Proceedings of the International MultiConference of Engineers and Computer Scientists 2019
IMECS 2019, March 13-15, 2019, Hong Kong

A Book Recommendation Method Based on Paragraph Vector and User's Book Arrangement

Tatsuya Miyamoto, Daisuke Kitayama

***Abstract*—In recent years, the evaluation of recommender systems has focused on not only accuracy but aspects such as novelty, diversity, explainability, coverage, and serendipity as well. This is because users are not always satisfied with book recommender systems focusing on accuracy alone. Recent evaluation measures provide a solution for this problem. In this paper, we propose a book recommendation method that considers serendipity and explainability based on user book arrangement. We focus on the book arrangement such that if the user arranges the books based on his/her own preferences, we perceive that even the same set of books will be arranged differently by each user. For example, a user arrenges for increasing comedy atmosphere, on the other hand, other user arrenges for increasing difficulty of mystery. We propose a book recommender system that understands the intention of a user's book arrangement and recommends books accordingly. We investigate the effectiveness of our proposed approach through experiments on novels from book review sites.**

***Index Terms*—Paragraph vector, Book recommendation, Regression**

I. Introduction

IN recent years, researchers have focused on not only the accuracy of recommender systems but also their novelty, diversity, explainability, coverage, and serendipity [1]–[3]. This is because users are not always satisfied with recommender systems that focus on accuracy alone.

In this paper, we propose a book recommendation method that considers serendipity and explainability and is based on user book arrangement. We focus on the book arrangement such that if the user arranges the books based on his/her own preferences, we perceive that even the same set of books will be arranged differently by each user (e.g., gradually increase comedy atmosphere, difficulty of mystery, and readability). In this approach, the recommender system understands the intention of a user's book arrangement and recommends books accordingly. The book arrangement is selected according to the user's recommendation, which is easy to understand. Thus, we expect this approach to have high explainability. Moreover, we expect this approach to have high serendipity because the system understands the implicit intention of the user's book arrangement.

The overview of our approach is depicted in Fig 1, 2. We generate a paragraph vector using book reviews. Next, the system performs a regression of the paragraph vector for the user's arrangement, to predict a feature vector for a certain position. Finally, the system recommends a book with the highest similarity to the predicted feature vector.

Manuscript received December 20, 2018; revised January 10, 2019.

This work was supported by ISPS KAKENHI of Grant-in-Aid for Scientific Research(C) Grant Number 18K11551.

T. Miyamoto is with the Graduate School of Engineering, Kogakuin University, Japan; e-mail: em18014@ns.kogakuin.ac.jp.

D. Kitayama is with the Faculty of Informatics, Kogakuin University, Japan; e-mail: kitayama@cc.kogakuin.ac.jp.

We first explain the related work in Section II and then propose our approach in Section III. We describe our experiments for evaluating the method in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

Many studies have already researched the topic of book recommendation. Givon et al. [4] suggested implementing book recommendation for the cold start problem. Toward this end, they proposed giving automatic social tags to newly published books. Minami et al. [5] suggested that an emphatic focus on reviewers improves user satisfaction. To this end, they proposed strengthening the influence of reviewers with viewpoints similar to those of the user using collaborative filtering.

Liu et al. [6] proposed a novel model for recommending user-generated item lists. They focused on item lists without an item order. In contrast, our approach focuses on arrangement in which the item order does not appear in time series.

Oku et al. [7] proposed a recommender system with a fusion-based approach. This approach finds new items that mix two user input item features. Our approach is not limited to the number of user input items and considers the positions of user input items. We therefore expect to recommend items of a wider variety.

Green et al. [8] used tag clouds to explain a recommended item. User can understand intuitively input items relevance to recommend item by looking at the word size in tag clouds. Our approach does not show words pertaining to item relevance. However, our approach recommends items that are relevant to the user's selection. Therefore, users can intuitively understand item relevance.

III. BOOK PREDICTION METHOD

In our method, we recommend books based on the user's book arrangement. First, the user inputs several books in an order depending on his/her likes. Our method predicts books to be inserted between input books based on the user's book arrangement.

