Bidirectional Recommendation Technology for Web Digital Texts

Yuji Wada[1], Syunsuke Matsuzawa[2], Mikuru Yamaguchi[3], Shinichi Dohi[1]
[1]Information Management
Tokyo Denki University
Chiba, Japan
{yujiwada, dohi}@sie.dendai.ac.jp

[2]Mitsubishi Electric Information Systems
Kanagawa, Japan

[3]NTT Data Systems Corporation
Tokyo, Japan

Journal of Digital Information Management

ABSTRACT: *Recently, e-learning systems have begun to be used in companies and education institutions. The spread of e-learning is mainly due to development of the learning management system (LMS), which includes SCORM and WebCT. Therefore, the introduction of learning the web digital web digital texts have become much simpler. The increase of contents of the web digital texts has realized a wider range of learning. As a result, we are concerned that learners will be confused with selecting the best learning web digital texts. In our research, we are developing a bidirectional recommendation system that extracts the relationship among learning the web digital texts with historical logs and recommends an effective web digital text for learners. In this paper, we first discuss the design of the bidirectional recommendation system, and second, we show its evaluation results. Finally, we conclude that our recommendation system is useful for learners.*

Categories and Subject Descriptors
H.3.3 [Information Systems]: Information Search and Retrieval-Clustering, information filtering

General Terms: Recommendation, E-learning, Web digital texts, Data mining

Keywords: Bidirectional recommendation, Collaborative filtering

Received: 11 July 2009; Revised: 28 August 2009; Accepted: 12 November 2009

1. Introduction

For lectures at education institutions and training at businesses, educational methods using e-learning have become widespread, and self-completing learning systems are sometimes seen. Many of the systems now in service are merely for browsing learning texts or for online testing and do not exhibit the advantages of e-learning. With the spread of SCORM [1], which means a unified standard for learning systems, learning history databases, and learning texts, the number of learning texts will increase and make it difficult for learners to select appropriate ones.

Considering these issues, we developed in this research a bidirectional recommendation system optimum for learning by seeking the relationship among learning texts from the historical log information of learning texts used by the learner, or the learning information indicating how the learner learned. We also developed An Individual Reviewing System (AIRS) [2]. This paper gives an outline of AIRS and describes the implementation and evaluation of the bidirectional recommendation system.

We surveyed the state-of-the-art about the recommendation technologies such as collaborative filtering, as follows. In [3], the design and implementation of a recommender system using social networks was described. In [4], a web content recommendation system based on the similarities is proposed. In [5], collaborative filtering based on C-SVM(Support Vector Machine) was proposed examined. In [6], data mining technologies, such as clustering and sequential pattern mining, for online collaborative learning data are studied. In [7], monitoring online tests, such as learner behavior and test quality, through data visualization are discussed. In [8], an automated learning and skills training system for a database programming environment is presented. On the other hand, we developed the new bidirectional recommendation system [9, 10]. In addition, we are currently studying a method for a collaborative learning recommendation system that mines the data of similar users sharing non-favorite subjects using historical logs and user attribute data [11], too.

2. Outline of AIRS

AIRS is a web-based system that distributes review-only learning texts for efficient reviewing by learners. Here, reviewing is defined as “rechecking contents once learned from a lecture, etc., and strengthening the understanding” [12]. Figure 1 shows a schematic of AIRS.

