WEBLORS – A Personalized Web-Based Recommender System

Mohammad Belghis-Zadeh[1], Hazra Imran[2], Maiga Chang[1], and Sabine Graf[1]

[1] Athabasca University, Edmonton, AB, Canada
mobelghis@yahoo.ca, maigac@athabascau.ca, sabineg@athabascau.ca
[2] University of British Columbia, Vancouver, BC, Canada
hazra@cs.ubc.ca

Abstract. Nowadays, personalization and adaptivity becomes more and more important in most systems. When it comes to education and learning, personalization can provide learners with better learning experiences by considering their needs and characteristics when presenting them with learning materials within courses in learning management systems. One way to provide students with more personal learning materials is to deliver personalized content from the web. However, due to information overload, finding relevant and personalized materials from the web remains a challenging task. This paper presents an adaptive recommender system called WEBLORS that aims at helping learners to overcome the information overload by providing them with additional personalized learning materials from the web to increase their learning and performance. This paper also presents the evaluation of WEBLORS based on its recommender system acceptance using data from 36 participants. The evaluation showed that overall, participants had a positive experience interacting with WEBLORS. They trusted the recommendations and found them helpful to improve learning and performance, and they agreed that they would like to use the system again.

Keywords: Recommender systems, Web mining, Personalization

1 Introduction

Although different students have different needs, learning management systems (LMSs) usually have fixed content that is presented to all students in the same way [1]. However, these systems can be enriched with personalization through recommender systems (RS). To date, many RSs are limited to recommend the available learning objects (LOs) that either have been created in the course, which greatly limits the variety of the recommendable objects, or have been collected in LO repositories (LOR) [2]. Using LORs provides RSs with access to a larger pool of LOs, however, the quality of recommendations is highly impacted by the quality of the metadata that was provided by users who created the objects [3]. Moreover, the available pool of LOs in a LOR could still be limited based on topics and types of LOs. However, there are more LOs and learning materials openly and freely available on the web that can be targeted by RSs [4]. However, due to the vast number of these objects on the web, different techniques need to be utilized to overcome the information overload and find relevant and personalized learning materials that fit students' needs [5].

In this paper, we introduce WEBLORS, a recommender system that aims at helping students by considering their individual needs and the ratings given by other learners to present the learner with additional learning materials from the web that are relevant to the learner and the topic he/she is currently learning. This paper also presents the evaluation of WEBLORS based on its recommender system acceptance.

The remainder of this paper is structured as follows: Section 2 presents related work. Section 3 discusses WEBLORS' architecture and approach. Section 4 explains the evaluation methodology and the results of the WEBLORS' evaluation and, section 5 concludes the paper.

2 Related Work

The idea of RSs in the learning domain has been around for decades and different recommendable objects such as courses, learning materials and academic papers have been targeted [6]. However, most literature in this area has been about LO recommendations, and one of the new research trends for LO recommendations is to broaden the search and recommend LOs from web-based LORs, social networks or even from the web. There are different ways how RSs decide which LOs to recommend. Many RSs recommend learning content based on users' past activities [7, 8]. For example, Dahdouh and colleagues proposed a recommender system that generates recommendations by considering learners' historical data as a factor and finds similarities between learners past activities collected from system logs [7]. Another example is the system built by Bourkoukou and colleagues that generates recommendations for learners based on the user's historical data collected from server logs and other attributes of learners [8]. Some other systems generate recommendations based on the keywords that are passed by the users [9-11]. For example, the RS built by Zapata and colleagues considers the keywords that are specified by a user and finds relevant LOs from a LOR called AGORA [9]. Similarly, Atkinson and colleagues proposed a system that accepts the queries as input from users and uses focused crawling and metadata extraction to find relevant web resources [10]. Rahman and colleagues also proposed a group-based recommender system that accepts users' queries, considers users' profiles, and uses Google search engine to recommend learning materials to learners based on their profiles [11]. After reviewing the existing literature, we identified some gaps for RSs in education that we addressed in our system. First, WEBLORS recommends LOs from the web and therefore, aims at advancing our knowledge in this new trending area. Second, many RSs consider past activities of learners as a major factor when generating recommendations. Therefore, cold start is a problem in these systems. To address this issue, WEBLORS does not rely on users' past activities and instead uses learners' learning styles, the opinions of other learners (if available) and the topic that is being studied. Third, many RSs with a broad search space often work similar to search engines and heavily rely on the search criteria that are passed by the users. In WEBLORS, this issue is avoided by creating keywords automatically through extracting them from the content that a learner is learning.

