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

WENG RENLOU

Graduate School of Library, Information and Media Studies

University of Tsukuba

March 2020

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The Internet allows us to access a large amount of information. However, it is hard for people to select the information they actually want to access because of the enormous information. Thus, it is important for people to filter unwanted information. To achieve this, recommender systems (RSs) are increasingly used in recent years. The RSs are utilized in many fields, such as books, movies, and other fields [1]. For example, the E-commerce site *Amazon.com* [2] uses RS to recommend some items that consumers may like. There are many RSs focusing on the accuracy of the recommendation algorithm based on user profile[3] [4]. However, they neglect the satisfaction of users. For example, if a user likes 1Q84 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 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 [5] [4]. We illustrate how to address the overspecialization problem in RS according to the previous studies.

There are many ways to overcome this problem (e.g., improving novelty, diversity, and serendipity in RS) [3]. We primarily introduce serendipity in this thesis.

1.2 Serendipity

According to the Cambridge Dictionary, the word *Serendipity* means "the fact of finding interesting or valuable things by chance" [6]. *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 that 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 other volumes of 1Q84.

On the other hand, the recommendation is likely to be serendipitous if the RS recommends some books written by an author who is unpopular but has a similar written style with Haruki Murakami. Because the books written by such an author may be difficult to be

discovered by the user considering his/her unpopularity. Moreover, the books of such an author may be relevant to the user's preference considering his/her written style.

1.3 Purpose & contribution

The purpose of this thesis 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 focuses on author similarity to generate book recommendation for improving serendipity. As we have illustrated the example of serendipitous recommendation in section 1.2, we can hypothesis that the recommendation based on similar authors may be more serendipitous than the recommendation based on similar books in that situation. Thus, we try to generate book recommendations based on the relationship between authors.

Secondly, we use Linked Open Data (LOD) resources which contain rich structured data for the public to use. The reason we use LOD resources for improving serendipity considering their semantic data. For instance, it may be difficult for a user to know/discover that who influenced Haruki Murakami if he/she does not know Haruki Murakami in detail. However, we can easily extract such data and use such author relationship in LOD. In this situation, the similar authors calculated in LOD may be difficult for a user to discover.

The main contribution of this research can be seen as follows:

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

(2) We constructed a book dataset which was consisted of LOD resource and the real-world book dataset Goodreads [7].

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

(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 thesis, we are trying to address these research questions (RQs) as follows:

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

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

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

RQ4. How about the user satisfaction of author-oriented recommendation list comparing to content-based recommendation list?

1.5 The structure of this thesis

The thesis is organized as follows: in Chapter 2, we review previous studies. In Chapter 3, we introduce our methodology in detail. In Chapter 4, we illustrate the evaluation of our

method and show the results we got from our user experiment. In Chapter 5, we discuss our results. Finally, we give a conclusion in Chapter 6.

Chapter 2

Related Works

In this section, we illustrate related works from three perspectives. Firstly, we focus on serendipity and explain its three components in detail. Secondly, we review the serendipity in RSs. Lastly, we introduce the LOD resource used in RSs.

2.1 Serendipity

2.1.1 Definition

Serendipity is a difficult concept to learn as it includes an emotional dimension [9]. There is no consensus on the definition of serendipity in RSs [4] [8]. However, serendipity contains three components according to [4] [8]. An item that is novel, unexpected and relevant to a user can be considered as a serendipitous item referring to Kotkov [4]'s survey research. It means that a serendipitous item should contain three components at the same time (Novelty $\cap$ Unexpectedness $\cap$ Relevance).

In this research, we do not directly illustrate the serendipity but try to explain it from its three components. In section 2.1.2, we illustrate the details of these three components (novelty, unexpectedness, and relevance) based on previous studies.

2.1.2 Components

In this section, we illustrate the definitions of three components based on previous studies. There are multiple definitions for each component. If an item meets one of the definitions in each component. We consider it as a serendipitous item for a user.

Novelty

An item is novel to a user can be summarized as follows [4] [8] [10]:

(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.

Unexpectedness

An item is unexpected to a user can be summarized as follows [4] [8] [11]:

(1) The 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.

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 [4] [8] [12].

2.2 Serendipity in RSs

In this section, we review the previous studies from two aspects. Firstly, we introduce what approaches the previous studies used for improving serendipity in RSs. Secondly, we review how did the previous studies evaluate serendipity in RSs.

2.2.1 Improving serendipity in RSs

Oku et al. [13] proposed a fusion-based RS that 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) [15] algorithm for improving more diverse recommendations.

Zheng et al. [16] presented a serendipitous recommendation that is both unexpected and useful for users. They consider the 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 [17] which makes an effective performance in capturing user's future interests was applied into recommendation algorithm.

The differences between the previous studies and our method

Comparing to Oku et al. [13], we did not mix item features for improving serendipity but focused on one of the features in an item. Besides, LOD resource was used for improving serendipity.

Comparing to Said et al. [14] and Zheng et al. [16], we did not generate a recommendation based on collaborative filtering algorithm which needs a user profile {user, item, rating} but give a recommendation based on a content-based algorithm which focuses on item's attributes.

2.2.2 Evaluation of serendipity in RSs

From previous studies, there are mainly two methods for measuring serendipity in RS, which are online and offline. A real user interaction with RS is needed in online evaluation. For offline evaluation, the researchers set a metric and evaluate the recommendation based on prepared dataset which contains user's rating data. We introduce online evaluation and offline evaluation previous studies as follows:

Online

Based on Oku et al. [13], Oku et al. [18] conducted a user experiment for evaluating serendipitous items by asking experiment participants questionnaires based on two definitions of the serendipity. The questions can be seen in Table 2.1.

Table 2.1: Oku et all. [18] experiment questions for recommended books

No.	Question
Q1	I did not know this book.
Q2	I have been interested in this book before the system presented it to me.
Q3	This book excited my interest for the first time.
Q4	I think that I could not find this book by myself

Said et al. [14] conducted an online user study to measure user satisfaction and other metrics (including serendipity) in a recommendation. However, it is questionable that Said et al. [14] measure serendipity only setting one question (*Are the recommendations serendipitous?*), since *serendipity* is a difficult word and concept to learn even for native English speakers [19] [20].

Kotkov et al. [8] measured serendipity based on three components (Novelty, Unexpectedness, Relevance). They conducted a user experiment and asked participants questionnaires which are referring to the definitions of three components. They defined that if an item contained three components, it was considered as a serendipitous item.

Offline

Zheng et al. [16] presented a "serendipity" metric which consists of two factors, "unexpectedness" and "usefulness". They defined that if an item was not included in the set of recommendations generated by the Predictive Model (PM), it was considered an unexpected item. Here PM is constructed by consisting of a set of items that are assumed to be expected for the users. Additionally, the "usefulness" metric was considered as a user's rating.

Kotkov et al. [12] proposed a modification based on Zheng et al.'s "serendipity" metric [16]. They consider that an item is unexpected to a user if the item has at least one new feature for the user. For example, a feature that a user not rated yet is unexpected to the user.

Murakami et al. [21] proposed an "unexpectedness" metric for measuring the serendipity of recommendation lists. They consider that "unexpectedness" is the deviation from the results produced by a primitive prediction method (PPM). Besides, if the item is related to the user's preference, it is considered as an unexpected item. They also proposed another "unexpectedness" metric considering the recommendation list ranking.

2.3 LOD in RSs

Di Noia et al. [22] proposed a recommendation method that used LOD to calculate the similarity between movies based on movies' properties (e.g., the director, the genre, the starring, etc.) in DBpedia [23].

Di Noia et al. used a direct relationship to give a recommendation. We do not calculate the direct similarity of items but consider a recommendation from an indirect perspective for improving serendipity.

Ichise et al. [24] calculated 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 thesis, we do not use a direct relationship between authors but trying to calculate similar authors considering their common property values. Besides, we used all of the properties in DBpedia ontology [23] to calculate the author similarity.

Chapter 3

Proposed Methodology

3.1 System Overview

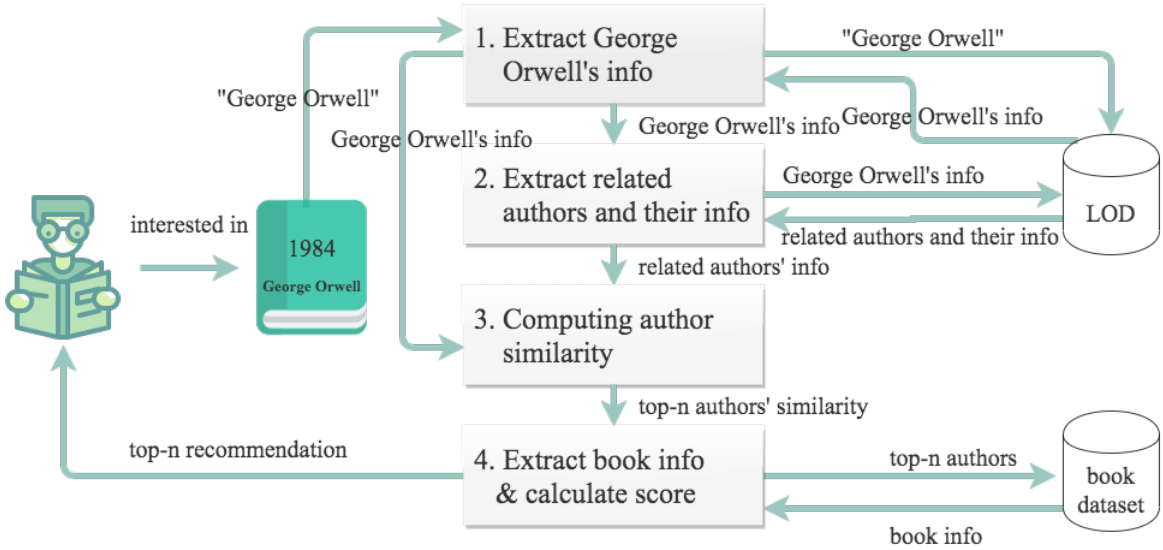

Figure 3.1: Methodology Overview

We suppose that if a user is interested in "1984" written by George Orwell. Besides, 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). Besides, it seems difficult for the user to find similarity authors by his/her 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 self.

To give such recommendation, we designed our system, as shown in Fig 3.1. Firstly, we extract the information of George Orwell from LOD. Secondly, we extract George Orwell's related authors and their information from LOD based on George Orwell's information.

Thirdly, we computing author similarity based on George Orwell and his related authors' information. Then, we extract the information of top-n authors' books from book dataset and calculate recommendation score. Finally, we recommend top-n books to user based on recommendation score.

The details in this overview can been seen as follows (here we take the book "1984" written by George Orwell as an example). Section 3.2 illustrates how to extract author information in LOD. In section 3.3, we explain how to extract the target author's (George Orwell's) related authors and their information. Then we describe how to compute author similarity in section 3.4. Finally, we illustrate how to calculate recommendation score in section 3.5

3.2 Extraction of author information

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 1. The reason why we extract information from LOD is that rich structured data are referring to a variety of data in LOD [22], such as author's genre, influencedBy, award and so on. Here we take DBpedia [23] which is one of the datasets in LOD as an example. Using DBpedia SPARQL endpoint [25], we exact properties in LOD. Fig 3.2 shows the SPARQL query for extracting author's properties in DBpedia ontology. The result of Fig 3.2 can be seen in Fig 3.3. The left column shows the author properties in DBpedia. The right column shows the count of these properties in DBpedia.

```
PREFIX  dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT  ?author_property (COUNT(?author_property) AS ?count)
WHERE
  { ?author  a                      dbpedia-owl:Writer ;
             ?author_property       ?author_info
    FILTER regex(?author_property, "http://dbpedia.org/ontology/")
  }
GROUP BY ?author_property
ORDER BY DESC(?count)
```

Figure 3.2: SPARQL query for extracting properties in LOD (taking DBpedia as an example)

Using DBpedia SPARQL endpoint [25], we can easily extract George Orwell's information from LOD, as shown in Fig 3.4. A part of results in Fig 3.4 can be seen in Fig 3.5. The left column in Fig 3.5 shows George Orwell's properties, and the right column denotes George Orwell's information corresponding to the left column.