Figure 2 displays the AIRS Japanese top page after a learner, who is going to review the database contents especially selection function, logs in AIRS. This page is comprised of the book marks (located at the upper side) of the contents available with AIRS, the contents menu (located at the left side) corresponding to the selected

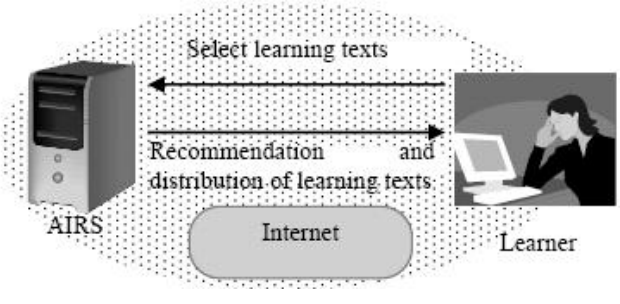

Figure 1. Schematic of AIRS

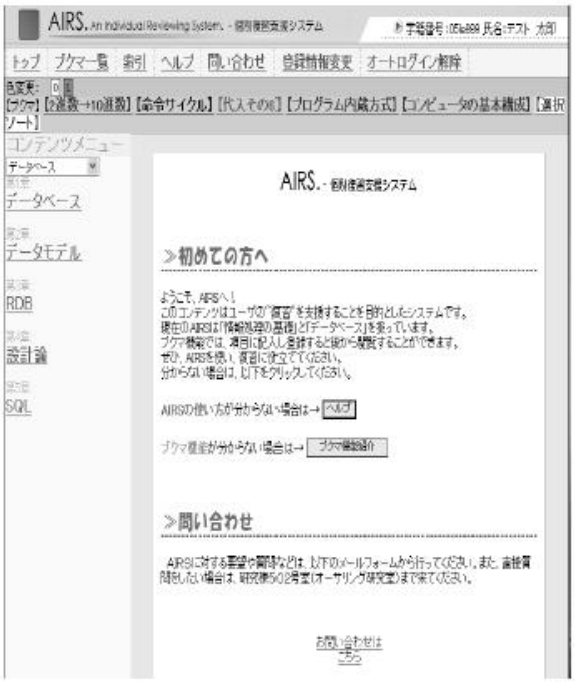

Figure 2. AIRS Japanese top page after login

book mark, and the help messages for beginners (located at the right side). In figure 2, the contents menu displays database, data model, RDB, design methodology, and SQL. The learner selects the book mark such as selection and sorting before the learner can select the corresponding database contents menu. Then, the learner can proceed to revie the database contents.

AIRS uses Java Servlets and JavaServer Pages (JSP). Tomcat5.5 is used for the Servlet container and Apache2 for the web server. MySQL is used for the learning history database of MySQL learners and for the recommendation database as shown in Figure 3.

This AIRS comprises a database server that runs databases, a content server that makes teaching content available, and a system server that runs AIRS. Figure 4 show the flow of texts contents recommendation with AIRS. First, a learner selects the content. Second, the optimum form of expression is requested. Finally, the content is provided from content server to the learner.

2.1 Learning texts

For AIRS, extracts from the courses, “Database System” and “Basics of Information Processing,” at our institution are distributed as learning texts to meet the flow of lessons. Each learning text is composed hierarchically by chapter and section and has three different expression methods for each class. In other words, the composition of the training text may be different even for the same contents.