3 Architecture of WEBLORS

WEBLORS consists of two main parts that are shown in Fig 1 and Fig 2 and are further described in the next two subsections.

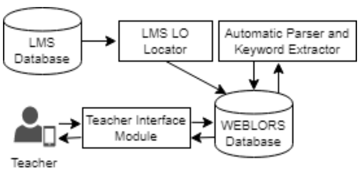

Fig. 1. Architecture of CLOA

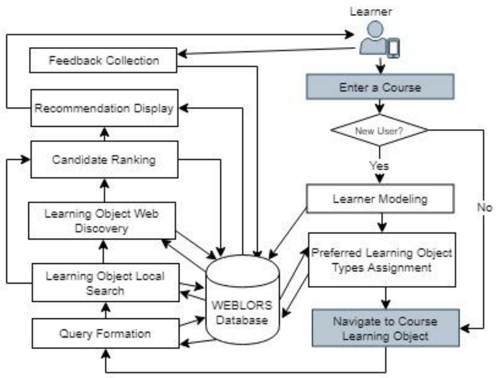

Fig. 2. Architecture of ALORS

3.1 Course LOs Analyzer (CLOA)

As shown in Fig 1, CLOA contains a set of modules and components. The aim of the **LMS LO Locator** module is to locate all LOs within the LMS and extract their content. As part of the installation process of WEBLORS, this module searches through each course and LO in the LMS database and stores its content and the searchable criteria into the WEBLORS database (DB). Also, when a new LO is added to the LMS by a teacher, this module stores the content and the searchable criteria of the newly added LO into the DB. The aim of the **Automatic Parser and Keyword Extractor** module is to parse the content of each LO, extract a set of candidate keywords and store the keywords into the DB. This module uses the RAKE algorithm [12] to discover the keywords and key phrases that best fit the LO. The aim of the **Teacher Interface** module is to display each LO and its extracted keywords to the instructor where he/she can confirm the accuracy, relevance and the importance of the keywords or overwrite them with a set of new keywords if required.

3.2 Adaptive LO Discovery and Recommender System (ALORS)

As shown in Fig 2, ALORS contains several modules and activities. The aim of the **Learner Modeling Module** is to capture learners' learning styles (LSs) based on Felder-Silverman Learning Style Model [13], a widely known and commonly used LS model. Based on this model, learners are classified in four dimensions: (1) active/reflective (Act/Ref), (2) sensing/intuitive (Sen/Int), (3) visual/verbal (Vis/Ver) and (4) sequential/global (Seq/Glo). This module uses a questionnaire called Index of Learning Styles (ILS) [14] that contains 44 questions. ILS was developed by Felder and Soloman and was found to be valid, reliable and suitable for identifying LSs [15]. ILS is presented to each user when he/she enters his/her first course for the first time and based on the provided answers, his/her LSs are calculated as four numeric values (each for one LS dimension). This module then builds a profile *(sp)* for each student *(s)* which is represented as a vector of 8 elements and is formed as *sp(s)*=(Act, Ref, Sen, Int, Vis, Ver, Seq, Glo). In *sp*, each LS dimension is represented with 2 elements where each element has a value between 0 and 2, representing the strength of the LS preference.

The aim of the **Preferred Learning Object Types Assignment** is to identify a set of preferred LO types (PLOTs) and their associated keywords for each learner based on their LSs. This module uses a mapping table (Table 1) that has been created based on the mapping proposed by El-Bishouty and colleagues [16] and has been extended with the LO type of videos that according to Felder and Silverman is suitable for visual and verbal learners [13]. In this module, each LO type (*lot*) is represented by a LO type profile (*lp*) which is a vector with the same 8 elements as the *sp*. Each element of *lp* is either 0 or 1 and is assigned per Table 1, indicating whether (1) or not (0) the LO is beneficial for that LS.