3.3 Extraction of related authors and their information

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. The SPARQL query can be seen in Fig 3.6. It extracts George Orwell's most related author and his/her information from LOD according to George Orwell's information. A part of results can be seen in Fig 3.7.

author_property	count
http://dbpedia.org/ontology/wikiPageExternalLink	92521
http://dbpedia.org/ontology/abstract	81427
http://dbpedia.org/ontology/birthDate	41185
http://dbpedia.org/ontology/birthPlace	37646
http://dbpedia.org/ontology/wikiPageID	32255
http://dbpedia.org/ontology/wikiPageRevisionID	32255
http://dbpedia.org/ontology/deathDate	21790
http://dbpedia.org/ontology/deathPlace	15784
http://dbpedia.org/ontology/occupation	14386
http://dbpedia.org/ontology/thumbnail	12260
http://dbpedia.org/ontology/genre	11768
http://dbpedia.org/ontology/nationality	10050
http://dbpedia.org/ontology/influencedBy	8255
http://dbpedia.org/ontology/notableWork	6815
http://dbpedia.org/ontology/country	6463
http://dbpedia.org/ontology/birthName	6038
http://dbpedia.org/ontology/almaMater	5843
http://dbpedia.org/ontology/activeYearsStartYear	5216
http://dbpedia.org/ontology/award	3858

Figure 3.3: A part of author properties in LOD (taking DBpedia as an example)

```
SELECT  ?author_property ?author_info
WHERE
  { <http://dbpedia.org/resource/George_Orwell>
              ?author_property  ?author_info
  }
```

Figure 3.4: SPARQL query for extracting George Orwell's info

3.4 Computation of author similarity

In the step 3, we calculate author similarity using Jaccard Similarity (3.1) 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 3.1.

$$Sim\_score(A,\ B) = \frac{|A_{pv} \cap B_{pv}|}{|A_{pv} \cup B_{pv}|} \tag{3.1}$$

where:

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

author_property	author_info
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#NaturalPerson
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q215627
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q24229398
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q36180
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q5
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Agent

Figure 3.5: A part of results on George Orwell's properties and values from Fig 3.4

```
PREFIX  dbpedia: <http://dbpedia.org/resource/>

SELECT  ?related_author ?related_author_info
WHERE
  { { SELECT  (?author AS ?related_author)
      WHERE
        { dbpedia:George_Orwell
                    ?author_property  ?author_info .
          ?author   ?author_property  ?author_info
          FILTER ( ?author != dbpedia:George_Orwell )
        }
      GROUP BY ?author
      ORDER BY DESC(COUNT(?author_info))
      LIMIT   1
    }
    ?related_author
                ?related_author_property  ?related_author_info
  }
LIMIT   10
```

Figure 3.6: SPARQL query for extracting George Orwell's related authors and their info

related_author	related_author_info
http://dbpedia.org/resource/Stephanie_White	http://www.w3.org/2002/07/owl#Thing
http://dbpedia.org/resource/Stephanie_White	http://xmlns.com/foaf/0.1/Person
http://dbpedia.org/resource/Stephanie_White	http://dbpedia.org/ontology/Person
http://dbpedia.org/resource/Stephanie_White	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent
http://dbpedia.org/resource/Stephanie_White	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#NaturalPerson
http://dbpedia.org/resource/Stephanie_White	http://www.wikidata.org/entity/Q215627
http://dbpedia.org/resource/Stephanie_White	http://www.wikidata.org/entity/Q24229398
http://dbpedia.org/resource/Stephanie_White	http://www.wikidata.org/entity/Q3665646
http://dbpedia.org/resource/Stephanie_White	http://www.wikidata.org/entity/Q5
http://dbpedia.org/resource/Stephanie_White	http://dbpedia.org/ontology/Agent

Figure 3.7: A part of results on George Orwell's related authors and their properties and values from Fig 3.6

In this thesis, 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, as shown in Fig 3.8.

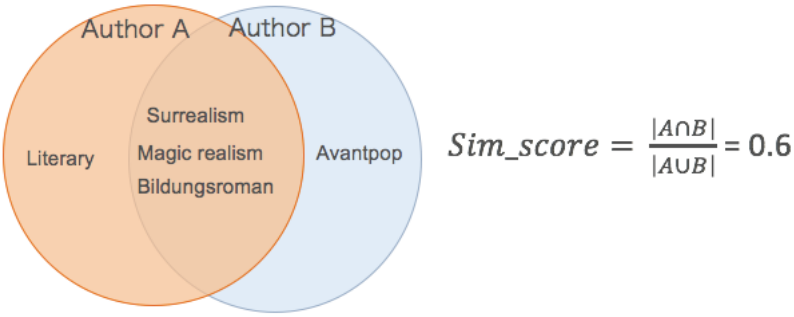

Figure 3.8: An example of calculating author similarity

3.5 Extraction of book information & calculation of recommendation score

In the step 4, we extract the book ratings of top-n authors from book dataset and calculate recommendation score (3.2) which combines book rating value and author similarity score (3.1). 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.

$$Score(book\ 1,\ A) = z(Sim\_score(A,\ B))\ + z(book\ 1\ rating) \tag{3.2}$$

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∼5 or 1∼10) are different, we calculate their z-scores (3.3) for normalization.

$$z\ =\ \frac{x - \bar{x}}{\sigma} \tag{3.3}$$

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.

Chapter 4

Evaluation

We illustrate our evaluation design in section 4.1. Moreover, the datasets we used in our system can be explained in section 4.2. In section 4.3, we illustrate the details of our system implementation. In section 4.4, we explain how to evaluate serendipity in our research. In section 4.5, we describe our user experiment in details. Finally, we show results from user experiment in section 4.6.

4.1 Design

We designed within-subject user experiment between proposed RS and baseline RS. To evaluate the quality of our proposed method, the datasets DBpedia [23] and Goodreads [7] are used in our evaluation part. As the serendipity in RS is difficult to measure and simulate by machine, a user study experiment is conducted.

The experiment consists of four steps (Fig4.1). It is designed based on [14].

Firstly, participants are asked to answer Questionnaire 1 which consists of some questions about participant's reading habit and personal information, as shown in Table 4.1. In Questionnaire 1, Q1 and Q2 denote the sex and age group of participants. Q3 means the types of books that participants often read most. Q4 means the reading frequency of participants. Finally, Q5 means the familiarity of RS for participants. To easily control the information of participants, we also embed the participant ID in this questionnaire.

Table 4.1: The questions of Questionnaire 1

No.	Question content
Q1	What is your gender?
Q2	Please select your age group.
Q3	Which of the following types of books do you read most?
Q4	How often do you read paper books or eBooks when it is your choice (not assigned in school) ?
Q5	Have you heard of book recommender system?

Secondly, participants were asked to rating a minimum of 5 books they have read and

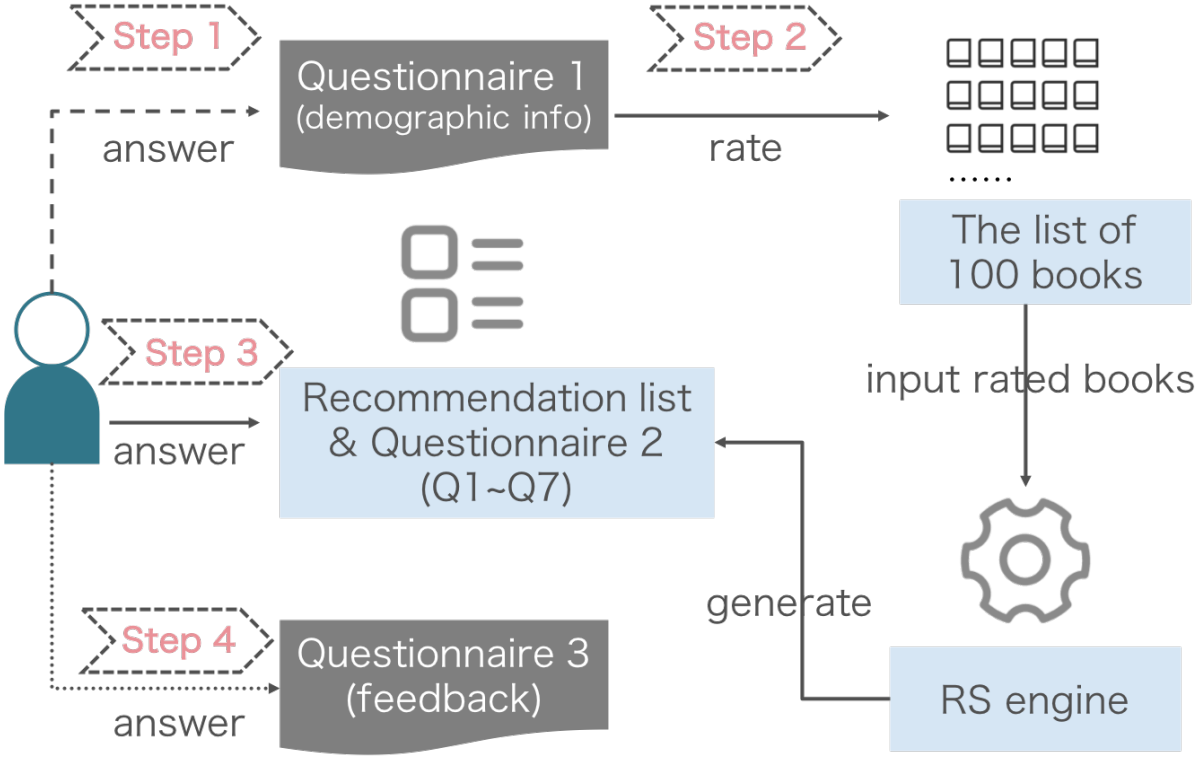

Figure 4.1: The Experiment Overview

liked from a page showing 100 books. From the result of our preliminary experiment, it was difficult for participants to select books they had read and liked at least 10. Thus we narrow down the scope to 5, which is different from [14]. Moreover, in case that there are no books or less than 5 books that the participants have read and liked, they are asked to select the books which they want to read and rate them. To show the books that the participants would be most familiar with [14], 100 books are selected from Goodreads popular shelf which exist in LOD as well.

Thirdly, having rated at a minimum of 5 books, a recommendation list is generated based on participant's rating. Then, participants are asked to answer Questionnaire 2 based on the recommendation list. There are 20 books in the recommendation list, which consists of author-oriented (top-10) and baseline-based (top-10) recommendation. To set the experiment under the same condition, 50% of the participants are presented with the order of recommendation lists as {baseline-based, author-oriented}, and 50% of the participants are presented with the order of recommendation lists as {author-oriented, baseline-based}.

At last, participants are asked to answer Questionnaire 3 for the comment and feedback of this experiment.

4.2 Dataset

DBpedia [23] and Goodreads [7] were used in our RS. We describe DBpedia in section 4.2.1 and explain Goodreads in section 4.2.2.

4.2.1 DBpedia

DBpedia [23] is one of the famous datasets in Linked Data, and its data is extracted from Wikipedia.

In this research, we use DBpedia dataset considering of its rich and useful data in LOD. Using DBpedia SPARQL queries in Fig 4.2 and Fig 4.3. Here, Fig 4.2 demonstrates the SPARQL Query for extracting book resources in DBpedia. Fig 4.3 demonstrates the SPARQL Query for extracting author resources in DBpedia. We find that there are 64,239 books and 32,512 authors in DBpedia.

```
PREFIX  dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT  (COUNT(?book) AS ?count)
FROM <http://dbpedia.org/>
WHERE
  { ?book  a                    dbpedia-owl:Book }
ORDER BY DESC(?count)
```

Figure 4.2: The SPARQL Query for extracting book resources in DBpedia (accessed in Oct 20, 2019)

```
PREFIX  dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT  (COUNT(?author) AS ?count)
FROM <http://dbpedia.org/>
WHERE
  { ?author  a                    dbpedia-owl:Writer }
ORDER BY DESC(?count)
```

Figure 4.3: The SPARQL Query for extracting author resources in DBpedia (accessed in Oct 20, 2019)

4.2.2 Goodreads

As there are not enough information about the books (such as book cover, book introduction and book rating) in DBpedia. We match another book dataset to fill up the information of books in DBpedia. We consider the real-world book dataset Goodreads [7], which is one of the biggest book website in the world. There are 90 million users and 2.6 billion books added in Goodreads, as shown at the "About Us" page of Goodreads [26].