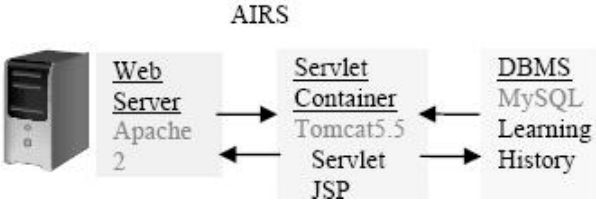

Figure 3. System Configuration of AIRS

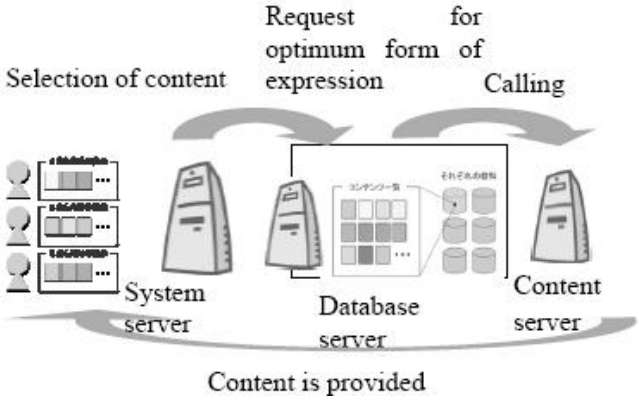

Figure 4. Recommendation flow with AIRS

Expression method 1 may use minimum necessary explanations such as formulas and definitions with figures only, Expression method 2 may use static explanations such as statements and figures, and Expression method 3 may use dynamic explanations using FLASH. For example, Section 8 of Chapter 3 in the database system describes the normal form only in Expression method 1 and attempts the dynamically form in Expression method 3. Table 1 defines the expression methods and Figure 5 shows the hierarchical structure of the learning texts. We call each expression method contents.

Expression method 1	**Formulas, definitions, and minimum necessary figures only**
Expression method 2	Static explanation
Expression method 3	Dynamic explanation

Table 1. Definition of expression methods

2.2 Recommendation of learning texts

AIRS has a learning text recommendation function. This section describes the recommendation function.

2.2.1. Recommendation of expression method

A recommendation technology now in use is collaborative filtering [13]. Collaborative filtering is a technology to analyze user characteristics from historical log data and to recommend information from the historical logs of another user having similar characteristics. For the automatic recommendation function of each expression method, AIRS uses this technology [14]. Suppose that a database has browsing history information indicating which section was browsed by

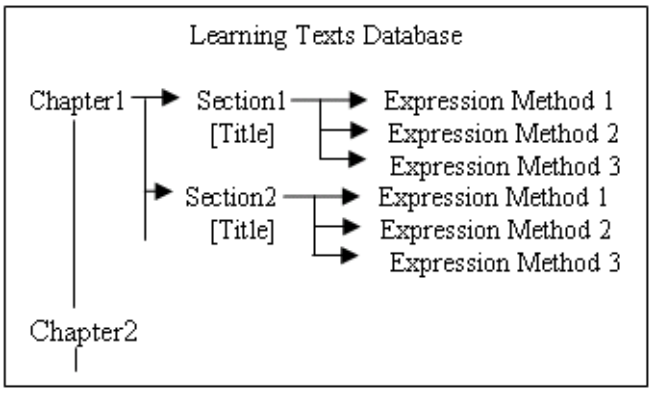

Figure 5. Hierarchical structure of learning texts

Browsing history of Learner A						
Section number	1	2	3	4	5	6
Expression method browsing historical log	0	1	0	1	3	2
Browsing history of Learner B						
Section number	1	2	3	4	5	6
Expression method browsing historical log	0	1	2	1	3	2

Figure 6. Schematic of recommendation by collaborative filtering

Learner A and which expression method was used for this browsing. Then, Learner B, who has the similar browsing history information, is found. Based on this information, an expression method is determined for the recommendation to Learner A. Figure 6 shows a schematic of a recommendation by collaborative filtering. The expression method browsing historical logs 1 to 3 indicates that each expression method was browsed and 0 indicates that the learning text was not browsed. When Learner A browses Learning texts 3, Expression method 2 is recommended by considering that Learner A may very possibly prefer Expression method 2 because Learner B, who has similar characteristics, browsed Expression 2 of Learning text 3.

2.2.2 Recommendation of section

This research examines the function of recommending and displaying a section of strong relationship by calculating the relationship among sections. By naming this as a "bidirectional recommendation system," we proposed and developed its algorithm.

3. Bidirectional recommendation system

AIRS is not a self-completing system because it supports efficient reviewing. This system is developed based on a concept that it grasps a general idea quickly and provides efficient reviewing by selecting exact points that need to be clarified and checked. However, learners strongly tend to browse learning texts sequentially from the first one and follow along a learning flow. For example, learning texts with as many as 210 sections may also be a factor that makes learners follow a sequence. Therefore, we think efficient reviewing is possible by presenting sections highly related to the current reviewing section if the sections related to the current browsing section can be recommended.