A. Generating Paragraph vector

A paragraph vector [9] is one of the distributed expressions for document paragraphs. We generate a paragraph vector using book reviews in Japanese. Book reviews contain book synopses and readers' impressions of the book. Therefore, we consider that the paragraph vector generated by these reviews can represent the content of the book.

The Japanese review has not been separated/analyzed in terms of the words it contains. Therefore, we use the morphological analyzer to divide the sentences of the review into words. At this time, the associated person's name is

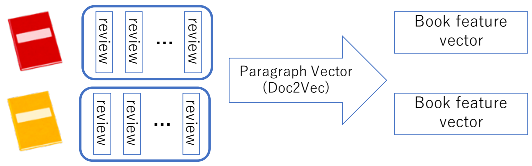

Fig. 1. Overview of create book feature vector.

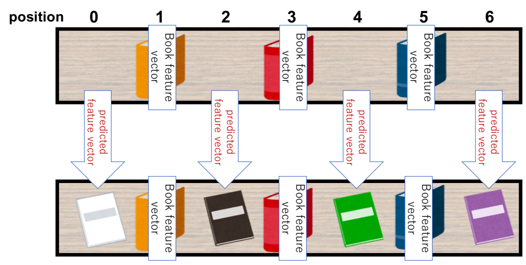

Fig. 2. Overview of book recommendation system with predicted feature vector.

converted into a symbol. We use MeCab [10] for the morphological analysis. The book feature vector is generated using Doc2Vec, and the input document is one document, which contains review grouping by book title. We adopt Paragraph Vector-Distributed Memory (PV-DM) as the learning model, with a feature vector size of 300, window size of 15, and min-count of 5. Table I lists books similar to "The Adventures of Sherlock Holmes" from the generated paragraph vector. These books share a common feature, namely, the famous detective solves a case in each. Therefore, we believe that the generated paragraph vectors can properly express the content of the books. Notably, the missing ranking is the Sherlock Holmes series in Table I for easy visibility.

B. Predicting feature vector

In this section, we describe the book prediction method using regression. Regression is performed for each element of the feature vector of the user's input book. The position number in the user's book arrangement is the independent variable. The dependent variable is the scalar of the feature vector element. We attempt linear or kernel ridge regressions. For each element, a predicted feature vector is generated from regression equations with the position number of the prediction. We define the predicted feature vector of the ith position as follows.

$$pred_i = [p_{i,1}, p_{i,2}, ..., p_{i,n}] \quad (1)$$

where $p_{i,1}$ is the predicted scalar of the 1st element using a regression method.

We obtain a different coefficient of determination for each vector element using linear regression. Figure 3 shows the differences in the coefficients of determination for each vector element. Element A has a high coefficient of determination. On the other hand, element B has a low coefficient of determination. We presume that a feature vector element with a high coefficient of determination involves book arrangement. In addition, element A has a high gradient, whereas element B has a low gradient. We presume that feature vector elements with high gradients involve book

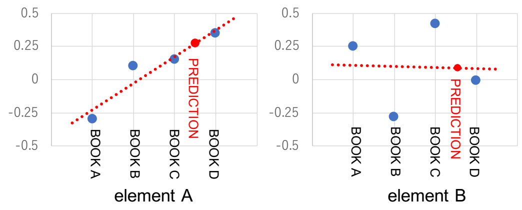

Fig. 3. Arrangement prediction performance for each element of the feature vector.

arrangement. Therefore, we define weight as follows.

$$w = [w_1, w_2, ..., w_n] \quad (2)$$

where w_1 is the weight value of the 1st element. We use one of the following as a weight value.

- Coefficient of determination
- Gradients
- Product of gradient and coefficient of determination

C. Book similarity

The similarity between the candidate book and predicted feature vector is calculated by the following method.

- Weighted cosine similarity
- Cosine similarity

The weighted cosine similarity between the book feature vector b_j of book j and the predicted feature vector $pred_i$ of the ith position is expressed as follows:

$$cos(pred_i, b_j) = \frac{\sum w_k^2 p_{i,k} b_{j,k}}{\sqrt{\sum (w_k p_{i,k})^2} \cdot \sqrt{\sum (w_k b_{j,k})^2}}. \quad (3)$$

We complete this calculation for the position at which the user's input book does not exist.

