Vol.2020-IFAT-137 No.1
2020/2/15

Author-Oriented Book Recommendation Using Linked Open Data for Improving Serendipity

WENG RENLOU[1,a)] TAKAKU MASAO[2,b)]

Abstract: Recent years, recommender systems (RSs) are being used in many scenarios, such as online shopping stores, movie website and so on. However, many recommendation algorithms focus on accuracy based on a user profile, which may lead to reducing the user's satisfaction. This paper focuses on improving serendipity in RSs. In order to improving serendipity in book RS, two approaches are used in this paper: Linked Open Data (LOD) resource and author-oriented method. In addition, we implement our book RS and conducted a user experiment for evaluating the serendipity in book RS. We set two metrics for evaluating serendipity. As a result, the ratio of serendipitous books in top-10 list is 38.57% for author-oriented. Additionally, our method shows higher Novelty than baseline, even if Unexpectedness and Relevance are the same level with the baseline. Moreover, our method based recommendation tends to be more difficult for users to discover and much to users' surprise.

1. Introduction

1.1 Background

The recommender systems (RSs) are utilized in many fields, such as books, movies and other fields [6]. For example, the E-commerce site Amazon.com[*1] uses RS to recommend some items that consumers may like. There are many RSs focusing on the accuracy of recommendation algorithm based on user profile[4] [10]. However, they neglect the satisfaction of users. For example, if a user likes 1Q84 and has bought 1Q84 BOOK 1 in Amazon. Based on this, Amazon suggests 1Q84 BOOK 2 and 1Q84 BOOK 3 to the user. The user tends to be bored with the recommendations which he/she has known before. Moreover, these recommendations may hurt the user's satisfaction when he/she uses such RS. This is so called overspecialization problem in RS [12] [10]. There are many ways to overcome this problem (e.g., improving novelty, diversity and serendipity in RS) [4]. We primarily introduce serendipity in this paper.

1.2 Serendipity

According to Cambridge Dictionary, the word *Serendipity* means "the fact of finding interesting or valuable things by chance"[*2]. *Interesting or valuable things* denotes that an item is relevant or useful to a user. *Finding things by chance* indicates that an item is unexpected to a user. Roughly speaking, an item which has the elements of relevance and unexpectedness may be serendipitous to a user. We take back to our example: if a user likes 1Q84 written by Haruki Murakami and has bought 1Q84 BOOK 1 in Amazon.com. Based on this, Amazon.com suggests 1Q84 BOOK 2 and 1Q84 BOOK 3 to the user. The recommendations may be too obvious for the user. Because it is not difficult to imagine that he/she may buy 1Q84's other volumes. On the other hand, the recommendation is likely to be serendipitous if the RS recommend some books written by an author who is unpopular but has similar written style with Haruki Murakami. Because the books written by such author may be difficult to be found by the user considering of his/her unpopularity. Moreover, the author's books may be relevant to the user's preference considering of his/her written style.

1.3 Purpose & contribution

The purpose of this paper is to improving serendipity in book RS. We use two approaches to improve serendipity in book RSs. Firstly, we use an author-oriented method which focus on author similarity to generate book recommendation for improving serendipity. Secondly, we use Linked Open Data (LOD) resources which contain rich structured data for public to use. The main contribution of this research can be seen as follows:

(1) We used an approach in book RS, which was focusing on author relationship using LOD resource for improving serendipity.