3.1 Recommendation types

As typical recommendation systems, reference [15] introduces the following two types of recommendations:

1. Recommendation by user relationship.

2. Recommendation by item relationship.

Recommendation by user relationship is making a recommendation from the viewpoint of the user relationship by analyzing the action history of each user. This is called user correlation and is used for the aforementioned collaborative filtering.

Recommendation by item relationship is making a recommendation from the viewpoint of the item relationship. This is called item correlation. These two recommendations can be classified into the following two analysis techniques:

1. Correlation based on the user's action history.
2. Correlation based on the item's information.

For bidirectional recommendation, the relationship among sections is analyzed by using the former correlation based on the user's action history.

3.2 Necessity of bidirection

The recommendation engines now in use for collaborative filtering are mainly used to recommend commodities. Therefore, there is no need to recommend having already purchased ones. In learning, however, not only learners advance from the current learning text to the next, but also learners return to the previous learning text. Therefore, we think it necessary to make not only a unidirectional recommendation that recommends new contents only but also a bidirectional recommendation that also recommends the previously browsed contents.

3.3 Outline of bidirectional recommendation

When a learner is browsing the learning text "Basics of substitution," it is natural to advance to the next step "Basics of 'while' statement" or "Basics of 'if' statement." However, browsing the basic contents "Variable types" again is also natural in learning. In other words, the learning efficiency is expected to improve by recommending not only learning texts frequently shifted from but also frequently shifted to "Basics of substitution." Figure 7 shows a schematic of a bidirectional recommendation.

3.4 Bidirectional recommendation algorithm

To realize bidirectional recommendation, a binary tree is created with each section as the root. Sections highly related to a section are linked as subnodes to form a small-scale tree up to Level 2. With each section as the root, trees for as many as all sections are created under the tree. The tree, different from the aforementioned hierarchical structures of learning texts, indicates the relationship among sections. By referencing this tree, highly related sections before and after are found and recommended.

3.4.1 Tree configuration

This tree for bidirectional recommendation is created from historical log data in which the shift of learning texts by each learner is recorded. The log data integrated from learners is called the learning text shift history. Figure 8 shows this integration.

To extract sections highly related to a section from this learning text shift screen, we apply the user count algorithm used in the web recommendation system PHOAKS by Terveen [16] et al. This is used to find an item used by many users. Sections where a learner shifted from a section can be considered as

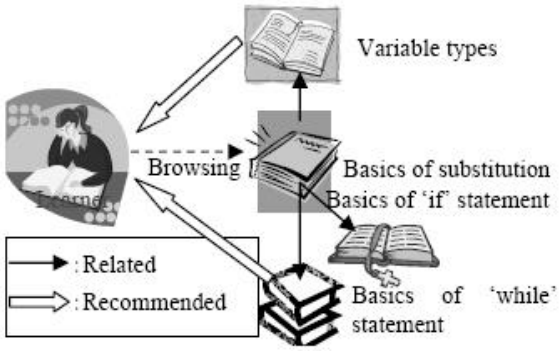

Figure 7. Schematic of bidirectional recommendation

Learning text shift history

user_id	access_time	content_id1	content_id2	content_id3
219	20080121205205	0	139	142
219	20080121205340	139	142	145
150	20080121205409	0	0	328
150	20080121205718	0	328	310
219	20080121205901	142	145	157