4.2.3 Matching of datasets

In this section, we introduce the matching of datasets (DBpedia to Goodreads).

We collect the resources (book title, book cover, book rating, book introduction and author name, etc.) presented in Goodreads [7] where everyone can assess it without login[1], as shown in Fig 4.4 and Fig 4.5.

As there are a lot of books do not have International Standard Book Number (ISBN) in DBpedia. We tried to map book information in DBpedia to Goodreads via book title and book author name as an identification. However, we did not map successfully because

[1] We collect the resource for academic use only.

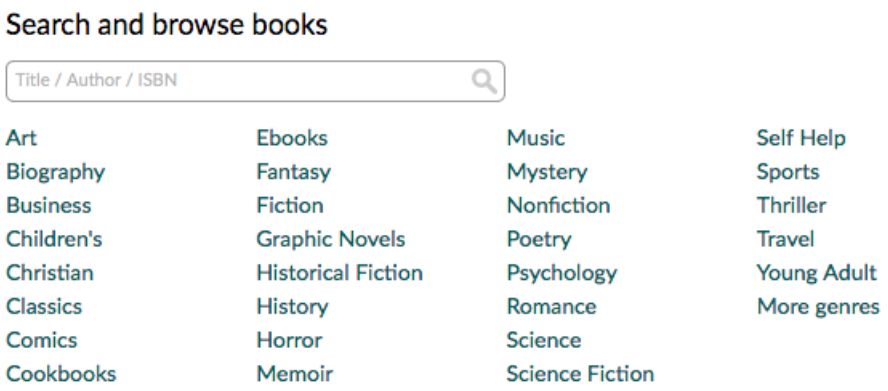

Figure 4.4: Book info retrieval page in Goodreads [7]

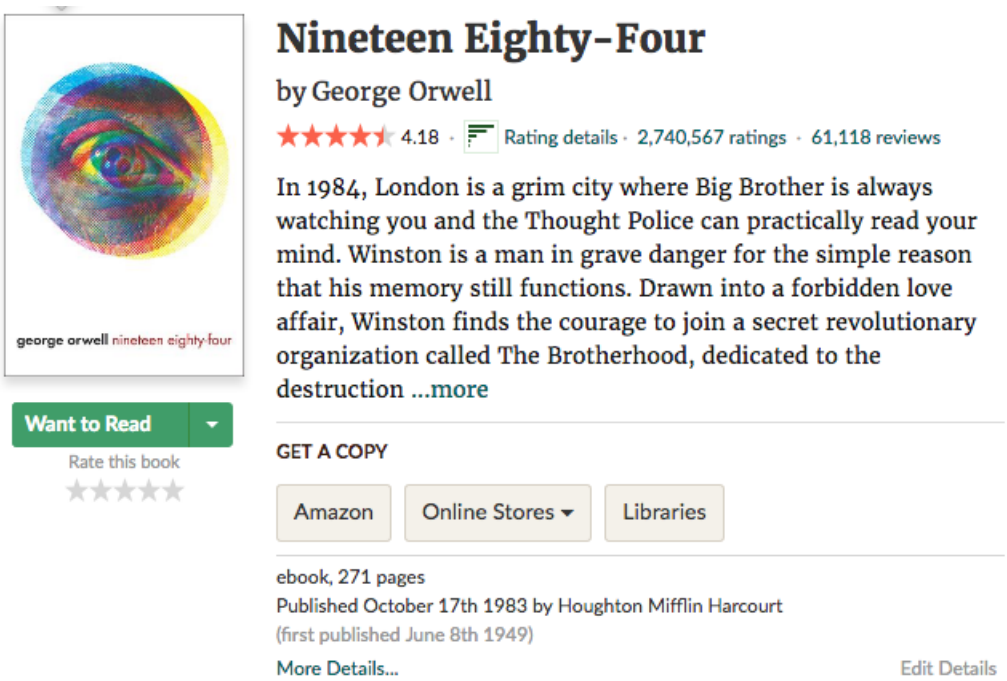

Figure 4.5: The info of "1984" in Goodreads [7]

book titles and author names are presented differently between DBpedia and Goodreads. Moreover, it took us time to check books with the same book titles or same author names but different books. As a result, 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. As some books have multiple ISBNs and multiple authors in DBpedia, there are 25,791 {ISBN, bookTitle, authorName} pairs extracted from DBpedia using the query in Fig 4.6. We found that there were 22,901 distinct ISBNs from 25,791 {ISBN, bookTitle, authorName} pairs.

Secondly, we extracted book information from Goodreads based on these 22,901 distinct ISBNs in DBpedia. The book information we extracted is book title, book author name,

the URL of book cover, book description, book rating and book id in Goodreads. As a result, 22,357 (97.62%) ISBN-based book information were successfully extracted.

After automatic mapping we found that there are 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 are 22,346 (97.58%) ISBN-based book information remained. We considered that there were some books had multiple ISBNs and authors. We identified 22,346 (97.58%) ISBN-based book information into the unique number of book titles, authors and {ISBN, bookTitle, authorName} pairs. As a result, we found that there are 22,168 distinct book titles and 10,485 distinct authors remained from 25,152 {ISBN, bookTitle, authorName} pairs. The number of each distinct item can be seen in Table 4.2. We extracted the data from DBpedia's online version and Goodreads' online version between Oct 24, 2019 and Oct 28, 2019.

```
PREFIX  dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT DISTINCT  *
FROM <http://dbpedia.org>
WHERE
  { ?bookTitle  a                  dbpedia-owl:Book ;
              dbpedia-owl:author    ?authorName ;
              dbpedia-owl:isbn      ?ISBN
  }
```

Figure 4.6: SPARQL query for extracting {ISBN, bookTitle, authorName} pairs (accessed in Oct 24, 2019)

Table 4.2: The unique number of records

Type of records	**The number of records**
{ISBN, bookTitle, authorName}	25,152
{ISBN}	22,346
{bookTitle}	22,168
{authorName}	10,485

4.3 Implementations

In matching of datasets, we use Java (1.8.0_121 version) in Eclipse IDE (201903 (4.11.0)) for extracting book data from [7]. In addition, we extract the data from DBpedia using Apache Jena 3.12.0. We extract the data in Goodreads from `https://www.goodreads.com/` directly. After we have constructed our dataset, we begin to implement our system which is consisted of author-oriented and baseline methods. We explain the details of two methods in section 4.3.1 and section 4.3.2.

4.3.1 Author-Oriented RS

As we have roughly explained the author-oriented method (proposed methodology) in Chapter 3, we only illustrate our author-oriented method in two perspectives which are the properties we used in LOD and how to compute recommendation when a book has multiple authors.

The properties we used in LOD

The properties we used in LOD [27] can be seen in Table 4.3. We use properties that

Table 4.3: The properties we used in LOD

Property Name	**URI**
The properties belonging to DBpedia ontology	`http://dbpedia.org/ontology/...`
Subject	`http://purl.org/dc/terms/subject`
skos:broader	`http://www.w3.org/2004/02/skos/core#broader`

belonging to DBpedia ontology because of their high quality, clean and well performed data in LOD [22]. We do not use the properties wikiPageExternalLink (`http://dbpedia.org/ontology/wikiPageExternalLink`) since it do not give any useful information to our recommendation. In addition, `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.

How to compute recommendation when a book has multiple authors

There are some books written by multiple authors. In this situation, we compute the z-score (3.3) of each book's recommendation score which is related by each author for normalization, as shown in Fig 4.7. If a book simultaneously appeared in multiple authors' related book lists, we plus their z-scores as well.

4.3.2 Baseline

Content-based RS recommend items which are similar to the target item (a given user liked in the past) based on its attribute [28]. 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. The overview of baseline can be seen in Fig 4.8.

We suppose that if a user has read "1Q84" written by Haruki Murakami. He/she is also fond of this book as well. In this situation, we can extract the similar books of "1Q84" from LOD according to their common property values. Using SPARQL query, we can easily extract the similar books of "1Q84" in LOD, which is shown in Fig 4.9. Here, we use properties in LOD as same as proposed method, which is shown in Table 4.3. The results shown in Fig 4.10 indicate that there are some of books are written by Haruki

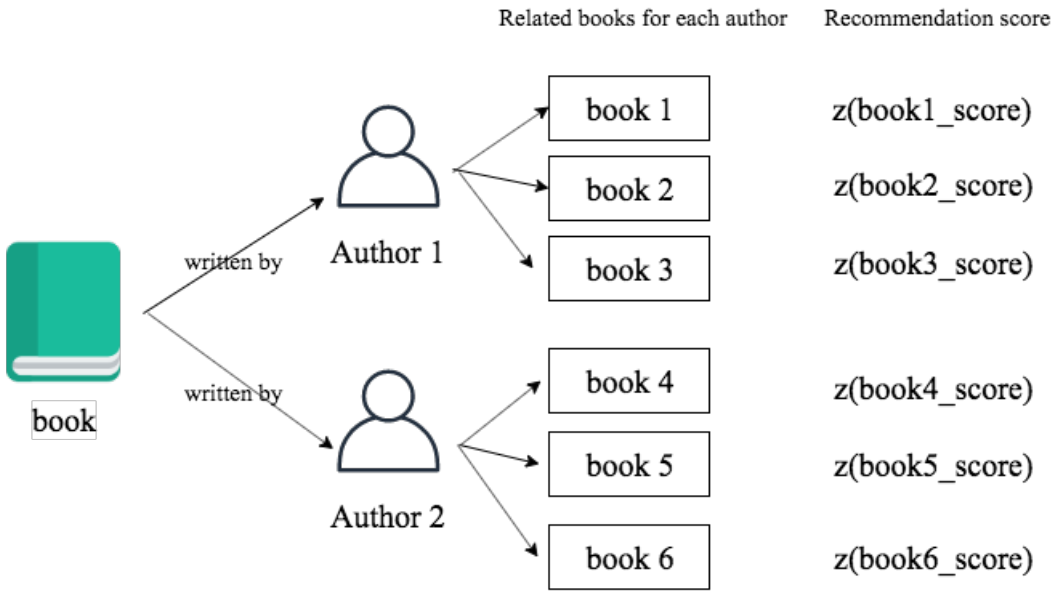

Figure 4.7: How to compute recommendation when a book has multiple authors

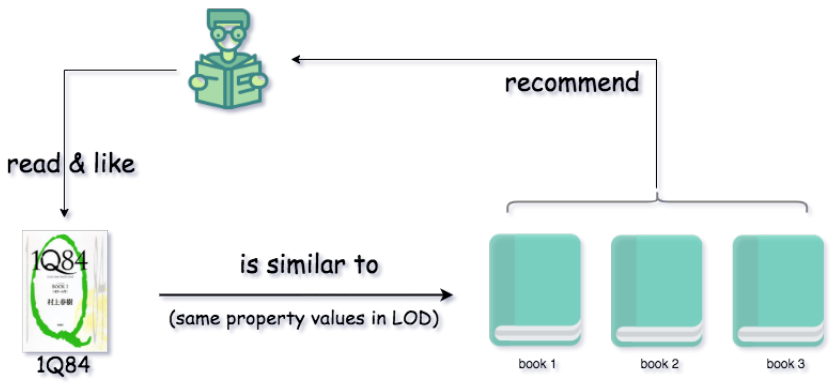

Figure 4.8: The system overview of baseline

Murakami, such as *Wind-Up Bird Chronicle*, *Wild Sheep Chase* and *Dance Dance Dance*. The left column is the similar books of "1Q84". The right column is the number of common property values with "1Q84". Then, we can calculate the recommendation score using Jaccard Similarity (formula 3.1) based on their property values. Finally, we recommend books based on the recommendation score.

4.3.3 Implementation of Step 2 in our experiment design

We have explained how to generate recommendations using author-oriented and baseline books in section 4.3.1 and section 4.3.2. We show them in one page for comparison.