(2) We constructed a book dataset which was consisted of LOD resource and the real-world book dataset Goodreads

[1] Graduate School of Library, Information and Media Studies, University of Tsukuba
[2] Faculty of Library, Information and Media Science, University of Tsukuba
[a)] s1821641@s.tsukuba.ac.jp
[b)] masao@slis.tsukuba.ac.jp
[*1] https://www.amazon.com/
[*2] https://dictionary.cambridge.org/dictionary/english/serendipity (accessed: 2019-12-21)

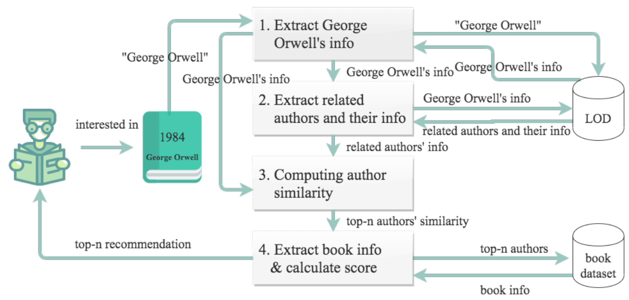

Fig. 1 System Overview

[*3].

(3) We implemented an author-oriented and content-based book RS using LOD resource.

(4) Based on Kotkov [8]'s user experiment for evaluating serendipity in RS, we designed a more thorough evaluation of serendipity in book RS.

1.4 Research questions

In this paper, we are trying to address the research questions (RQs) as follows:

RQ1. How many serendipitous books in book recommendation list while using author-oriented method?

RQ2. How many books are novel, unexpected and relevant to users in book recommendation list comparing author-oriented to content-based recommendation?

RQ3. Whether author-oriented book recommendation is better than content-based recommendation for improving serendipity or not?

2. Related Works

Oku et al. [13] proposed a fusion-based RS which selected the mixed features of two user-input items together for improving serendipity. Said et al. [14] proposed a k-furthest neighbor (kFN) algorithm which is a modification of k-nearest neighbor (kNN) [2] algorithm for improving more diverse recommendations. Zheng et al. [15] presented a serendipitous recommendation that is both unexpected and useful for users. They consider unexpected metric into two facets, which are item rareness and item dissimilarity from the user profile. Considering that items may be too unexpected from user's interest, PureSVD [3] which makes an effective performance in capturing user's future interests was applied into recommendation algorithm. Comparing to Oku et al. [13], we do not mixed item features for improving serendipity but focused on one of the features in item. For Said et al. [14] and Zheng et al. [15], we did not generate recommendation based on collaborative filtering algorithm which need a user profile {user, item, rating} but give a recommendation based on content-based algorithm which focus on item's attributes.

3. Proposed Methodology

We suppose that if a user is interested in "1984" written by George Orwell. In addition, the user is also fond of George Orwell because of his written genre, he/she may be familiar with other books written by George Orwell. If RSs recommend such books to the user, it may be obvious and not surprise for the user. On the other hand, if RSs recommend books which are written by George Orwell's similar authors who are not famous but have same written genre with George Orwell. This recommendation seems relevance and novel for the user. As the similar authors is not famous (novel) and have same written genre with George Orwell (relevance). In addition, it seems difficult for the user to find similarity authors by his/her own self, which may be unexpected for the user as the books written by the similar authors who are difficult for the user to find by his/her own self. In order to give such recommendation, we designed our system, as shown in Fig 1.

In the step 1, we hypothesis that a user is interested in George Orwell's book "1984". We extract George Orwell's information from Linked Open Data (LOD). In the step 2, we get related authors based on George Orwell's information. If authors are common in property values with George Orwell, such as same written genre with George Orwell, we consider them as related authors with George Orwell. In general, the more property values the authors have in common with George Orwell, the more similar they will be. In the step 3, we calculate author similarity using Jaccard Similarity (1) based on the information of George Orwell and his related authors. As a result, we can get top-n related authors according to author similarity score. Here we take an author A and B as an example, as shown in formula 1.

$$Sim\_score(A,\ B) = \frac{|A_{pv} \cap B_{pv}|}{|A_{pv} \cup B_{pv}|} \qquad (1)$$

where:

A_{pv} means author A's property values, B_{pv} means author B's property values.