There are five methods, LC, LM, LG, L, K, combining two types of regressions, two types of similarities, and three types of weights, as seen in Table II. Recommend book is the book with the feature vector having the highest similarity to the predicted feature vector. However, we filter out books written by the same author from the inputted and recommended books. Otherwise, the recommended book would likely belong to the same series or be written by the same author.

IV. EXPERIMENT RESULTS

We evaluate predicted feature vector for the five methods. We have experiment with a novel. We obtain a total of 238,135 book metadata of novels from Rakuten Books[1] and a total of 10,102,455 reviews of the books from Dokusyo Metor[2]. We generate a feature vector as described in Section III-A from these reviews.

A. Comparison method

There are five comparison methods, as described in Table II of Section III-C, excluding the average method (AVG). The book recommended as per AVG shows the feature vector most similar to the average of the left and right input book feature vectors positioned between those books.

[1]https://books.rakuten.co.jp/
[2]https://bookmeter.com/

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
BOOKS SIMILAR TO "THE ADVENTURES OF SHERLOCK HOLMES"

Rank	Title	Author	Cos similarity
7	Poirot Investigates	Agatha Christie	0.70
15	The Roman Hat Mystery	Ellery Queen	0.66
19	The Innocence of Father Brown	Gilbert Keith Chesterton	0.64
20	The Thirteen Problems	Agatha Christie	0.63
21	Cases of Akechi Kogoro	Edogawa Rampo	0.63

TABLE II
5 TYPES METHODS

	Regression	Similarity	Weight
LC	Linear	Weighted cosine similarity	Coefficient of determination
LM	Linear	Weighted cosine similarity	Multiplying
LG	Linear	Weighted cosine similarity	Gradients
L	Linear	Cosine similarity	-
K	Kernel ridge	Cosine similarity	-

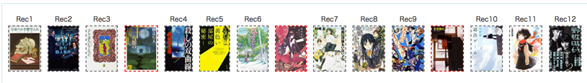

Fig. 4. System with an arrangement for 15 books.

TABLE III
THREE QUESTIONS FOR EACH METHOD

No.	Question	Answer
Q1	The book arrangement is similar to my book arrangement.	seven-level scale
Q2	I am interested in the recommended book.	yes/no in each book
Q3	I have already read the recommended book.	yes/no in each book

B. Experimental method

We use CrowdWorks[3] to gather data. Twenty-six subjects participated in our study and 33 data items were gathered. Subjects could participate in the experiment up to four times. The experimental procedure is as follows:

1) A subject searches for and selects three different books.
2) The subject rearranges the input books using a intentional book arrangement and provides information about the intention of the book arrangement.
3) The system shows an arrangement of 15 books, including recommended and input books (Table 4). The input books are placed in the fourth, eighth, and twelfth positions from the right. Therefore, the system recommends 12 books for each method.
4) The subject answers three questions for each method (Table III).

At this time, we omit some intentions in step 2 (e.g., release, size, randomness, and reader's preference). Reviews are typically not provided as part of the bibliographic information. Therefore, our approach cannot understand the intention of book arrangements based on bibliographic information. Therefore, we omit some intentions. In addition, we exclude rough intentions because they do not provide an accurate evaluation (e.g., in terms of the reader's liking/preference and randomness). In addition, each subject answered freely about his/her impressions of the system.

[3]CrowdWorks is a crowdsourcing service in Japan. https://crowdworks.jp

TABLE IV
RESULT FOR EACH METHOD

	Matches my intention	Interest	Have read	Rec Book Sim
LC	4.44	6.52	1.09	0.36
LM	3.76	6.18	1.03	0.31
LG	3.88	6.03	1.03	0.32
L	4.64	6.76	1.12	0.41
K	4.48	6.33	0.91	0.42
AVG	4.52	6.67	1.12	0.45

C. Results

Table IV shows the result of all the data averages. "Matches my intention" is the result of the Q1 average (the higher the value, the better the match). "Interest" denotes the recommended book count average the subject is interested in, while "Have read" is the average of recommended books have read already by each subject before the experiment. "Rec Book Sim" is the average of the cosine similarity between the recommended books.

