solving the mismatch and non-English problems. To further our research, we plan to use all of the properties of authors. And we are also considering to weight for property values. Further, we consider to complete our evaluation part. In the evaluation, we ask a user to answer questions based on the definition of serendipity with respect to Denis Kotkov [3]'s research.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Number JP17K00449.

REFERENCES

[1] Francesco, Ricci; Lior, Rokach; Bracha, Shapira: Recommender Systems Handbook, Springer US, pp. 1-35, 2015.

[2] Linked Data: https://www.w3.org/standards/semanticweb/data, (accessed 2019-3-31)

[3] Denise, Kotkov et al: "A Survey of Serendipity in Recommender Systems", Knowledge-Based Systems, Vol. 111, pp. 180-192, 2016.

[4] Oku, Kenta; Fumio, Hattori: "Fusion-based Recommender System for Improving Serendipity", Proceedings of the 1st ACM RecSys Workshop on Novelty and Diversity in Recommender Systems, pp. 19-26, 2011.

[5] Yamaba, Hisaaki, et al: "On a Serendipity-oriented Recommender System based on Folksonomy and its Evaluation", Proceedings of KES2013, pp. 276-284, 2013.

[6] Di Noia, Tommaso, et al: "Linked open data to support content-based recommender systems", Proceedings of the 8th International Conference on Semantic Systems, pp. 1-8, 2012.

[7] DBpedia: http://dbpedia.org, (accessed 2019-3-19).

[8] 一瀬詩織; 大西可奈子; 小林一郎:「Linked Data を用いた情報 推薦への取り組み」, 情報処理学会第 74 回全国大会, pp. 649-650, 2012.

[9] DBpedia SPARQL endpoint: http://dbpedia.org/sparql, (accessed 2019-4-9)

[10] Cai-Nicolas, Ziegler; Sean, M. McNee; Joseph, A. Konstan; Georg, Lausen: "Improving Recommendation Lists Through Topic Diversification", Proceedings of the 14th International World Wide Web Conference, pp. 22-32, 2005.