In this paper, we use A_{pv} and B_{pv} as the set of property values. For example, if Author A's A_{pv} is {Literary, Surrealism, Magic realism, Bildungsroman} and Author B's B_{pv} is {Avantpop, Surrealism, Magic realism, Bildungsroman}. As a result, their common set is {Surrealism, Magic realism, Bildungsroman}. According to Jaccard Similarity, their similar score is 0.6. In the step 4, we extract the book ratings of top-n authors from book 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 same author's books. Thus, we try to add another score to avoid this situation.

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

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

[*3] https://www.goodreads.com/

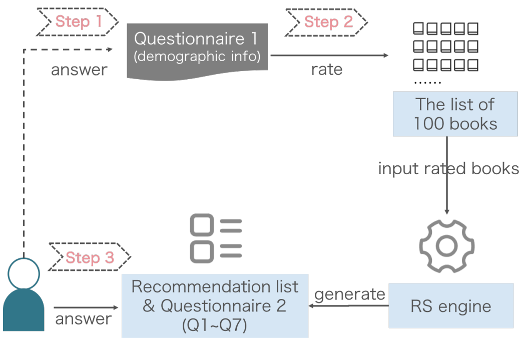

Fig. 2 The overview of user study

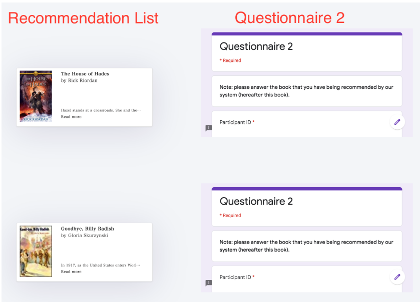

Fig. 3 The recommendation page

rating value (1∼5 or 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.

4. Evaluation

We design within-subject user study for the comparison of proposed method and baseline, as shown in Fig 2. It is designed based on [14]. In Step 1, participants are asked to answer Questionnaire 1 which consists of demographic information. In Step 2, participants are asked to rating a minimum of 5 books they have read and liked from a page showing 100 books. A recommendation list is generated based on participants' rating. In Step 3, participants are asked to answer Questionnaire 2 based on each recommendation. The recommendation page is shown in Fig 3. The recommendation list consists of author-oriented (top-10) and baseline (top-10) recommendations. In order to set the experiment under the same condition, 50% of the participants are presented with the order of recommendation lists as {baseline, author-oriented}, and 50% of the participants are presented with reverse order.

4.1 Datasets

4.1.1 DBpedia

We use DBpedia dataset considering of its rich and useful data in LOD. There are 64,239 books and 32,512 authors in DBpedia.

4.1.2 Goodreads

As there are not enough information about the books (such as book cover) in DBpedia. We consider the real-world book dataset Goodreads to fill up the information of books in DBpedia. There are 90 million users and 2.6 billion books added in Goodreads.

4.1.3 Matching of datasets

We mapped book information in DBpedia to Goodreads dataset using book ISBN as an identification. Firstly, we extracted the books which had ISBN and author name both in DBpedia. We found that there were 22,901 distinct ISBNs. After automatic mapping we found that there were 11 ISBN-based book information not correct, such as the wrongness of DBpedia author property values in DBpedia and the non-English books in Goodreads. We manually fixed them and there were 22,346 (97.58%) ISBN-based book information remained. We extracted the data from DBpedia's online version and Goodreads' online version between Oct 24, 2019 and Oct 28, 2019.

4.2 Implementations

For the matching of datasets, we extract the data from DBpedia using Apache Jena 3.12.0. We collect the resources (book title, book cover, book rating, etc.) presented in Goodreads where everyone can assess it without login[*4] We implement our system which is consisted of author-oriented and baseline methods.

4.2.1 Author-Oriented

We use properties that belonging to DBpedia ontology because of their high quality, clean and well performed data in LOD [5]. We do not use the property wikiPageExternalLink (`http://dbpedia.org/ontology/wikiPageExternalLink`) since it do not give any useful information to our recommendation. In addition, `dct:subject` is used in our computation because of its rich and useful data. Moreover, we use `skos:broader` property considering of its implicit information in LOD.