The highest score for "Matches my intention" is that of method L. Thereafter, the highest values appeared in the order of AVG, K, LC, LG, and LM. The method scores that used weights were worse. The highest score for "Interest" is that of method L. Thereafter, the highest values appeared in the order of AVG, LC, K, LG, and LM. The orders for "Matches my intention" and "Interest" are similar. We take into account the possibility that if each subject is interested in many recommended books, the score for Q1 would be high. The correlation coefficient between "Matches my intention" and "Interest" is 0.398, and thus, a weak positive linear relationship exists between those. The score for "Have read" is about 1 for all the methods. Most higher score method is AVG in "Rec Book Sim". The highest score method for "Rec Book Sim" is obtained using AVG, and thus, AVG recommends book similarity well. The other methods tend to recommend books more broadly compared to the AVG method.

Next, we analyze the results in detail. We divide the data in terms of three cases of input books' similarities. The division method is as follow:

A) There is a possibility that the input books are arranged in a linear relationship ($Y + 0.05 < X$).
B) Input books may arranged in a horizontal relationship, but a few vector elements may arranged in a linear relationship ($Y - 0.05 < X \leq Y + 0.05$).

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C) The pattern for the middle input book is different, and thus, the input books did not arrange in a linear relationship ($X \leq Y - 0.05$).

X denotes the similarity average of the input books. Y refers to the similarity of the input books at both ends.

Table V shows the result for case A. Case A includes 10 data items. The score for "Matches my intention" is slightly low when using all methods. Furthermore, the method that provides the highest score for "Matches my intention" changes to K. The score with method LC is the same as that for L for "Matches my intention." We conclude that as the input books line up in a linear relationship, the coefficient of determination is high for many elements. The average coefficient of determination is 0.503 for all the data. In contrast, the average coefficient of determination is 0.556 for case A. Thus, the weight of the coefficient of determination influences/weakens the recommendation provided by LC, which then resembles that provided by L. Additionally, the correlation coefficient between "Matches my intention" and "Interest" is 0.472. Thus, there is a moderate positive relationship between these aspects. This correlation coefficient is higher than that before the sub-divisions by cases.

Table VI shows the result for case B. Case B includes 19 data items. The results of the score relationships for Case B are roughly the same as those before the sub-divisions by cases. We conclude that this is because the more than half data items were classified as case B. The score for "Rec Book Sim" has risen slightly for all the methods. In addition, the correlation coefficient between "Matches my intention" and "Interest" is 0.484. Therefore, a moderate positive relationship exists between these coefficients. The correlation coefficient is higher than before the sub-divisions by cases, similar to case A.

Case C includes 4 data items. This is quite a small amount of data. Hence, we abandon the analysis for case C. However, we conclude that this number of data shows that each subject arranged the input books seriously. If each subject would have arranged the input books randomly, the number of data would have increased in case C.

As a result, we conclude in the event many recommended books of interest exist, there is a high probability that each subject gives a high score for "Matches my intention." Therefore, there is a possibility that the "Matches my intention" score does not completely indicate our intention pertaining to Q1.

The weighted methods (LC, LM, and LG) received low score in all cases. As per the doc2vec model, each element is different from LDA and it is not easy to understand specific factors (e.g., genre, story, or readability). For example, some elements deeply express the fantasy factor, while others do so only slightly. Therefore, if we were to weight each element, the balance of the book factor collapses. Thereby, we conclude that the weighted methods score low in all cases.

In addition, we analyze the answers of each subject with regard to the intentions of the book arrangements. In some cases, the "Matches my intention" score was lower than 4 for all methods. Table VII shows examples of poor "Matches my intention" scores. These intentions refer to the content of the book. We generate a book feature vector using reviews. However, reviews do not touch upon the deeper intention of the book in many cases. Therefore, we conclude that the system

TABLE V
RESULT IN A LINEAR ARRANGED: CASE A (AVERAGE OF 10 DATA)

	Matches my intention	Interest	Have read	Rec Book Sim
LC	4.14	7.60	1.30	0.33
LM	3.14	6.90	1.60	0.29
LG	3.57	6.70	1.30	0.29
L	4.14	7.40	0.90	0.37
K	4.28	6.90	1.00	0.39
AVG	4.00	7.00	1.00	0.41