In order to achieve the design of step 2 in our experiment, we give weights to each rated books based on participant's rating, which is shown in Fig 4.11. For example, if a user rate `book 1` with 2 stars and `book 2` with 3 stars, the weight of `book 1` is 0.4 and the weight

```
PREFIX  dcterms: <http://purl.org/dc/terms/>
PREFIX  skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX  dbpedia: <http://dbpedia.org/resource/>
PREFIX  dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT  (replace(str(?book_title), "http://dbpedia.org/resource/", "") AS ?similar_books)
(( ( COUNT(?subject) + COUNT(?skos_object) ) + COUNT(?dbpedia_p) ) AS ?count)
FROM <http://dbpedia.org>
WHERE
  { { SELECT DISTINCT  ?book_title
      WHERE
        { ?book_title  a                  dbpedia-owl:Book ;
                    dbpedia-owl:author    ?author ;
                    dbpedia-owl:isbn      ?isbn
        }
    }
    <http://dbpedia.org/resource/1Q84> (dbpedia-owl:wikiPageRedirects)* ?actualName
      { ?actualName  dcterms:subject  ?subject .
        ?book_title   dcterms:subject  ?subject
      }
    UNION
      { ?actualName  dcterms:subject  ?subject .
        ?subject  skos:broader     ?skos_object .
        ?book_title   dcterms:subject  ?skos_object
      }
    UNION
      { ?actualName  ?p  ?dbpedia_p .
        ?book_title   ?p  ?dbpedia_p
        FILTER regex(?p, "http://dbpedia.org/ontology/")
        FILTER ( ( ?p != dbpedia-owl:wikiPageExternalLink ) )
      }
    FILTER ( ?book_title != ?actualName )
  }
GROUP BY (?book_title)
ORDER BY DESC(?count)
```

Figure 4.9: The SPARQL Query for extracting the similar books of "1Q84" in LOD (accessed in Oct 20, 2019)

similar_books	count
Nineteen_Eighty-Four	12
Masha,_or_the_Fourth_Reich	6
Battle_Royale	6
1985_(Anthony_Burgess_novel)	5
The_Wind-Up_Bird_Chronicle	4
A_Wild_Sheep_Chase	4
Heliopolis_(Jünger_novel)	4
Dance_Dance_Dance_(novel)	4

Figure 4.10: A part of results on 1Q84 ' s similar books in LOD according to Fig 4.9.

of `book 2` is 0.6. if the `book 3` is considered as `book 1`'s related book by author-oriented and baseline method and the `book 3` has a recommendation score 0.9 with `book 1`, `book 3`'s recommendation score is 0.36 in this situation.

As it takes time to generate related books based on our author-oriented and baseline method in the experiment which may has an effect in the user satisfaction of our RS, we generate the related books of 100 books in advance. As a consequence, we can generate recommendations to participants based on participants' rating in the step 2 of our experiment design (Fig 4.1).

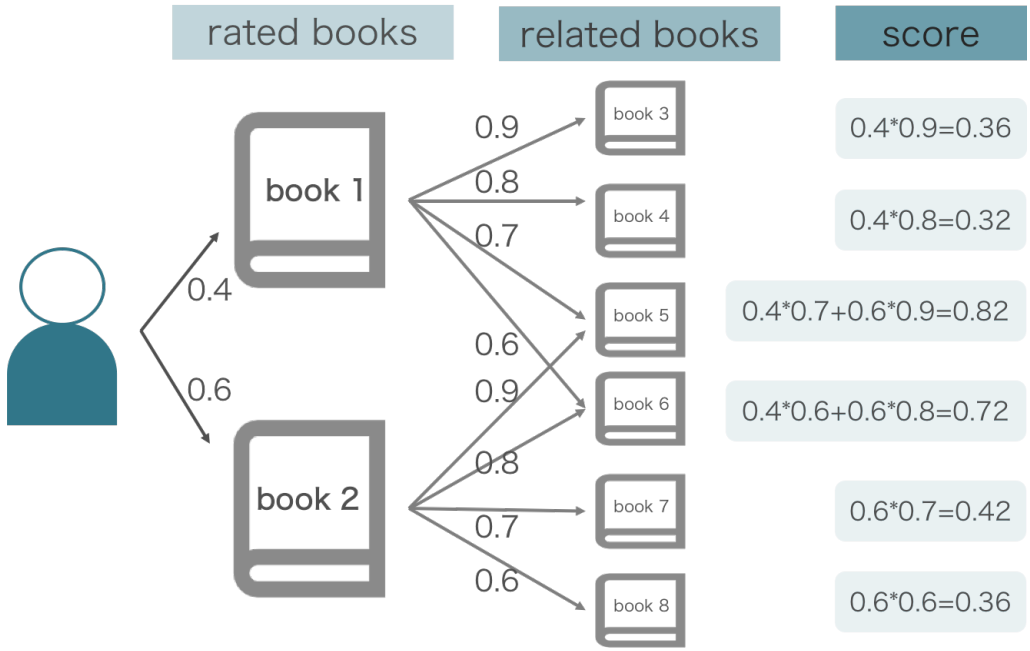

Figure 4.11: The weight calculation of rated books

4.3.4 RS interface

In this section, we show our RS interface based on the experiment design (Fig 4.1).

In the step 1, it can be seen in Fig 4.12. After answered Questionnaire 1 in the step 1, we show 100 books page to participants, as shown in Fig 4.13. The participants can give a rating to books by click the 5-Star rating, as shown in Fig 4.14. In order to show more information about books, we also offer 'Read more' function in Fig 4.13. Participants can click the 'Read more' button and scroll the text if they want to learn more about the book they are interested in, as shown in Fig 4.15.

Then, the recommendation page is shown to participants when they rated at least 5 books, as shown in Fig 4.16. The left part is the recommendation list. The right part is the questionnaires consisted of Questionnaire 2.

4.4 How to evaluate serendipity

4.4.1 Questionnaire design

In this section, we illustrate our questionnaire design.

We have discussed that how previous studies evaluated serendipity in section 2.2.2. As serendipity contains an emotion dimension [9] which is difficult to judge by machine, we do not consider to evaluate it by offline. For online evaluation, it is easy to find that the previous studies evaluate serendipity by requiring participants to answer questionnaires. Said et al. [14] only set one question in questionnaire to evaluate serendipity. Oku et al. [18] and Kotkov et al [8] set questionnaires according to the components of serendipity. However, comparing to Oku et al. [18], Kotkov et al's questionnaire [8] is thorough as they cover the definitions of three components. Thus, we evaluate serendipity according to its three components and design experiment questionnaire based on Kotkov et al.'s research [8].

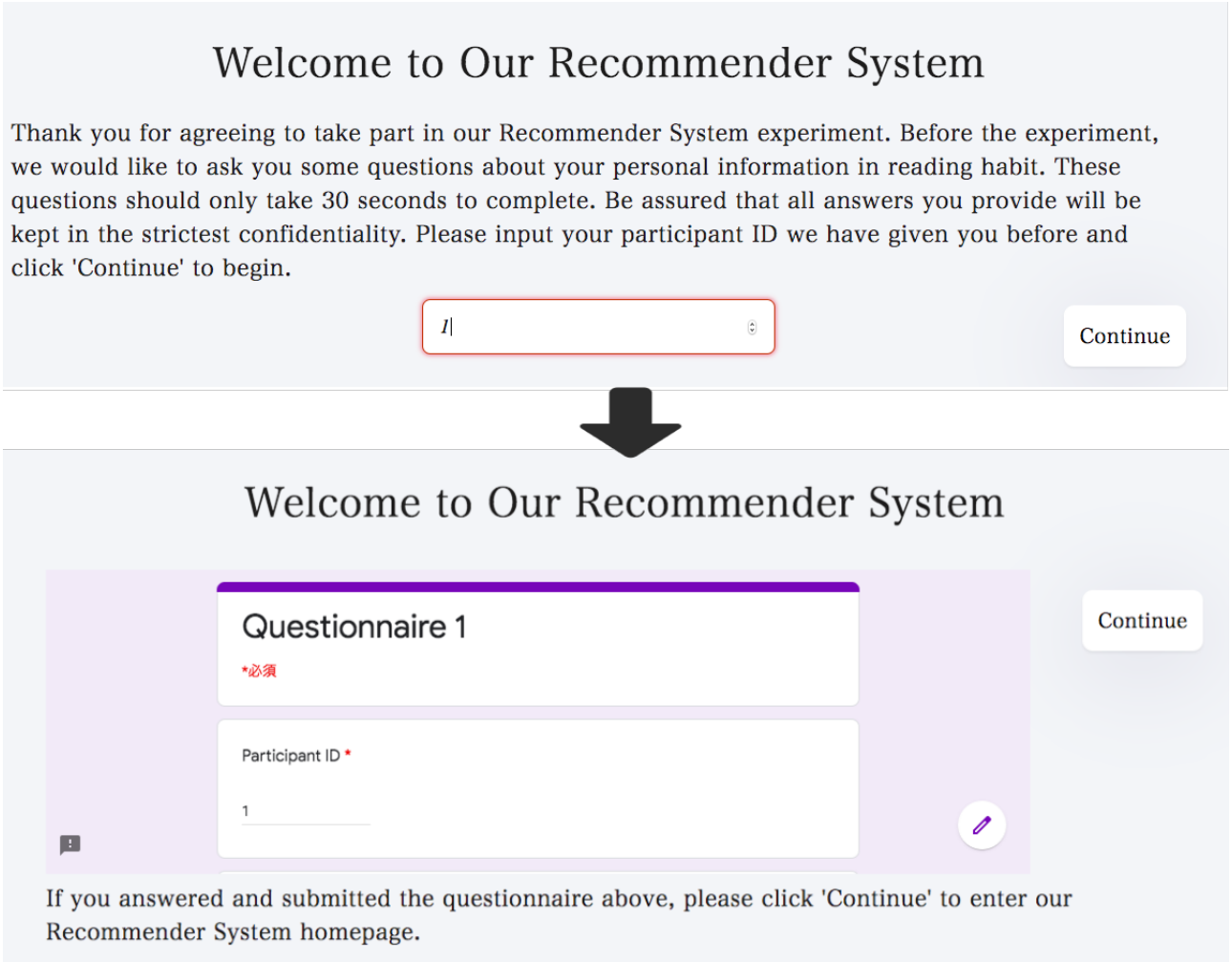

Figure 4.12: The entry page of our RS

As we have illustrated the three components of serendipity in 2.1.2, we design questionnaire based on these definitions which are also discussed according to Kotkov et al.'s studies [4] [8].

For the component of "Novelty", we set two questions responding to its first two definitions, as shown in Table 4.4. Since the last definition of "Novelty" is depending on user's memory which cannot grasp by him/her self, we do not use it.

For the component of "Unexpectedness", we set three questions responding to its last three definitions, as shown in Table 4.5. According to Kotkov et al. [8], they set four questions concerning the four definitions of "Unexpectedness". In this research, we do not use the first definition as it contains the element of relevance, which leads to set the similar question when we consider of relevance metric.

For the component of "Relevance", we set two questions responding to its definition, as shown in Table 4.6. According to Kotkov et al. [8], they do not evaluate the "Relevance" metric. They only ask users about the items they rated with at least 3.5 stars (scale: 0.5∼5 stars) and assume that all the items they ask users about is relevant to users [8]. As Kotkov et al. [8] conduct the evaluation based on real-world RS Movielens[2] which have user groups

[2]Movielens URL: https://movielens.org/

Table 4.4: Novelty questions corresponding to their definitions

Question No.	**Question**	**Definition**
Q1	The first time I heard of this book was when this system suggested it to me.	The user has never heard about the item.
Q2	I heard of this book before, but did not read it.	The user has heard about the item, but has not consumed it.
Unused	Unused.	The user has consumed the item and forgot about it.

Table 4.5: Unexpectedness questions corresponding to their definitions

Question No.	**Question**	**Definition**
Unused	Unused.	The user does not expect this item to be relevant to them.
Q3	I was surprised (not expected) that this system recommend this book to me.	The user does not expect this item to be recommended to them.
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.	The user would not have found this item on their own.
Q5	This book is different (e.g., in style, genre, topic) from the books I usually read.	The item is significantly dissimilar to items the user usually consumes.