4.2.2 Baseline

Content-based RS recommend items which are similar to the target item (a given user liked in the past) based on its attribute [11]. For example, a content-based book RS may calculate similarity between books considering of their genre, publish year, book introduction, author name, etc. In this research, we set the traditional content-based RS as our baseline. We use properties in LOD as item's attribute (book's attribute), and calculate the similarity using Jaccard Similarity.

4.3 How to evaluate serendipity

4.3.1 The definition of three components

For Novelty, an item is novel to a user can be summarized as follows [10] [8] [7]: (1) The user has never heard about the item. (2) The user has heard about the item, but has not consumed it. (3) The user has consumed the item and forgot about it. For Unexpectedness, an item is unexpected to a user can be summarized as follows [10] [8] [1]: (1) The

[*4] We collect the resource for academic use only.

user does not expect this item to be relevant to them. (2) The user does not expect this item to be recommended to them. (3) The user would not have found this item on their own. (4) The item is significantly dissimilar to items the user usually consumes. For Relevance, an item is relevant to a user if the user express or will express their preference for the item in the future depending on a particular scenario [10] [8] [9].

4.3.2 Questionnaire design

We evaluate serendipity according to its three components and design experiment questionnaire based on Kotkov et al.'s research [8]. For the component of "Novelty", we set two questions responding to its first two definitions. For the component of "Unexpectedness", we set three questions responding to its last three definitions. According to Kotkov et al. [8], they set four questions with respect to the four definitions of "Unexpectedness". For the component of "Relevance", we set two questions responding to its definition. We ask each user to answer the questions using the four-scales (Strongly Agree, Agree, Disagree, Strongly Disagree).

4.3.3 Evaluation metrics

We define that a book which is serendipitous for a user should meet the formula as follows:

$$Serendipity_{binary} = Novelty \cap Unexpectedness \cap Relevance \tag{4}$$

where:

$$Serendipity_{binary} = \begin{cases} 1 & \text{if an item is serendipitous;} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Here, $Novelty$ means that an item is novel to a user when the user answered the Q1 or Q2 at least 3 (Agree). $Unexpectedness$ denotes that an item is unexpected to a user when the user answered the Q3, Q4 or Q5 at least 3 (Agree). $Relevance$ denotes that an item is relevant to a user when the user answered the Q6 or Q7 at least 3 (Agree). In this formula, $Novelty$, $Unexpectedness$ and $Relevance$ are binary variables.

We set another serendipity metric which can be seen as follows:

$$Serendipity_{graded} = \frac{Nov_{in} + Unex_{in} + Rel_{in}}{Max(questionnaire\_scaling) \times n} \tag{6}$$

where:

$$Serendipity_{graded} = \begin{cases} > 0 & \text{if } Nov_{in} * Unex_{in} * Rel_{in} \neq 0; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

$Serendipity_{graded}$ is the serendipitous intention of an item for a user. Here, Nov_{in} is considered as the sum value of Q1 and Q2 answers or 0. $Unex_{in}$ is considered as the sum value of Q3, Q4 and Q5 answers or 0. In the same way, Rel_{in} is considered as the sum value of Q6 and Q7 or 0. $Max(questionnaire\_scaling)$ denotes the max scaling of our questions. n is the number of questions.

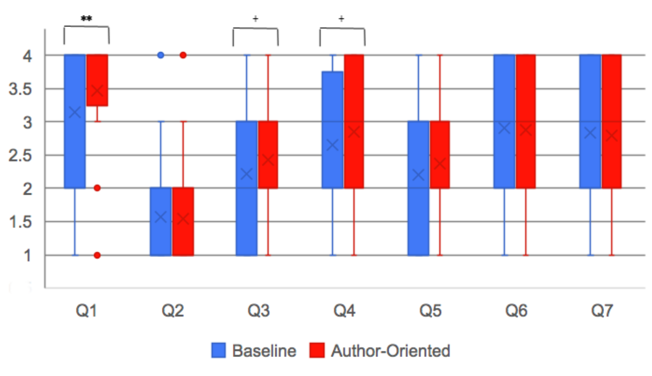

Fig. 4 The distribution of questions (Q1∼Q7) in Questionnaire 2. Significance codes: "**": $p \leq 0.01$, "+": $p \leq 0.1$