TABLE VI
RESULT IN A HORIZONTAL ARRANGED: CASE B (AVERAGE OF 19 DATA)

	Matches my intention	Interest	Have read	Rec Book Sim
LC	4.40	6.05	1.10	0.37
LM	3.67	5.84	0.84	0.31
LG	3.67	5.53	1.00	0.33
L	4.60	6.31	1.37	0.43
K	4.33	6.00	0.95	0.44
AVG	4.67	6.37	1.32	0.46

cannot understand the intentions of book arrangements. On the other hand, if the intention is easy to comprehend (as in certain genres), we conclude that the system can understand the intentions of the book arrangements.

We also conclude that L and AVG fundamentally understand the intention of book arrangement. Method L is "Rec Book Sim" provides a score lower than that of AVG. Therefore, method L can provide recommendations for various kinds of books, unlike AVG. This fact is consistent with our intention of wishing to recommend various kinds of books (high serendipity). Thereby, we conclude that method L is better than AVG.

We also analyze the correlation coefficients of "Matches my intention" for all the methods (Table VIII). A strong positive relationship is noted between all the methods. The strongest correlation is that of method L with K, followed by L with AVG Hence, relatively speaking, AVG does not necessarily understand the intention of book arrangement while L does.

V. CONCLUSION

In this paper, we proposed a method for book recommendation that considers serendipity and explainability based on user book arrangement. We focused on book arrangement. If the user arranges the books based on his/her own preferences, we perceive that even the same set of books will be arranged differently by each user (e.g., gradually increase comedy atmosphere, difficulty of mystery, and readability). This approach ensures that the recommender system understands the intention of the user's book arrangement and recommends books that match that intention. We experimentally evaluated which of our methods can correctly understand a user's book arrangement. The experimental results showed the normal linear method (L) can best understand intention of a user's book arrangement. In addition, the results showed that the weighted method cannot maintain the correct balance with regard to the book factor, and thus, the weighted method is not be able to understand the intention of user book arrangement.

We conduct a more detailed future study of this approach by comparing it with other recommender systems from the viewpoint of serendipity and explainability.

TABLE VII
EXAMPLE OF POOR "MATCHES MY INTENTION" SCORES

Intention of book arrangement	LC	LM	LG	L	K	AVG
Ordered in terms of ease of understanding the content.	1	1	1	1	1	1
It is a novel subject matter of Go, elementary school students are arranged in order from the target to the adult.	2	2	1	1	1	1
Arranged in order of the era.	2	2	2	2	3	2

TABLE VIII
CORRELATION COEFFICIENT OF "MATCHES MY INTENTION" BETWEEN METHODS

	LM	LG	L	K	AVG
LC	0.72	0.72	0.87	0.74	0.78
LM		0.75	0.73	0.66	0.66
LG			0.74	0.75	0.69
L				0.85	0.76
K					0.74

REFERENCES

[1] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[2] Yehuda Koren. Tutorial on recent progress in collaborative filtering. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 333–334, New York, NY, USA, 2008. ACM.

[3] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 257–260, New York, NY, USA, 2010. ACM.

[4] Sharon Givon and Victor Lavrenko. Predicting social-tags for cold start book recommendations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 333–336, New York, NY, USA, 2009. ACM.

[5] Daichi Minami and Taketoshi Ushiama. A collaborative filtering method based on empathy with reviewers. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, IMCOM '17, pages 33:1–33:6, New York, NY, USA, 2017. ACM.

[6] Yidan Liu, Min Xie, and Laks V.S. Lakshmanan. Recommending user generated item lists. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 185–192, New York, NY, USA, 2014. ACM.

[7] 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)*, pages 39–44, 2012.

[8] Stephen J. Green, Paul Lamere, Jeffrey Alexander, François Maillet, Susanna Kirk, Jessica Holt, Jackie Bourque, and Xiao-Wen Mak. Generating transparent, steerable recommendations from textual descriptions of items. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 281–284, New York, NY, USA, 2009. ACM.

[9] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[10] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *In Proc. of EMNLP*, pages 230–237, 2004.