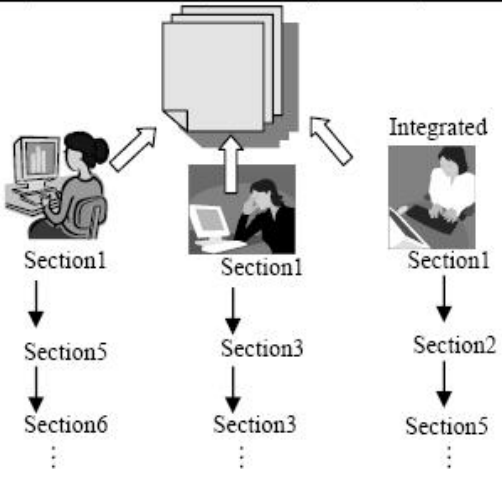

Figure 8. Integration of historical log data

highly related. The following Equation (1) is used to calculate the degree of the relationship with a learning text from the learning text shi history:

$$p_{a,j} = \frac{\sum_{u_i \in U} binaryOf(q_{i,j})}{|U|} \quad (1)$$

where

$$binaryOf(q_{i,j}) = \begin{cases} 0(q_{i,j} = 0.0) \\ 1(q_{i,j} > 0.0) \end{cases} \quad (2)$$

If the frequency of shift from learning text U_i to U_j is represented as $q_{i,j}$, the degree of relationship $p_{a,j}$ of U_j with U_a can be calculated. The right side of Equation

1. becomes equal to the ratio of learners who shifted to learning text U_a.

A binary tree is created with the top two sections shifted from Section 1 on Level 1 and the top two sections shifted from there on Level 2. Repeating this on every section produces as many trees of relationship as the sections. Figure 9 shows a tree indicating the relationship of the section "Function."

3.4.2 Recommendation method using tree

A bidirectional recommendation is made by using the tree mentioned in the preceding section. When a learner is browsing the section "Function," the system refers to the tree of the relationship shown in Figure 9. The system determines the sections

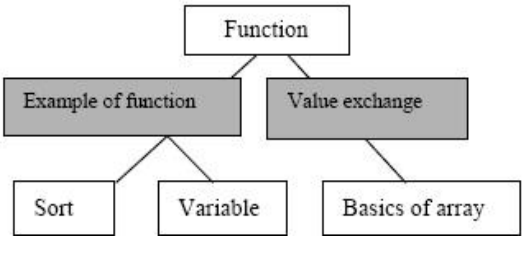

Figure 9. Sample tree indicating a relationship

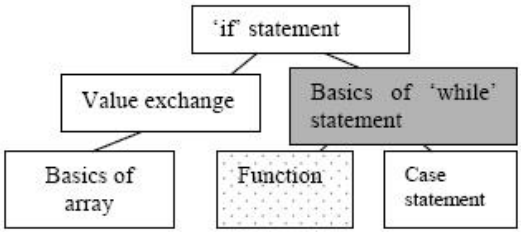

Figure 10. Recommendation from a tree containing the section "Function"

of "Example of function" and "Value exchange" most possibly related to this tree as recommendation items.

Next, all tress are searched to find the trees containing the section "Function." Here, the section "'if' statement" shown in Figure 10 is found. From this tree, the section "Basics of 'while' statement" is determined as a recommendation item because it is found highly related to the section "Function." Figure 10 shows this recommendation and gives a sample tree of the section "'if' statement."

In addition, when a tree containing the section "Function" is found, "Function" is also determined by similar processing as a recommendation item.

The system presents the learner with the recommendation items determined by the above processing. This enables the recommendation of a highly probable shift to sections before and after the current learning section. In other words, the contents to be learned next and reviewed can be recommended in two directions. The system of recommending related items before and after was named "bidirectional recommendation system."

Figure 11 shows the general flowchart of the bidirectional recommendation. Through the above flow, sections to be presented to a learner browsing the section "Function" as related pages are determined.

Up to four pages can be presented. Figure 12 shows the presentation of related Japanese homepages in our developed recommendation system. Precisely looking at figure 12, four recommended items such as example of function, value exchange, basic of `while` statement, and inds of variabes are displayed.