4.4 User Experiment

We conduct a user experiment for verifying whether our proposed method give an effective performance or not comparing to baseline. We recruited 14 participants who often usually English books. The age of participants is between 18 and 39 years old. 71.4% of them are females and 28.6% are males. The experiment was conducted in the library of T University from Nov 22, 2019 to Dec 11, 2019.

5. Results

Fig 4 demonstrates the distribution of questions (Q1∼Q7) in Questionnaire 2 comparing author-oriented to baseline. For Q1, the mean rating of baseline is 3.14. The mean rating of author-oriented is 3.47. There is a significant difference (p=0.01). For Q2, the mean rating of baseline is 1.57. The mean rating of author-oriented is 1.54. There is no significant difference (p=0.75). For Q3, the mean rating of baseline is 2.22. The mean rating of author-oriented is 2.44. There is a marginally significant difference (p=0.06) between baseline and author-oriented. For Q4, the mean rating of baseline is 2.65. The mean rating of author-oriented is 2.86. There is a marginally significant difference (p=0.06) between baseline and author-oriented. For Q5, the mean rating of baseline is 2.20. The mean rating of author-oriented is 2.38. There is no significant difference (p=0.13). For Q6, the mean rating of baseline is 2.91. The mean rating of author-oriented is 2.88. There is no significant difference (p=0.80). For Q7, the mean rating of baseline is 2.84. The mean rating of author-oriented is 2.80. There is no significant difference (p=0.76).

In Fig 5, we demonstrates the mean of serendipity's three components based on $Serendipity_{binary}$. For the $Novelty$ of baseline, its mean is 0.92. For the $Novelty$ of author-oriented, its mean is 0.97. There is a significant difference (p=0.05) between baseline and author-oriented. For the $Unexpectedness$ of baseline, its mean is 0.69. For the $Unexpectedness$ of author-oriented, its mean is 0.72. There is no significant difference (p=0.46). For the $Relevance$ of baseline, its mean is 0.66. For the $Relevance$ of author-oriented, its mean is 0.65. There is no significant difference (p=0.90). For the $Serendipity_{binary}$ of baseline, its mean

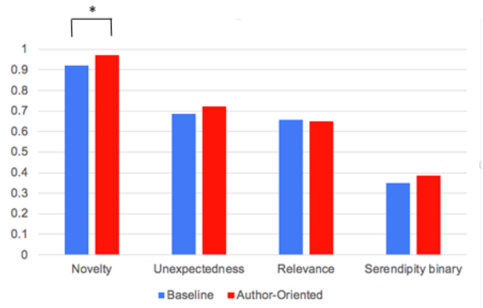

Fig. 5 The mean of serendipity's three components ($Serendipity_{binary}$). Significance codes: "*": p$\leq$ 0.05

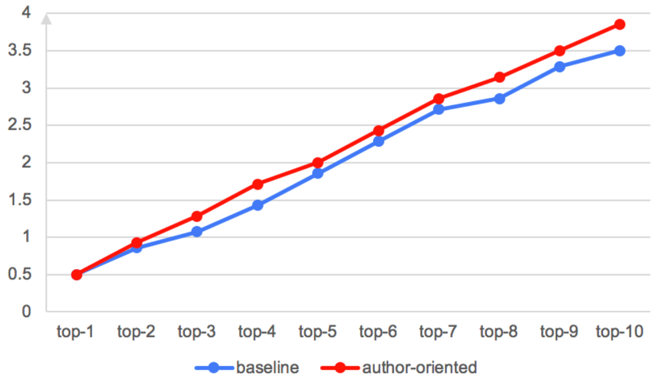

Fig. 6 The mean of serendipitous books in top-n recommendation list according to $Serendipity_{binary}$

is 0.35. For the $Serendipity_{binary}$ of author-oriented, its mean is 0.39. There is no significant difference (p=0.49). Fig 6 demonstrates the mean of serendipitous books in top-n recommendation list according to $Serendipity_{binary}$ comparing baseline to author-oriented. For baseline, the mean of serendipitous books in top-10 recommendation list is 3.50. For author-oriented, the mean of serendipitous books in top-10 recommendation list is 3.86. There is no significant difference (p>0.05).