Table 1. Mapping Table (based on [16])

LO Type	LO Type Keyword	Act	Ref	Sen	Int	Vis	Ver	Seq	Glo
Exercises	exercise 1	1	0	1	0	0	0	0	0
Examples	example 1	0	1	1	0	0	0	0	1
Real Life Application	real world application	0	0	1	0	0	0	0	1
Video	video	0	0	0	0	1	1	0	0
Self-Assessment Test	questions and answers	1	0	1	0	0	0	0	0
Additional Reading Material	pdf	0	1	0	1	0	1	1	0

Next, this module calculates a numeric value for each LO type that is called **Relevance** value (Rel(*s*,*lot*)) which is the scalar product of *sp(s)* and *lp(lot),* and is used to determine the most preferred LO types for a given student with a certain LS. All LO types that have a positive Rel(*s*,*lot*) form the student's preferred LO types (PLOT).

The aim of the **Query Formation Module (QFM)** is to take the previously extracted keywords from the LO that the student is currently visiting and the LO type keywords associated with each PLOT of the student (per Table 1) as input and form one query per PLOT. WEBLORS considers three different categories of LOs when generating recommendations: (1) course LOs, (2) local LOs and (3) web LOs. Course LOs are objects that are created by the teacher and are part of the course. Local LOs are the objects that have been previously discovered from the web, recommended to learners and stored in the DB. Web LOs are the objects that are discovered from the web for the first time.

The aim of **Learning Object Local Search Module (LOLSM)** is to select a set of local LOs for each query that has been formed by the QFM and mark them as candidate local LOs and pass them to the Candidate Ranking module for further processing. Local LOs are considered to be a candidate local LO if they are of a LO type that the given query has been created for and satisfy one of the following conditions: (1) local LOs that have been previously rated (with values between 1 and 5) by five or more users and the weighted average rating for them (WAvg(*lo*)) is greater than or equals to 3.5 out of 5 (i.e., ≥ 70% of agreement) or (2) all local LOs that have been rated less than five times (to give enough chance to new local LOs to be recommended and rated by users).

The **Learning Object Web Discovery Module (LOWDM)** aims at using the Google API to execute the queries that are created by the QFM on the web and finding the candidate web LOs. To ensure that only educational materials are being targeted, a new Google Custom Search Engine (CSE) was created and configured to only target learning resources, scholarly articles and educational materials on the web. Also, to narrow down the search and control the number of results that are returned from the web, this module appends the index of the first result that should be returned (*start*) and the number of results that should be retrieved (*num*) to each query before running them. Both *num* and *start* parameters can be configured. The *num* parameter is set to 5 by default to enforce the query to return only 5 results at a time. In order to find at least one web LO that has not been recommended before, the *start* parameter is used in a way that if all 5 LOs that are returned by the query exist in the DB, then the system increases the *start* parameter by 5, reruns the query and returns the next 5 results until at least one new LO is found in those 5 results. At this point, the 5 results are checked and those that have not been previously recommended to any user (1 to 5 web LOs) are considered as candidate web LOs and are passed to the Candidate Ranking module. This process is repeated for each query so that there are 1-5 web LOs passed to the Candidate Ranking module for each query.

The aim of the **Candidate Ranking Module** is to accept the candidate local and web LOs from the LOLSM and LOWDM as input and decide which of them should be recommended to the learner. To generate the list of recommendations for a given student (*s*), this module calculates an **Importance** value (Imp*(lo)*) as the scalar product of the relevance value (Rel*(s,lot)*) and the weighted average rating for each candidate LO (WAvg*(lo)*). A default value of 2.5 (average rating) is used as WAvg(*lo*) for web LOs and the local LOs with less than five ratings. Next, all candidate LOs are ranked in ascending order in a way that the candidate LO with the lowest Imp(*lo*) gets the rank of 1. Subsequently, the Fitness Proportionate Selection algorithm (FPS) [17] is used to select the recommendable objects in a way that the LOs with a higher Importance value have higher chance to be selected, but LOs with lower Importance value still have a small chance to be recommended. In order to select *N* candidate LOs where *N* is the number of LOs that should be recommended to the student, FPS is applied *N-1* times. Next, the list of already selected LOs is checked. If at least one LO from the web is already selected, FPS is applied one more time. Otherwise the web LO with the highest Importance value is selected as the *N*th LO.

The aim of the **Recommendation Display Module** is to accept the recommendation list from the Candidate Ranking module and display them to the learner. Also, a five-star rating system is presented for each recommended LO where the learner can rate the quality of the recommendation. The aim of the **Feedback Collection Module** is to collect the ratings that were provided by the users and store them in the DB.