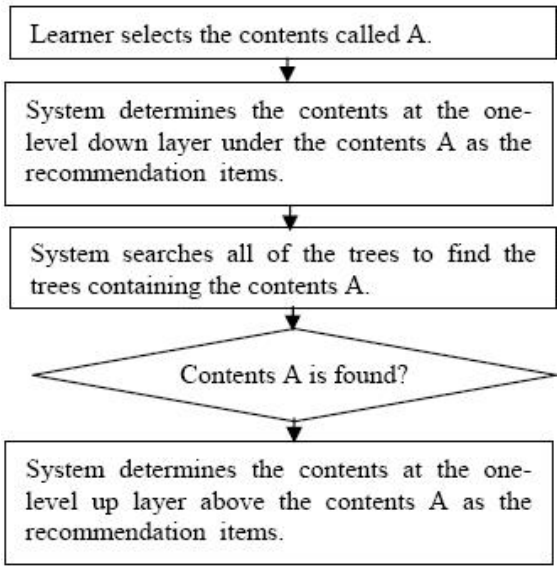

Figure 11. Flowchart of bidirectional recommendation system

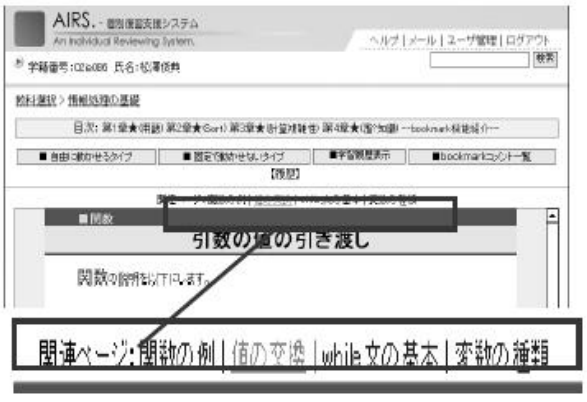

Figure 12. Presentation of related Japanese homepages

4. Evaluation

This section compares the recommendation results available from the log data of learning texts arranged along the lesson flow with that from the log data of learning texts arranged at random. Thus, we verified the validity of the related pages presented by the bidirectional recommendation system and obtained the following results.

4.1 History log data for verification

In our university department, the course "Database system" was held for sophomores in September 2007. A total of 125 students who took the course used AIRS. The acquisition of learning text historical log data began on September 19, 2007. A total of 7355 historical logs acquired by the end of the course on December 27, 2007 were used for verification. Since the system is in actual operation, the links of learning texts are arranged along the course flow. The tree of relationship created from the historical log data are used as Experimental data 1.

To acquire a learning text shift history for comparison, a total of 24 people used AIRS (one postgraduate, one research student, 22 undergraduates (13 seniors and 9 juniors)). In the period of use from January 8, 2008 until January 18, 2008, AIRS was used for approximately 10 minutes a day for learning. For comparison, the links of learning texts were arranged at random, totally irrespective of the course flow. The total number of historical logs reached 768. The group of trees created from the historical log data is called Experimental data 2. Table 2 summarizes the experimental data.

	Experimental data 1	Experimental data 2
Period of use	Sept. to Dec. 2007	Jan. 2008
User	125 students	24 students
Log count	7355	768
Arrangement	Along course	At random

Table 2. Summary of experimental data

Table 3 shows part of the tree stored in a database. The content_id represents a section, layer1_1 and layer1_2 represent the first layer of the tree, and layer2_1 to layer2_4 represent the second layer.

content_id	layer1_1	layer1_2	layer2_1
14	20	23	11
15	173	10	4
16	19	12	1

Table 3. Tree stored in a database

4.2 Precision of recommendation

By using the two tree groups mentioned in the previous section, bidirectional recommendations were made and the recommendation results were compared. The object learning texts here are composed of 58 sections from Chapters 1 and 3 of "Database system." The recommendation using Experimental data 1 presented 152 related pages and that using Experimental data 2 presented 212 related pages. The recommendation using Experimental data 2 proposed more related pages because various shift historical logs could be acquired due to the random arrangement of learning texts.