For the $Serendipity_{graded}$ of baseline, its mean is 0.24. For the $Serendipity_{graded}$ of author-oriented, its mean is 0.28. Fig 7 demonstrates the mean of serendipity's three components based on $Serendipity_{graded}$. For the $Novelty$ of baseline, its mean is 4.55. For the $Novelty$ of author-oriented, its mean is 4.91. There is a significant difference (p=0.01) between baseline and author-oriented. For the $Unexpectedness$ of baseline, its mean is 5.61. For the $Unexpectedness$ of author-oriented, its mean is 6.34. There is no significant difference (p=0.11). For the $Relevance$ of baseline, its mean is 4.60. For the $Relevance$ of author-oriented, its mean is 4.55. There is no significant difference (p=0.90). According to $Serendipity_{graded}$, the mean of serendipitous intention in top-n recommendation list comparing author-oriented to baseline is no significant difference as well.

6. Discussion

RQ1. How many serendipitous books in book recommendation list while using author-oriented method?

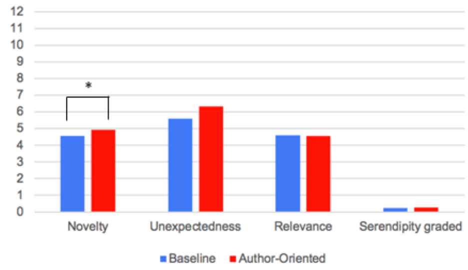

Fig. 7 The mean of serendipity's three components ($Serendipity_{graded}$). Significance codes: "*": p $\leq$ 0.05

The ratio of serendipitous items in book recommendation list is 54 while using author-oriented method. On the other hand, the ratio of serendipitous items in book recommendation list is 49 while using baseline method. As a result, 38.57% books are serendipitous to our participants in a top-10 book recommendation list while using author-oriented method.

RQ2. How many books are novel, unexpected and relevant to users in book recommendation list comparing author-oriented to content-based recommendation?

According to the definitions of three elements (novelty, unexpectedness, relevance) shown in section 4.3.3, we compared author-oriented to baseline. For `Novelty`, author-oriented is 97.14% and content-based is 92.14%. There is a significant difference between them (p=0.05). For `Unexpectedness`, author-oriented is 72.14% and content-based is 68.57%. There is no significant difference between them (p=0.46). For `Relevance`, author-oriented is 65.00% and content-based is 65.71%. There is no significicant difference between them (p=0.90). It indicates that generating recommendations based on indirect relationship (author-oriented) are more novel than using direct relationship (baseline).

RQ3. Whether author-oriented book recommendation is better than content-based recommendation for improving serendipity or not?

In this paper, we set two metrics to evaluate serendipity. For $Serendipity_{binary}$, it regards whether an item is serendipitous to a user or not as 0 or 1. For $Serendipity_{graded}$, it regards the serendipitous intention of an item for a user. According to $Serendipity_{binary}$, the mean of serendipitous books in top-10 recommendation list is 3.50 for baseline. For author-oriented, the mean of serendipitous books in top-10 recommendation list is 3.86. There is no significant difference (p>.05) between baseline and author-oriented. According to $Serendipity_{graded}$, the mean of serendipitous intention in top-n recommendation list is 2.39 for baseline. For author-oriented, the mean of serendipitous intention in top-n recommendation list is 2.80. There is no significant difference (p>.05) between baseline and author-oriented.