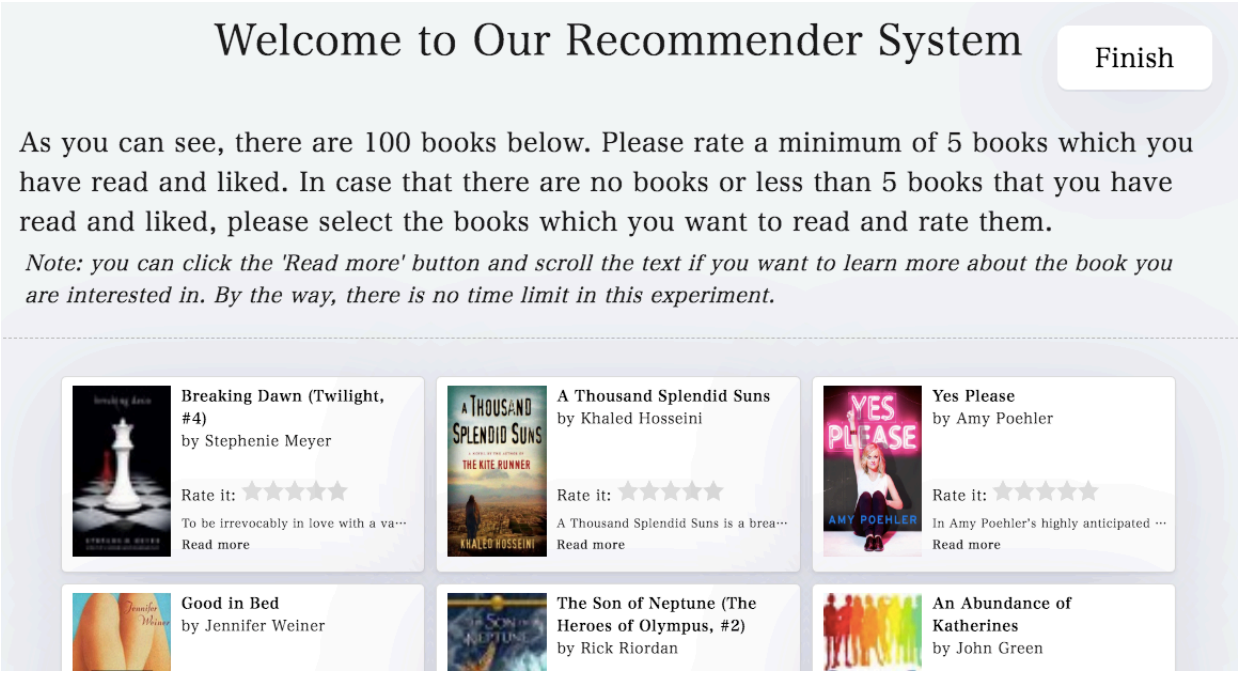

Figure 4.13: The page participants are asked to rating books

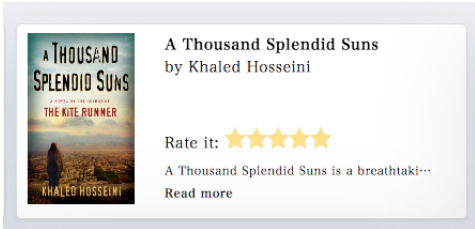

Figure 4.14: 5-Star rating

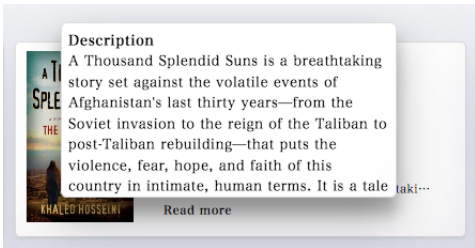

Figure 4.15: Read more function

already, our RS limits us to do the same thing. Thus, we set two questions responding to its definition. Here we divide "Relevance" definition into two questions.

We ask each user to answer the questions using the four-scales as follows:

1- Strongly Disagree

2- Disagree

3- Agree

4- Strongly Agree

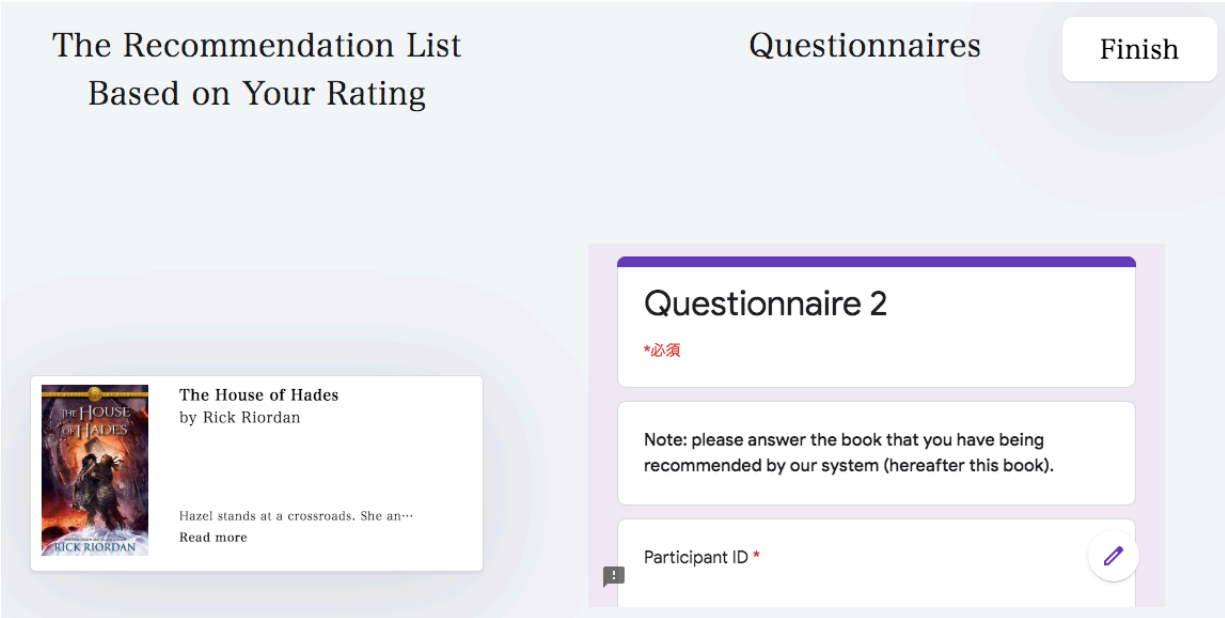

Figure 4.16: The recommendation page

Table 4.6: Relevance questions corresponding to their definitions

Question No.	Question	Definition
Q6	I am interested in this book.	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.
Q7	I want to read this book.	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.

4.4.2 Evaluation metrics

In this section, we introduce the metrics we used in our evaluation.

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

$$Serendipity_{binary} = Novelty \cap Unexpectedness \cap Relevance \tag{4.1}$$

where:

$$Serendipity_{binary} = \begin{cases} 1 & \text{if an item is serendipitous;} \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

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.

However, formula (4.1) only regard serendipitous item as 0 or 1. If a user rate item 1 and item 2 as shown in Table 4.7, item 1 and item 2 are both considered as serendipitous items according to formula 4.1.

Table 4.7: Example of item rating in Questionnaire 2

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
item 1	4	2	4	4	4	4	4
item 2	3	1	3	1	1	3	3

As the user rated the high value on item 1, it is obvious that the user has a strong intention of serendipity in item 1 comparing to item 2. In order to distinguish this intention, we set another serendipity metric which can be seen as follows:

$$Serendipity_{graded} = \frac{Novelty_{intention} + Unexpectedness_{intention} + Relevance_{intention}}{Max(questionnaire\_scaling) \times n} \tag{4.3}$$

where:

$$Novelty_{intention} = \begin{cases} Q1_{answer} + Q2_{answer} & \text{if an item is novel to a user;} \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

$$Unexpectedness_{intention} = \begin{cases} Q3_{answer} + Q4_{answer} + Q5_{answer} & \text{if an item is unexpected to a user;} \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

$$Relevance_{intention} = \begin{cases} Q6_{answer} + Q7_{answer} & \text{if an item is relevant to a user;} \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

$$Serendipity_{graded} = \begin{cases} > 0 & \text{if } Novelty_{intention} \times Unexpectedness_{intention} \times Relevance_{intention} \neq 0; \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

According to the formula 4.3, $Serendipity_{graded}$ is the serendipitous intention of an item for a user. Here, $Novelty_{intention}$ means the sum value of Q1 and Q2 answers. $Unexpectedness_{intention}$ means the sum value of Q3, Q4 and Q5 answers. In the same way, $Relevance_{intention}$ means the sum value of Q6 and Q7. $Max(questionnaire\_scaling)$ denotes the max scaling of our questions. n is the number of questions. For example, the serendipitous intention of item 1 for a user can be $\frac{(4+2)+(4+4+4)+(4+4)}{4\times7} = 0.93$ based on Table 4.7. For item 2, it can be $\frac{(3+1)+(3+1+1)+(3+3)}{4\times7} = 0.54$.

As a result, we set two metrics to evaluate serendipitous items, which are $Serendipity_{binary}$ and $Serendipity_{graded}$. For $Serendipity_{binary}$, it can be seen in formula 4.1. For $Serendipity_{graded}$, it can be seen in formula 4.3.

4.5 User Experiment

We conduct a user experiment for verifying whether our proposed method give an effective performance or not comparing to baseline.

4.5.1 Participants

Preliminary experiment (pilot study) was performed to ensure that whether our experiment could work well or not. Then, we recruited 14 participants in University of Tsukuba who often read English books. They are from China, Philippines, Indonesia and Japan. Moreover, the age of participants is between 18 and 39 years old. 71.4% of them are females and 28.6% are males. The distribution of their affiliations can be seen as follows:

(1) Graduate School of Comprehensive Human Sciences (8 participants)
(2) Graduate School of Systems and Information Engineering (2 participants)
(3) Graduate School of Life and Environmental Sciences (2 participants)
(4) Graduate School of Humanities and Social Sciences (1 participant)
(5) Graduate School of Library, Information and Media Studies (1 participant)

As we implemented our system on Amazon server, the participants can directly access our system via our server IP address. To conduct our experiment smoothly, we conducted our experiment and the experiment steps can be seen in Section 4.1.

4.5.2 Procedures

As we have introduced the design of our experiment in section 4.1, we show the details of our experiment procedures in this section.

We do the experiment one by one.

Firstly, a target participant is asked to enter our book RS and enter their participant ID we have given him/her before.

Then, he/she is required to answer the Questionnaire 1 which consists of demographic information.

Next, there is a page consisted of 100 books shown to him/her after he/she answered and submitted Questionnaire 1. He/she are asked to rate a minimum of 5 books which he/she has read and are fond of them as well. In case that there are no books or less than 5 books that he/she has read and are fond of. He/she are asked to select the books which he/she want to read and rate them. There is also a note that he/she can click the 'Read more' button and scroll the text if he/she want to learn more about the book he/she is interested in. We also tell him/her that there is no time limit in our experiment.

After that, there is a recommendation page shown to him/her after he/she rated a minimum of 5 books and submitted the response. A recommendation page is consisted of Questionnaire 2 and book recommendations. As there are 20 book recommendations, he/she are asked to answer Questionnaire 2 corresponding to each book recommendation. After answered a half of questionnaires (Questionnaire 2) shown in recommendation page, he/she are asked to take a break about 10 minutes.

Finally, he/she is asked to answer the feedback (Questionnaire 3) of our system after he/she answered all of questionnaires in recommendation page.

4.5.3 Apparatus

The equipment and its version we used in the experiment is shown in Table 4.8. We conducted the experiment from Nov 22, 2019 to Dec 11, 2019. The experiment was conducted in University of Tsukuba Library's (Central Library and LIS Library) seminar room.

Table 4.8: User experiment equipment and its version

Equipment	**Version**
Server (Amazon server)	Amazon Linux AMI release 2018.03
PC (MacBook Air)	13-inch, Early 2015, OS version 10.12.6
Browser (Firefox)	70.0.1 (64 bit)

4.6 Results

In this section, we show four parts of the summary data getting from our experiment.

4.6.1 Part 1: The answer of Questionnaire 1

Firstly, we show the answers from our Questionnaire 1, which contains participant demographic information in our experiment, as shown in Table 4.9.

In Table 4.9, "ID" means Participant ID. "Most Read" and "Heard of Book RS" correspond to the questions "Which of the following types of books do you read most?" and "Have you heard of book recommender system?" in Questionnaire 1. Table 4.9 demonstrates that 71.4% of participants are female and 28.8% are male in our experiment. Additionally, their age group percent can be shown as follows:

(1) 7.1% of them are 18-20 years old.
(2) 35.7% of them are 21-23 years old.
(3) 35.7% of them are 24-26 years old.
(4) 14.3% of them are 27-29 years old.
(5) 7.1% of them are 30-39 years old.