As shown in Table 4, when the recommendation results using Experimental data 2 were compared with those using Experimental data 1, the number of the same recommendation items were 32 and the hit ratio of the items against the related pages using the Experimental data 2 was 15.1% (=32/212). The number of sections containing at least one hit was 25 and the average ratio of the hit against the entire sections was 43.1% (=25/58).

	Complete Hit	Approximate Hit
Count (page)	32	74
Ratio (page)	15.1%	34.9%
Count (section)	25	45
Ratio (section)	43.1%	77.6%

Table 4. Number and ratio of hits

Then, recommendations of not complete but approximate hits were also compared as shown in Table 4. These can be considered to be identical recommendations, such as "key insertion into B+ tree," identical to "key deletion from B+ tree," and "union operation," identical to "difference operation." The number of hits under this condition was 74 and the ratio was 34.9% (=74/212). The number of recommendations containing at least one hit was 45 and the average ratio was 77.6% (=45/58).

We measured the number of the recommended contents with the bidirectional recommendation system on the base of the flow of lessons previously mentioned. Table 5 shows the number of the recommended after pages and the recommended before pages using Experimental data 1 and Experimental data 2. In addition, Table 5 indicates the ratio of the recommended after pages and the recommended before pages, too.

	Experimental data1	Experimental data2
Count (after)	80	85
Ratio (after)	52.6%	40.0%
Count (before)	72	127
Ratio (before)	47.4%	60.0%

Table 5. Number and ratio of the bidirectional recommendations

In case of Experimental data 1, the ratio of the recommended before pages is 47.4 %, which is less than that of the after ones. On the other hand, in case Experimental data 2 the ratio of the recommended before pages is 60.0 % , which is greater than that of the after ones.

4.3 Discussion

The comparison this time may not be very accurate because the counts and periods of experimental data were different. By comparison, however, we could at least confirm the ratio of almost equally recommended sections to be as high as 77.6%. In addition to that, we found that the bidirectional recommendation system realizes the number of the recommended before pages is almost equal to that of the recommended after pages, such as 50 to 60%.

Collaborative filtering or any other recommended engine using personal historical log data cannot make an optimum recommendation to the first-time user. A recommend engine that calculates the item relationship from the historical log data of all users makes a recommendation that can present the same result even for the first-time user. The bidirectional recommendation system using this method can recommend learning texts appropriately to all learners if there is historical log data about the learning texts. We show the comparison results of the bidirectional recommendation system and collaborative filtering in the Table 6.

	Bidirectional recommendation	**Collaborative filtering**
Recommended items	Sections	Expression methods
Recommendation criterion	Associations between contents	Associations between user log
Recommendation results	Common to all learners	Different to each learner
Required log data	All learners' log data	Individual log

Table 6. Comparison of bidirectional recommendation and collaborative filtering

5. Conclusion

By experiments, we could prove the bidirectional recommendation system capable of making an appropriate recommendation, regardless of the arrangement of learning texts. The bidirectional recommendation system not requiring personal historical log data was also proved effective for recommending learning texts.

We currently consider the following three future research subjects. The first subject is to verify that learning texts can be recommended appropriately to learners by the recommendation of sections by combining the bidirectional recommendation and the recommendation of expression using collaborative filtering. The second subject is to develop the analysis method of the recommendation evaluation results. The third subject is to compare our approach with existing data mining techniques such as the correlation between users (resp. items) could be extracted by association rules or sequential pattern approaches.

6. Acknowledgments

We would like to express our deepest gratitude to the subjects of the experiment and to the many students who used AIRS.

This research was partially supported by a Grant-in-Aid for Scientific Research C (Subject No. 18500731: Research on Learning Texts Recommendation Technology for E-Learning) and is partially supported by a Grant-in-Aid for Scientific Research C (Subject No. 21500908: Research on Adaptive Recommendation Technology based on Bi-directional Recommendation Technology for E-Learning Texts).