6.1 Implications

According to the answer of RQ2, we can find that author-oriented recommendations are more novel to participants comparing to baseline, which supports that using indirect relationship (author-oriented) is more novel than using direct relationship (content-based). For Unexpectedness, the results of Q3 (*I was surprised (not expected) that this system recommend this book to me*) and Q4 (*This is the type of book I would not normally discover on my own. For example, I need a recommender system like this system to find books like this one*) indicate that author-oriented is greater than baseline. This might suggest that books recommended by our method are more difficult for a user to discover by his/her own self comparing to baseline according to Q4. Moreover, author-oriented recommendation gives more surprise to a user comparing to baseline according to Q3.

6.2 Limitations & Future work

There are several limitations in our research. Firstly, none of our participant is native in English, which may have an influence on the decision of questionnaire answers. For the number of participants, we only recruited 14 participants in our experiment. Secondly, a few of books have no description and no book cover, which may influence the decision of questionnaire answer according to user feedback. Thirdly, we did not construct a traditional book RS (not using LOD resource) to compare the effectiveness of LOD-based RS. We cannot judge that whether LOD resource is effective in book RS or not. Our future study will focus on two aspects. For user experiment, we mainly try to recruit some participants who are English native speakers and increase the scale of participants. For RS improvement, we will construct a traditional book RS to do a comparison.

7. Conclusion

In this paper, we use two approaches to improving serendipity in book RS: author-oriented method and LOD resource. In order to evaluate our method, we implemented our book RS and conducted a user experiment. We recruited 14 participants in our user experiment. Our book RS regarded 25,152 books in total and content-based book RS was set as a baseline for comparison. In our book RS, we generated the recommendation list consisting of author-oriented and baseline to each participant based on their rating on the books they have read or want to read. We asked them to answer the questionnaire which was designed by the definitions of serendipity's three components. In addition, we set two metrics to evaluate that whether a book is serendipitous to a user or not based on user responses.

As a result, our proposed method shows an effective performance for improving serendipity in book RSs on both of our metrics, but comparing to baseline method there is no significant difference with baseline. Although there is no significant difference for Unexpectedness and Relevance, our proposed method recommendation is more novel to a user comparing to baseline. It indicates that our method is helpful because the method shows higher Novelty, even if Unexpectedness and Relevance are the same level with the baseline. Moreover, proposed method recommendation is more difficult for a user to discover by his/her own self comparing to baseline.

References

[1] Panagiotis Adamopoulos and Alexander Tuzhilin. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):54, 2015.

[2] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[4] Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto. An investigation on the serendipity problem in recommender systems. *Information Processing & Management*, 51(5):695–717, 2015.

[5] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 1–8, New York, NY, USA, 2012. ACM.

[6] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[7] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A Konstan, and Paul Schrater. I like to explore sometimes: Adapting to dynamic user novelty preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 19–26. ACM, 2015.

[8] Denis Kotkov, Joseph A Konstan, Qian Zhao, and Jari Veijalainen. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1341–1350. ACM, 2018.

[9] Denis Kotkov, Jari Veijalainen, and Shuaiqiang Wang. A serendipity-oriented greedy algorithm for recommendations. In *WEBIST 2017: Proceedings of the 13rd International conference on web information systems and technologies*, volume 1. SCITEPRESS Science And Technology Publications, 2017.

[10] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192, 2016.

[11] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[12] Sean M McNee, John Riedl, and Joseph A Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.

[13] Kenta Oku and Fumio Hattori. Fusion-based recommender system for improving serendipity. *DiveRS@ RecSys*, 816:19–26, 2011.

[14] Alan Said, Ben Fields, Brijnesh J. Jain, and Sahin Albayrak. User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 1399–1408, New York, NY, USA, 2013. ACM.

[15] Qianru Zheng, Chi-Kong Chan, and Horace HS Ip. An unexpectedness-augmented utility model for making serendipitous recommendation. In *Industrial conference on data mining*, pages 216–230. Springer, 2015.