There are 28.6% of them read "Science/technology" genre and 14.3% of them read "Art/architecture/design". For participants' reading frequency , 35.7% of them read 1-2 times a month. 28.6% of them read 1-2 times a week and 35.7% of them do the reading almost everyday. Lastly, 57.1% of them have heard of book RS. 28.6% of them answered "maybe". However, 14.3% of them did not know the book RS at all.

4.6.2 Part 2: Participants rated books

In this section, we demonstrate the data which generated from the step 2 of user experiment (Fig 4.1). It can be seen in Table 4.10. It shows that there are average 7.5 books selected by our experiment participants in the step 2 (Fig 4.1). Additionally, their average rating is 3.70. The average of standard deviation (SD) is 0.75.

Table 4.9: Participant demographic information (Questionnaire 1's answer)

ID	Gender	Age Group	Most Read	Reading Frequency	Heard of Book RS
1	Female	24-26 years old	Humanities/religion	1-2 times a month	No
2	Female	21-23 years old	Travel guides	Everyday or almost everyday	Maybe
3	Male	27-29 years old	Education/personal development	Everyday or almost everyday	No
4	Female	21-23 years old	Art/architecture/design	1-2 times a week	No
5	Female	21-23 years old	Children	1-2 times a week	Maybe
6	Male	27-29 years old	Life/health/cookery	Everyday or almost everyday	No
7	Female	18-20 years old	Science/technology	1-2 times a month	No
8	Female	21-23 years old	Science/technology	Everyday or almost everyday	No
9	Male	30-39 years old	Science/technology	1-2 times a month	No
10	Female	24-26 years old	Art/architecture/design	Everyday or almost everyday	Yes
11	Male	21-23 years old	Science/technology	1-2 times a week	Maybe
12	Female	24-26 years old	Life/health/cookery	1-2 times a week	Maybe
13	Female	24-26 years old	Literary/criticism	1-2 times a month	No
14	Female	24-26 years old	History/geography	1-2 times a month	Yes

4.6.3 Part 3: The answer of Questionnaire 2

Fig 4.17 demonstrates the overall distribution of answers in Questionnaire 2. For the responses of Q1 in Questionnaire 2, there are 188 responses of `Strongly Agree`, 31 responses of `Agree`, 20 responses of `Disagree` and 41 responses of `Strongly Disagree`. For the responses of Q2, there are 28 responses of `Strongly Agree`, 18 responses of `Agree`, 35 responses of `Disagree` and 199 responses of `Strongly Disagree`. In the same way, the responses of Q3∼Q7 can be seen in Fig 4.17.

Fig 4.18 demonstrates the distribution of questions (Q1∼Q7) in Questionnaire 2 comparing author-oriented to baseline recommended books.

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) between baseline and author-oriented according to t-test. 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) between baseline and author-oriented according to t-test. 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 according to t-test. 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 according to t-test. 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) between baseline and author-oriented according to t-test. 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) between baseline and author-oriented according to t-test. For Q7, the

Table 4.10: Participants rated books' data

ID	The Number of Selected Books	Selected Books Rating Set	Average Rating	SD
1	6	{5,4,4,4,4,4}	4.17	0.41
2	6	{4,4,4,3,3,3}	3.50	0.55
3	6	{4,5,4,3,3,5}	4.00	0.89
4	5	{4,4,5,4,4}	4.20	0.45
5	7	{3,3,2,2,2,4,3}	2.71	0.76
6	11	{3,4,4,5,5,3,3,5,3,3,3}	3.73	0.90
7	8	{4,5,5,4,4,5,4,4}	4.38	0.52
8	15	{3,3,4,4,4,2,3,3,4,3,3,5,3,4,4}	3.47	0.74
9	11	{3,5,3,2,2,5,5,5,3,5,4}	3.82	1.25
10	6	{5,4,3,3,2,3}	3.33	1.03
11	6	{4,3,4,4,4,4}	3.83	0.41
12	6	{4,4,2,4,4,2}	3.33	1.03
13	5	{3,3,4,4,5}	3.80	0.84
14	7	{4,4,3,3,3,5,3}	3.57	0.79

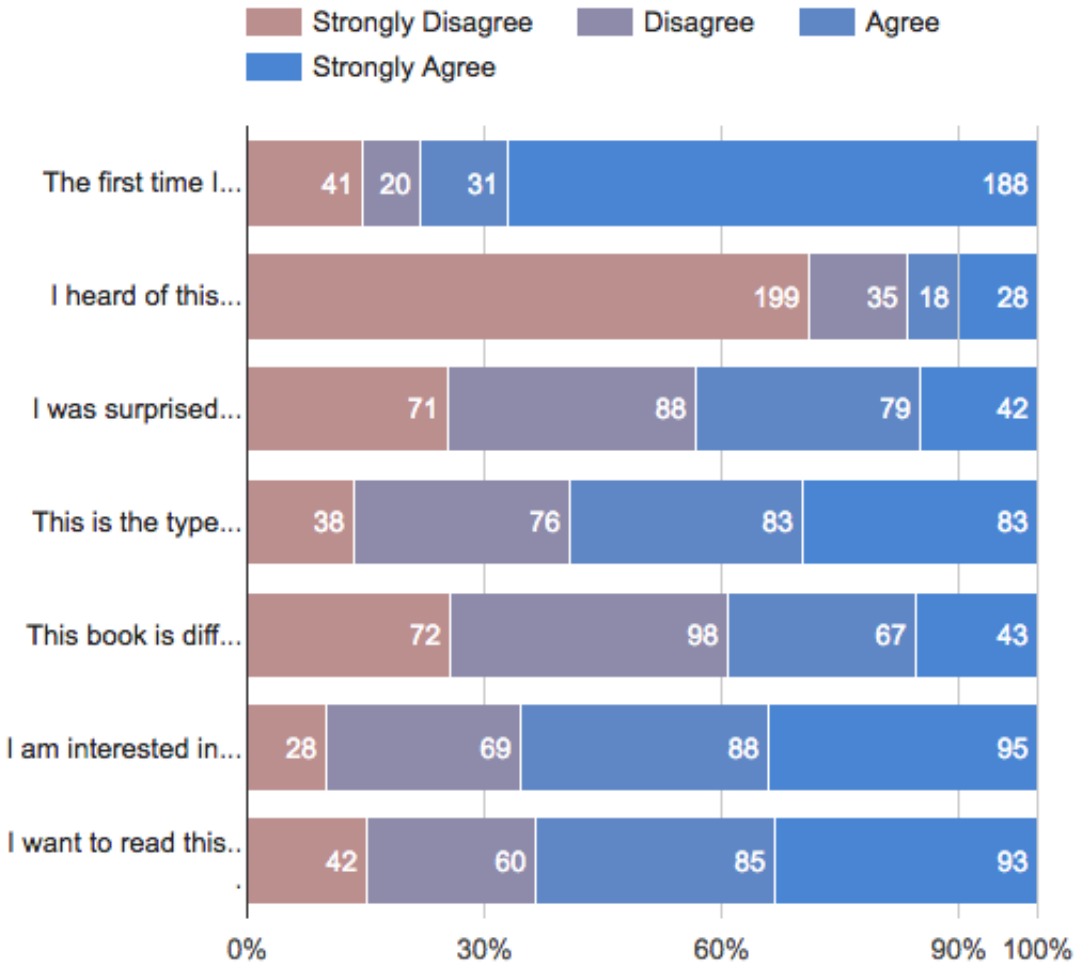

Figure 4.17: The overall distribution of answers in Questionnaire 2

mean rating of baseline is 2.84. The mean rating of author-oriented is 2.80. There is no significant difference (p=0.76) between baseline and author-oriented according to t-test.

$Serendipity_{binary}$

For $Serendipity_{binary}$, we have introduced that our serendipity and its three components are binaries according to our definition. In Fig 4.19, we demonstrates the mean of serendipity's three components based on $Serendipity_{binary}$ comparing author-oriented to baseline. 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 according to t-test. 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) between baseline and author-oriented according to t-test. 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) between baseline and author-oriented according to t-test. For the $Serendipity_{binary}$ of baseline, its mean is 0.35. For the $Serendipity_{binary}$ of author-

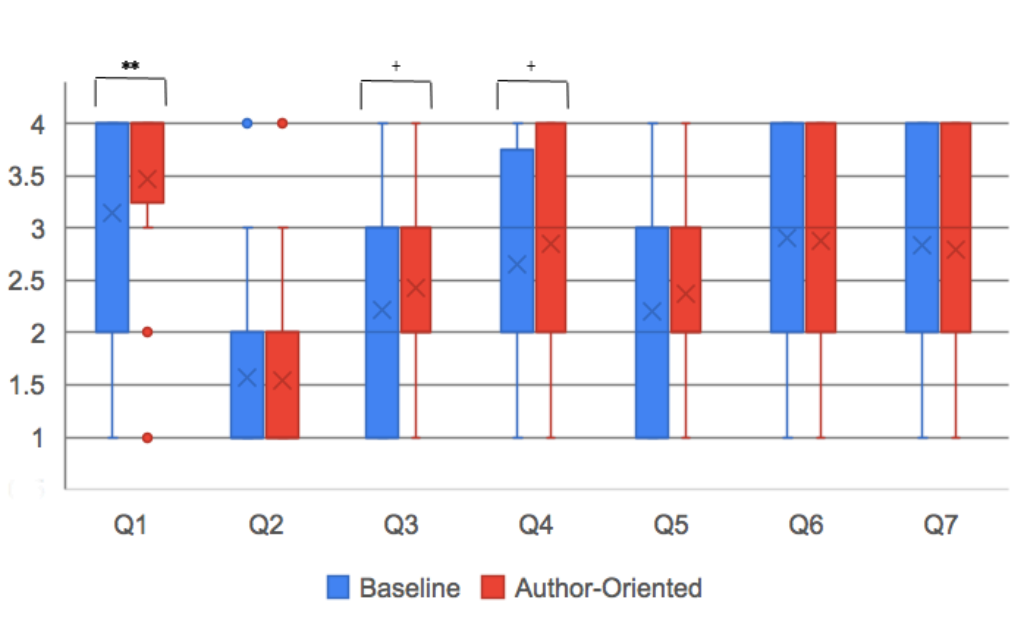

Figure 4.18: The distribution of questions (Q1∼Q7) in Questionnaire 2 (comparing author-oriented to baseline). Significance codes: "**": p≤0.01, "+": p≤0.1

oriented, its mean is 0.39. There is no significant difference (p=0.49) between baseline and author-oriented according to t-test.

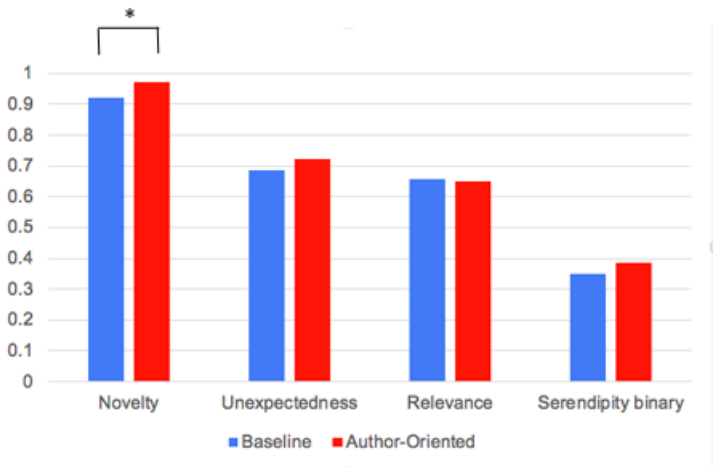

Figure 4.19: The mean of serendipity's three components comparing author-oriented to baseline. Significance codes: "*" $\leq$ 0.05

Fig 4.20 demonstrates the mean of serendipitous books in top-n (from top-1 to top-10) 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>.05) between baseline and author-oriented according to t-test.