References

[1] R. Wisher, "Sharable Content Object Reference Model (SCORM) 2004 4th Edition Overview Version1.0", @ Advanced Distributed Learning, pp.1-18 (31 March 2009).

[2] M. Yamaguchi, Y. Takahashi, T. Matsuzawa, T. Takahashi, and S.Dohi, "Development and Evaluation of e-Learning System AIRS with Multi-Expression Contents", *IEICE-ET2006-118* , vol.106, no.583, pp. 69-74 (2007).

[3] H. Hotta, and M. Hagiwara, "Design and Implementation of Social-network-based Recommender System", *IPSJ Transactions on Databases*, vol. 2, no.1, pp. 46-56 (2009).

[4] A. Sasaki,T. Miyata,Y. Inazumi, A. Kobayashi, and Y. Sakai, "Web Content Recommendation System Based on Similarities among Contents Cluster of Social Bookmark", *IPSJ Transactions on Databases*, vol. 48, no.SIG20, pp. 14-27 (2007).

[5] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura, "Context-Aware SVM for Context-Dependent Information Recommendation", *DBSJ Letters,* vol. 5, no.1, pp. 5-8, (2006).

[6] D. Perea, J. Kay, I. Koprinska, K. Yacef, and O.R. Zaiane, "Clustering and Sequential Pattern Mining of Online Collaborative Learning Data", *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, no.6, pp. 759-772 (2009).

[7] G. Costagliola, V. Fuccela, M. Giordano, and G. Polese, "Monitoring Online Tests through Data Visualization", *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no.6, pp. 773-784 (2009).

[8] C. Pahl and C. Kenny, Interactive Correction and "Recommendation for Computer Language and Training", I*EEE Transactions on Knowledge and Data Engineering*, vol. 21, no.6, pp. 854-865 (2009).

[9] Y. Wada, T. Matsuzawa, M. Yamaguchi, and S. Dohi, "Proposal and Verification of Bidirectional Recommendation System for Learning Web Digital Texts", *Proceedings of the Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT2009)*, pp. 210-215 (August 2009).

[10] T. Matsuzawa, M. Yamaguchi, Y. Wada, and S.Dohi, "Development of Bidirectional Recommendation System for Texts", *2007 PC Conference*, pp. 197-198 (2007-8).

[11] Y. Wada, Y. Hamadume, S. Dohi, and J. Sawamoto, "Technology for Recommending Optimum Learning Texts Based on Data Mining of Learning Historical Data", *Proceedings of International Workshop on Informatics (IWIN 2009)* (September 2009) (to be appeared).

[12] Y. Takahashi, T. Matsuzawa, M.Yamaguchi, S. Dohi, and Y. Wada, "Recommendation and Delivery of the Optimum Texts for e-Learners", *IPSJ SIG Technical Reports 2007-CE-88(22)*, vol. 207, no.12, pp.157-162 (2007).

[13] N. Ohsugi, A. Monden, S. Morisaki, and K. Matsumoto, "Software Function Recommender System Based on Collaborative Filtering", *IPSJ Journal,* vol. 45 Cno.1, pp. 267-278 (2004).

[14] T.Matsuzawa, M.Yamaguchi, Y.Wada, and S.Dohi, “Implementation of recommend system for learning texts”, I*PSJ SIG Technical Reports 2008-CE-93(18)*, vol. 208, no.13, pp.127-132 (2008).

[15] N.Yamazaki, *Software Design ISSN 0916-629, Vol. 268* CGijutsu-Hyohron Co., Ltd CTokyo Japan, 2007-8.

[16] L.Terveen, W. Hill, B. Amento, D. Mcdonald, and J. Creter, “PHOAKS: a system for Sharing recommendations”, *Commnucations of the ACM*, vol. 40, no3, pp. 59-62(1997).