4 Evaluation

In this section, the methodology used to evaluate the users' acceptance of the system is introduced. The research design, participants selection, and the results are explained in the next three subsections.

4.1 Research Design

For this evaluation, WEBLORS was integrated into an instance of Moodle [18] and a sample course on the topic of Data Presentation in Computers was created that contained 5 LOs. Also, a four-step process was designed and published on the evaluation website where participants were asked to complete the following tasks: (1) watch a video that contains a demo of the system, (2) complete a pre-test that contains 9 questions about the course topic and one trick question, (3) login to the course, fill out the ILS, read and learn each of the LOs, and read, learn and rate the generated recommendations (5 recommendations are generated for each LO), (4) complete a post-test, which consisted of the same questions as the pre-test and can demonstrate a students' knowledge increase, (5) complete a feedback questionnaire that contains one trick question and 6 multiple-choice questions (created based on [19] and [20]) where users could rate their experience on a scale of 1 (strong disagreement) to 5 (strong agreement). Questions 1 to 6 are listed in table 2.

4.2 Participants Selection

For this evaluation, a new task was created on Amazon Mechanical Turk and 95 users accepted the task. To ensure that only valid data is included in the analysis, the following acceptance criteria were defined. Users should have completed all steps of the evaluation, answered all trick questions correctly, read at least 3 out of 5 LOs in the course, read and rated more than one third of the generated recommendations (9 or more out of 25), and spent at least 35 minutes on the sample course. Based on our assessment, at minimum, 35 minutes are required to complete the ILS, read at least 3 out of 5 LOs and 9 out of 25 recommendations and complete the post test. Although extracted times spent gathered from data logs might not be the exact time that users spent on the resources, it still provides valuable insights into the reliability of the collected data. After validating the collected data, responses from 36 participants (out of 95) met the acceptance criteria, and the rest were excluded from the evaluation.

4.3 Results

In order to analyze the data, the answers given to the 6 multiple-choice questions (Q1 -Q6) by the 36 accepted participants were aggregated. Each question has five possible answers that are shown in table 2 with respective scores provided in brackets. In addition, the weighted average score was calculated for each question. Based on the results shown in table 2, very high average scores have been given to Q1, Q2, Q3, Q4 and Q6 indicating that overall users agreed with the statements in these questions. These scores show that most users trusted the recommendations, found the system very useful, and believed that this system can increase learners' performance and help them in their learning process. In addition, users stated that they like to use WEBLORS frequently and have such system available to them while studying other courses. Q5 was a negative question and the low score that was given to this question shows that on average users disagreed with the statement in this question and believed that WEBLORS does not put much extra work on users to provide ratings.

Table 2. Results of quantitative analysis

Question	Total	Strongly Agree (5)	Agree (4)	Neither Agree nor Disagree (3)	Disagree (2)	Strongly Disagree (1)	**Weighted Average Score**
Q1- I would like to use WEBLORS frequently	36	16	16	1	2	1	4.22
Q2- I would like to see such recommendations in other courses as well	36	16	16	3	1	0	4.31
Q3- I trusted the recommendations provided by WEBLORS	36	21	13	1	1	0	4.50
Q4- I think recommendations provided by WEBLORS will be helpful in increasing students' performance	36	20	15	1	0	0	4.53
Q5- I think WEBLORS will put extra work on students for providing ratings	36	4	2	4	6	20	2.00
Q6- I think recommendations provided by WEBLORS will be helpful in increasing students' learning	36	17	17	1	1	0	4.39

5 Conclusion

The focus of this paper is on explaining the architecture of WEBLORS as well as the evaluation of the system in terms of recommender system acceptance. WEBLORS is a RS that considers the topic that the learner is learning as well as the ratings of LOs given by other learners and provides the learner with relevant learning materials from the web that are beneficial for him/her based on his/her LSs. Recommended materials are selected from a set of relevant LOs that are either discovered from the web for the first time or have been previously recommended to other learners and were given high ratings (or have been rated by less than 5 users), with the condition that at least one new LO from the web is recommended every time that WEBLORS generates recommendations. The results of the evaluation show that the 36 users provided promising feedback with respect to recommender system acceptance. Based on the result, users like to use WEBLORS frequently and are interested to have such system available to them in other courses as well. Also, users trusted the generated recommendations and believed that the provided recommendations can help students in their learning process and will have a positive impact on students' performance. Also, the results show that most users believe that asking users to rate the recommendations does not add lots of overhead and does not put much extra work on students. To conclude, the results show that WEBLORS fills a gap in LMSs by recommending extra personalized learning materials from the web and helping with information overload by only recommending LOs relevant to the topic that is being studied and which fits students' LSs. Future work will deal with evaluating the system further based on other aspects such as ease of use, user friendliness, knowledge increase of users after using WEBLORS, and others. In addition, future work will deal with the broad use of the system in different courses.