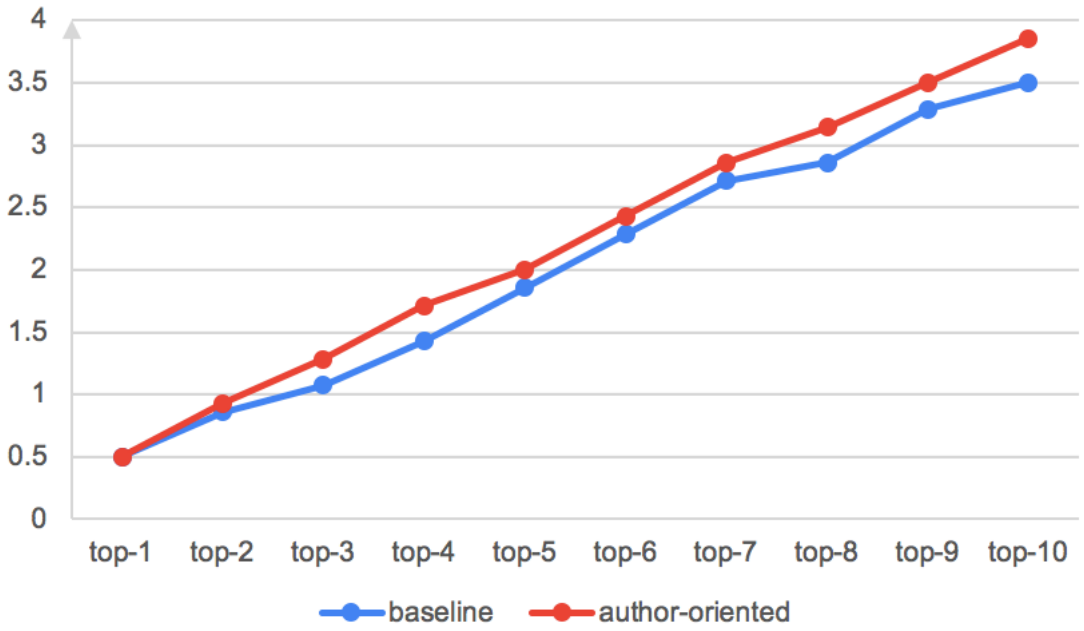

Figure 4.20: The mean of serendipitous books in top-n recommendation list according to $Serendipity_{binary}$ (comparing author-oriented to baseline)

$Serendipity_{graded}$

For $Serendipity_{graded}$, we have introduced that the three components of serendipity are the sum of question answers according to their definitions. For the $Serendipity_{graded}$ of baseline, its mean is 0.24. For the $Serendipity_{graded}$ of author-oriented, its mean is 0.28. Fig 4.21 demonstrates the mean of serendipity's three components based on $Serendipity_{graded}$ comparing author-oriented to baseline. 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 according to t-test. 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) between baseline and author-oriented according to t-test. 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) between baseline and author-oriented according to t-test.

Fig 4.22 demonstrates the mean of serendipitous intention in top-n (from top-1 to top-10) recommendation list comparing author-oriented to baseline. For baseline, the mean of serendipitous intention is 2.39 in top-10 recommendation list. For author-oriented, the mean of serendipitous intention is 2.80 in top-10 recommendation list. There is no significant difference (p>.05) between baseline and author-oriented according to t-test.

4.6.4 Part 4: The answers of Questionnaire 3 & Recommendation List Satisfaction

In this section, we show the feedback from participants, as shown in Appendix A. We classified the feedback into 5 categories by manual: Tag 1: RS Satisfaction, Tag 2: RS Improvement, Tag 3: The Satisfaction of Recommendation, Tag 4: Serendipitous Discovery,

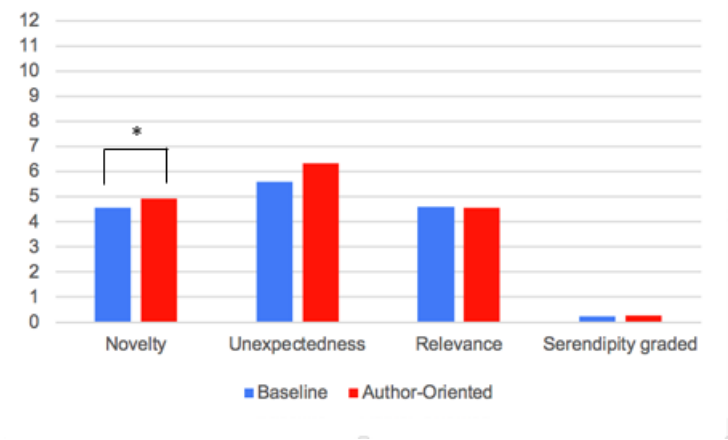

Figure 4.21: The mean of serendipity's three components comparing author-oriented to baseline. Significance codes: "*": $p \leq 0.05$

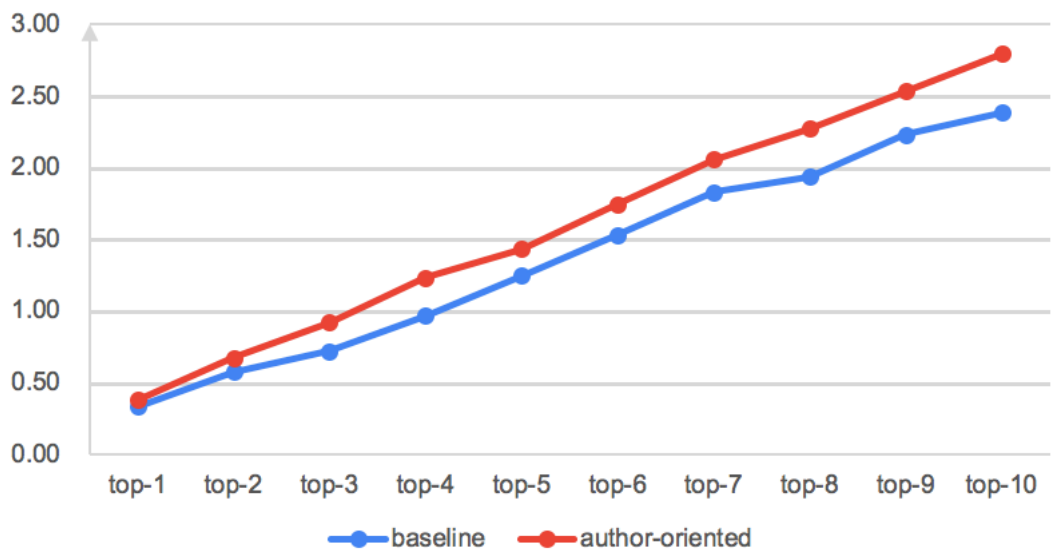

Figure 4.22: The mean of serendipitous intention according to $Serendipity_{graded}$ (comparing author-oriented to baseline)

Tag 5: The Satisfaction of Questionnaire.

The feedback of Tag 1 is the user satisfaction of our system, as shown in Table 4.11. The left column is participant ID and the right column is the feedback about our system satisfaction. It shows that almost all of participants are satisfied with our RS. Only one participant said that `it is somehow helpful but can be better`. Moreover, it also demonstrates that the accuracy of reader preference in our RS need an improvement according to participant ID 6.

Tag 2 is the feedback of our system improvement, as shown in Table 4.12. The left column is participant ID and the right column is the feedback about our system satisfaction. It shows that our participants are mainly dissatisfied with the description and its word size of books.

Tag 3 is the satisfaction of recommendations, as shown in Table 4.13. The left column is

Table 4.11: The feedback of RS satisfaction

ID	RS Satisfaction
3	it is somehow helpful but can be better
6	very useful for reader to looking for the next book to read, the accuracy of reader preference need an improvement
7	might even pay for it to recommend me some books routinely
9	nice, nearly perfect
12	easy to read by the web design
13	would like to use it frequently.
14	50∼60% of recommended books I was interested in, very good

Table 4.12: The feedback of our system improvement

ID	RS Improvement
3	recommendation is a bit too many, the word size of the introduction of books is unfriendly
5	It will be great if i can see the key word of the book
7	the description are too small and some are repeated with non-sense
9	the synopsis of each books are too long for some, the book thumbnails are too small to see, the need of clicking the 'read more' button
12	Could not notice there are more books recommended below

participant ID and the right column is the satisfaction of recommendations. It shows that some of them satisfied with recommendation list 1 and others satisfied with recommendation list 2. Only participant ID 2 shows that he/she is satisfied with both of recommendation lists. Besides, there are also some of them want more other books.

Table 4.13: The satisfaction of each recommendation list

ID	The Satisfaction of Recommendations
1	group2 leave deeper impression
2	The recommendation lists are both close to my interest.
4	The first book-list does not cover the books that I usually read.
10	want more other books
11	books are more representative.

Tag 4 is the discovery of serendipity, as shown in Table 4.14. The left column is participant ID and the right column is the discovery of serendipity. There are three of participants show that some of our recommendations are surprised to them.

Table 4.14: The discovery of serendipity

ID	The Discovery of Serendipity
3	a bit surprised that it can dig out some books i am interested in
7	some books rarely discovered by myself, and surprisingly
9	surprised by the some of the recommended books that was shown to me.

Tag 5 is the satisfaction of our questionnaire, as shown in Table 4.15. The left column is participant ID and the right column is the satisfaction of our questionnaire. It shows that one of our participant think that our questionnaire is through.

Table 4.15: The satisfaction of our questionnaire

ID	The Satisfaction of Our Questionnaire
8	This questionnaire is thorough and helpful.

Besides, as our recommendation list is generated by order, we asked participants about the recommendation list satisfaction [3] after we finished the experiment, as shown in Table 4.16. The left column is participant ID and the right column is the satisfaction of each recommendation list. It shows that 8 participants have a good satisfaction on author-oriented recommendation list. On the other hand, 5 participants have a good satisfaction on baseline-

[3]We only showed recommendation list ID in the experiment, participants were asked to answer the satisfaction of each list.

based recommendation list. Only one participant answered that the recommendation lists generated by author-oriented and baseline were just the same.

Table 4.16: Recommendation list satisfaction

ID	Recommendation List Satisfaction
1	author-oriented
2	just the same
3	baseline
4	author-oriented
5	author-oriented
6	author-oriented
7	baseline
8	baseline
9	author-oriented
10	baseline
11	author-oriented
12	author-oriented
13	baseline
14	author-oriented

Chapter 5

Discussion

In this chapter, we try to answer the research questions (RQs) based on the results we have gotten from our user experiment. Then, we illustrate why do these results matters. Finally, we show our limitation in this thesis and recommend what practical actions or scientific studies should follow in the future study.

5.1 Research questions

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

We recommended a recommendation list consisting of top-10 author-oriented books and top-10 baseline-based books to each participant. We found that 54 books are serendipitous to participants while using author-oriented method. Thus, the ratio of serendipitous items in book recommendation list is 38.57% (54/140≈0.3857) while using author-oriented method. On the other hand, the ratio of serendipitous items in book recommendation list is 35.00% (49/140=0.35) 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. Moreover, the percent change between author-oriented and baseline-based is 3.57%.

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 4.4.2, 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 significant difference between them (p=0.90).

We can find that author-oriented recommendations are more novel to participants comparing to baseline-based recommendations. It indicates that generating recommendations based on indirect relationship (author-oriented) are more novel than using direct relationship (baseline).

Although the mean ratings of author-oriented in 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`) are greater than the mean ratings of baseline. There is also a marginally significant difference between author-oriented and baseline based on Q3 and Q4. However, there is no significant difference for Unexpectedness according to t-test even the ratio of author-oriented is greater than the ratio of baseline. In the same way, the ratio of baseline is greater than author-oriented. There is also no significant difference for Relevance according to t-test. In other words, author-oriented recommendations do not lower the preference of participants comparing to baseline, even though they are more novel to participants.

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

In this thesis, we set two metrics to evaluate serendipity. Both of them indicates that whether an item is serendipitous to a user or not in an explicit value. For metric 1 ($Serendipity_{binary}$), it regards whether an item is serendipitous to a user or not as 0 or 1. For metric 2 ($Serendipity_{graded}$), it regards the serendipitous intention of an item for a user.

Fig 4.20 shows the mean of serendipitous books in top-n (n=1,2,...10) recommendation list based on $Serendipity_{binary}$. 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 ($p>.05$) between baseline and author-oriented according to t-test.

Fig 4.22 shows the mean of serendipitous intention in top-n (n=1,2,...10) recommendation list based on $Serendipity_{graded}$. For baseline, the mean of serendipitous intention in top-n recommendation list is 2.39. 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 according to t-test.