Acknowledgment. The authors acknowledge the support of Athabasca University, Alberta Innovates – Technology Futures (AITF), Ministry of Advanced Education of Canada, the National Science and Engineering Research Council of Canada (NSERC) [funding reference number: 402053-2012-RGPIN], and Mitacs.

References

1. Imran, H., Belghis-Zadeh, M., Chang, T.-W., Kinshuk and Graf, S.: PLORS: a personalized learning object recommender system. Vietnam Journal of Computer Science 3(1), 3-13 (2016)
2. Dwivedi, P. and Bharadwaj, K.: e-Learning recommender system for a group of learners based on the unified learner profile approach. Expert Systems 32(2), 264-276 (2015)
3. Sabourin, J., Kosturko, L. and McQuiggan, S.: Where to next? A comparison of recommendation strategies for navigating a learning object repository. In: Proceedings of the International Conference on User Modeling, Adaptation, and Personalization, pp. 208-215. Springer, Cham (2015)
4. Al Abri, M. and Dabbagh, N.: Open educational resources: A literature review. Journal of Mason Graduate Research 6(1), 83-104 (2018)
5. Akhtarzada, A., Calude, C. S. and Hosking, J.: A multi-criteria metric algorithm for recommender systems. Fundamenta Informaticae 110, 1-11 (2011)
6. Manouselis, N.: Recommender systems for learning. Springer, New York (2013)
7. Dahdouh, K., Oughdir, L., Dakkak, A. and Ibriz, A.: Building an e-learning recommender system using association rules techniques and R environment. International Journal of Information Science & Technology 3(2), 11-18 (2019)
8. Bourkoukou, O. and Bachari, E. E.: Toward a hybrid recommender system for e-learning personnalization based on data mining techniques. International Journal On Informatics Visualization 2(4), 271-278 (2018)
9. Zapata, A., Menéndez, V. H., Prieto, M. E. and Romero, C.: A framework for recommendation in learning object repositories: An example of application in civil engineering. Advances in Engineering Software 56, 1-14 (2013)
10. Atkinson, J., Gonzalez, A., Munoz, M. and Astudillo, H.: Web metadata extraction and semantic indexing for learning objects extraction. Applied Intelligence 41(2), 649-664 (2014)
11. Rahman, M. M. and Abdullah, N. A.: A personalized group-based recommendation approach for web search in e-learning. IEEE Access 6, 34166-34178 (2018)
12. Rose, S., Engel, D., Cramer, N. and Cowley, W.: Automatic keyword extraction from individual documents. In Text Mining, pp. 1-20, John Wiley & Sons Ltd (2010)
13. Felder, R. and Silverman, L.: Learning and teaching styles in engineering education. Engineering Education 78(7), 674-681 (1988)
14. Felder, R. M. and Soloman, B. A.: Index of learning style questionnaire, https://www.webtools.ncsu.edu/learningstyles/, last accessed July 20, 2019.
15. Felder, R. and Spurlin, J.: Applications, reliability and validity of the index of learning styles. International Journal of Engineering Education 21, 103-112 (2005)
16. El-Bishouty, M. M., Saito, K., Chang, T.-W., Kinshuk and Graf, S.: An interactive course analyzer for improving learning styles support level. In: Proceedings of the Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data, pp. 136-147. Springer, Berlin, Heidelberg (2013)
17. Bäck, T.: Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York, NY (1996)
18. Dougiamas, M.: Moodle, http://moodle.org/, last accessed July 20, 2019.
19. Brooke, J.: SUS: A quick and dirty usability scale. In P.W. Jordan, et al. (eds.) Usability Evaluation in Industry, pp. 189-194. Taylor and Francis, London (1996)
20. Lewis, J. R. and Sauro, J.: The factor structure of the system usability scale. In: Proceedings of the International Conference on Human Centered Design, pp. 94-103. Springer, Berlin, Heidelberg (2009)