Unexpectedness and Relevance do not show a significant difference and only Novelty show significant difference according to the answer of RQ2. Although we find that the mean rating of author-oriented book recommendation is higher than content-based (baseline) recommendation in top-n recommendation list on both of our serendipity metrics, there is no significant difference between author-oriented and baseline.

RQ4. How about the user satisfaction of author-oriented recommendation list comparing to content-based recommendation list?

To answer this question, we asked each participant about the recommendation list satisfaction after the experiment. We found that 8 participants showed a great interest in author-oriented recommendation list and 5 participants showed a great interest in content-based (baseline) recommendation list. Only one participant answered that both of them are just the same. As a result, author-oriented recommendation list showed a satisfactory performance comparing to content-based recommendation list (percent change: 21.43%).

5.2 Implications: why do the results matter?

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, there is no significant difference between author-oriented and baseline. However, 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 self comparing to baseline according to Q4. Moreover, author-oriented recommendation gives more surprise to a user comparing to baseline according to Q3. It might support the hypothesis that the books of related author may be difficult for a user to find.

There is no significant difference between author-oriented and baseline for Relevance. 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.

In addition, comparing to content-based recommendation list, author-oriented recommendation list satisfies users relatively. It supports the hypothesis of our research.

In summary, we can say that our book RS is helpful for recommending novel books that are familiar with user preference as well. Our book RS tend to be helpful for recommending books that cannot be normally discovered by users.

5.3 Limitations

There are several limitations in our research. We illustrate them from three aspects: user experiment, our book RS and questionnaire design.

User Experiment

As our book recommendation is English text, we recruit participants who often read English books. However, none of them is native in English. Thus, some book descriptions may be difficult for them to understand, which may influence the decision of questionnaire answers.

For the number of participants, we planed to recruit 20 participants in our experiments at first. However, we only recruited 14 participants as it is difficult for us to recruit the target participants.

System

Some of book descriptions we extracted from Goodreads [7] repeated with nonsense. Besides, a few of books have no description and no book cover. All of these problems may influence the decision of questionnaire answer according to user feedback.

Based on our results, the data tends to demonstrates that our book RS makes an effective performance in user satisfaction and improving serendipity while using LOD resource. However, as 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.

Questionnaire design

Although we design a through questionnaire to evaluate serendipitous items. We did not consider other metrics (such as user satisfaction, preference broadening) to evaluate each book. Our study is limited to evaluate the relationship between serendipity and user satisfaction. We only got the satisfaction of each recommendation list from user feedback.

5.4 Recommendations: what practical actions or scientific studies should follow?

According to the limitation of this thesis, our future study will focus on three aspects: user experiment improvement, RS improvement, questionnaire design. For user experiment, we mainly try to recruit some participants who are English native speaker online (such as Goodreads [7] users) and increase the scale of participants. For RS improvement, we will fill up the book information from other book datasets and construct a traditional book RS to do a comparison. For questionnaire design, to evaluate the relationship between serendipity (or its three components) and satisfaction, we will add some user satisfaction metrics in our questionnaire.

Chapter 6

Conclusion

In this thesis, we use two approaches to improving serendipity in book RS: author-oriented method and LOD resource. 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. Besides, 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.

For the three components of serendipity (novelty, unexpectedness, relevance), we found that the novelty showed an effective performance in our proposed method comparing to baseline.

For Novelty, the results show that there are 97.14% books generated by our proposed method are novel to users, which shows that our proposed method recommendations are more novel to users comparing to baseline (92.14%) based on the definition of Novelty.

For Unexpectedness, the mean ratings of author-oriented in Q3 and Q4 are greater than baseline. There is also a marginally significant difference (p=0.06) between author-oriented and baseline according to Q3 and Q4. However, there is no significant difference between author-oriented and baseline based on the definition of Unexpectedness. In other words, our proposed method cannot show an effective performance in Unexpectedness element comparing to baseline.

For Relevance, the results show that there are 65.00% books generated by our proposed method are relevant to users, while 65.71% books are relevant to users generated by baseline. However, there is no significant difference between author-oriented and baseline based on the definition of Relevance. In other words, baseline cannot show an effective performance in Relevance element comparing to author-oriented.

Moreover, comparing to content-based recommendation list, author-oriented recommendation list showed an effective performance according to our user feedback (shown in Table 4.16). There are 8 (57.14%) participants showed a great interest in author-oriented recommendation list. On the other hand, there are 5 (35.71%) participants showed a great

interest in baseline-based recommendation list.

In summary, 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.

However, it cannot deny the fact that our experiment and book RS have limitations. Firstly, the attribute and number of participants and our book RS need improvement. Secondly, we cannot evaluate the relationship between serendipity and user satisfaction because of our user experiment design.

In the future, we focus on the improvement of user experiment design and book RS construction. For user experiment design, we mainly try to recruit some participants who are English native speaker (such as Goodreads [7] users) and increase the scale of participants. In addition, we will add some user satisfaction metrics in our questionnaire for evaluating the relationship between serendipity (or its three components) and satisfaction. For book RS improvement, we will fill up the book information from other book datasets and construct a traditional book RS to do a comparison.

Acknowledgment

Firstly, I would like to express my special thanks of gratitude to my dearest Professor Dr. Masao Takaku for his advice in my master program. Without his guides, I could not finish this paper at all. Although I could not learn things well at first, although I always do things in a roundabout way, although I toke the attitude of my studies lightly at first... It is he that gives me a lot of freedom to do my research and always encourages me when I fell into a trap. It is he that teaches me to think logically and try things in a more effective way. It is he that let me feel rewarding when I put my research results on an academic conference. Takaku sensei, thank you!

Secondly, I wish to show my gratitude to my secondary advisors: Professor Dr. Norihiko Uda and Professor Dr. Sarcar Sayan. For Uda sensei, thank you for giving me advice on the direction of my research. For Sarcar sensei, thank you for giving me the advice in my master thesis (e.g. experiment part, the introduction part, the discussion part) and making time for me in your busy day. I would also like to thanks other Professors for their precious suggestions in my research.

Thirdly, I am indebted to express my thanks to the members of Takaku Laboratory for helping me directly and indirectly throughout 2.5 years at the University of Tsukuba. For Kikkawa Jiro, thank you for always giving me good advice on my research and teaching me to use Latex properly. For Cong Yan, thank you for always supporting me in my behind and always behaving like a big sister when I felt sad. For Sato Itsumi, thank you for helping me recruit the participants of my experiment when I was concerned about my user study. Although I did not mention names directly, I would like to express my thanks to other members of Takaku Laboratory. Thank you, guys!

Lastly, I wish to acknowledge the support and great love of my family: my father, my mother, my big sisters. Thank you for providing me with moral and emotional support in my life.

Thanks for all your encouragement!

by WENG RENLOU
2020/02/04

Gatsby believed in the green light, the orgastic future that year by year recedes before us. It eluded us then, but that's no matter—tomorrow we will run faster, stretch out our arms farther... And then one fine morning...

—F. Scott Fitzgerald

References

[1] Dietmar Jannach et al. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[2] *E-commerce site: Amazon*. URL: `https://www.amazon.com/`. (accessed: 2019-12-25.)

[3] Marco De Gemmis et al. "An investigation on the serendipity problem in recommender systems". In: *Information Processing & Management* 51.5 (2015), pp. 695–717.

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

[5] 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*. ACM. 2006, pp. 1097–1101.

[6] Cambridge University Press. *Cambridge Dictionary: serendipity*. URL: `https://dictionary.cambridge.org/dictionary/english/serendipity`. (accessed: 2019-12-21).

[7] *Goodreads*. URL: `https://www.goodreads.com`. (accessed: 2019-12-12.)

[8] Denis Kotkov et al. "Investigating serendipity in recommender systems based on real user feedback". In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM. 2018, pp. 1341–1350.

[9] Allen Foster and Nigel Ford. "Serendipity and information seeking: an empirical study". In: *Journal of documentation* 59.3 (2003), pp. 321–340.

[10] Komal Kapoor et al. "I like to explore sometimes: Adapting to dynamic user novelty preferences". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM. 2015, pp. 19–26.

[11] Panagiotis Adamopoulos and Alexander Tuzhilin. "On unexpectedness in recommender systems: Or how to better expect the unexpected". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.4 (2015), p. 54.

[12] 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*. Vol. 1. SCITEPRESS Science and Technology Publications. 2017.

[13] Kenta Oku and Fumio Hattori. "Fusion-based Recommender System for Improving Serendipity." In: *DiveRS@ RecSys* 816 (2011), pp. 19–26.

[14] Alan Said et al. "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. San Antonio, Texas, USA: ACM, 2013, pp. 1399–1408.

[15] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

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

[17] 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. Barcelona, Spain: ACM, 2010, pp. 39–46.

[18] Kenta Oku and Fumio Hattori. "User evaluation of fusion-based approach for serendipity-oriented recommender system". In: *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012)*. 2012, pp. 39–44.

[19] Yves Avérous. *Untranslateable words - serendipity*. URL: `http://www.translorial.com/essays/untranslatable-words-serendipity/`. (accessed: 2019-12-12.)

[20] Matt Bramowicz. *10 MORE Untranslatable Words*. URL: `http://www.lackuna.com/2012/02/22/10-more-untranslatable-words-for-english/`. (accessed: 2019-12-12.)

[21] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. "Metrics for evaluating the serendipity of recommendation lists". In: *Annual conference of the Japanese society for artificial intelligence*. Springer. 2007, pp. 40–46.

[22] Tommaso Di Noia et al. "Linked Open Data to Support Content-based Recommender Systems". In: *Proceedings of the 8th International Conference on Semantic Systems*. I-SEMANTICS '12. Graz, Austria: ACM, 2012, pp. 1–8.

[23] *DBpedia*. URL: `http://dbpedia.org`. (accessed: 2019-12-12.)

[24] 一瀬詩織, 大西可奈子, 小林一郎, et al. "Linked Data を用いた情報推薦への取り組み". In: *第 74 回全国大会講演論文集* 2012.1 (2012), pp. 649–650.

[25] *DBpedia SPARQL endpoint*. URL: `http://dbpedia.org/sparql`. (accessed: 2019-12-12.)

[26] *Goodreads About Us*. URL: `https://www.goodreads.com/about/us`. (accessed: 2019-12-12.)

[27] *Linked Open Data (LOD)*. URL: `https://lod-cloud.net`. (accessed: 2019-12-12.)

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

Appendix A

The Answer of Questionnaire 3

Table A.1: The answer of Questionnaire 3 (1)

ID	Feedback
1	gruop1 is mainly about the novels and fictions, which is commonly ignored when surfing the website. However, the books recommended in group2 leave deeper impression and I am personally quite interested in them.
2	the recommendation lists are both close to my interest, but the cover and the introduction of the books will play a big role for me to decide which to read at the end.
3	20 items of recommendation is a bit too many to focus on what i really want to read. the word size of the introduction of books is unfriendly. a bit surprised that it can dig out some books i am interested in and want to read. it is somehow helpful but can be better.
4	The first book-list does not cover the books that I usually read.
5	it will be great if i can see the key word of the book, so that i can quickly understand if i am interested in it!
6	I think this recommendation system is very useful for reader to looking for the next book to read. Even the accuracy of reader preference at this moment need an improvement. I hope this system will success and give beneficial for readers.
7	I pretty much like the recommendation system that I might even pay for it to recommend me some books routinely. Just the words written in the description are too small and some are repeated with non-sense; otherwise, everything is fine! Some books are rarely discovered by myself, and surprisingly, I'd love to give it a try.

Table A.2: The answer of Questionnaire 3 (2)

ID	Feedback
8	This questionnaire is thorough and helpful.
9	The recommender system is nice, however the only comment or recommendation that I would like to see is that the synopsis of each books are too long for some. it would be better if each has only 2-3 sentences of summary or synopsis in it. Also I feel like the book thumbnails are too small to see. It would also be much better if the description is readily available without the need of clicking the 'read more' button as I find it an additional hassle. Lastly, everything is almost perfect and I was surprised by the some of the recommended books that was shown to me. In fact, I would really love to read those books if I have some spare time. Nonetheless, the recommendation system is nearly perfect.
10	I want more books from ancient years and about the scientific or artistic theories
11	choose books that are more representative.
12	It is easy to read by the web design. Could not notice there are more books recommended below at first though...
13	I didn't know this system before, but now I would like to use it frequently.
14	50∼60% of recommended books that I was interested in, very good system!