Personalized Navigation Recommendation for E-book Page Jump

Boxuan Ma[1], Li Chen[2] and Min Lu[3]

[1] Kyushu University, Faculty of Art and Science, Fukuoka, Japan
[2] Kyushu University, Faculty of Information Science and Electrical Engineering, Fukuoka, Japan
[3] Akita University, Faculty of Engineering Science, Akita, Japan

Abstract

As the utilization of digital learning materials continues to rise in higher education, the accumulated operational log data provide a unique opportunity to analyze student reading behaviors. Previous works on reading behaviors for e-books have identified jump-back as frequent student behavior, which refers to students returning to previous pages to reflect on them during the reading. However, the lack of navigation in e-book systems makes finding the right page at once challenging. Students usually need to try several times to find the correct page, which indicates the strong demand for personalized navigation recommendations. This work aims to help the student alleviate this problem by recommending the right page for a jump-back. Specifically, we propose a model for personalized navigation recommendations based on neural networks. A two-phase experiment is conducted to evaluate the proposed model, and the experimental result on real-world datasets validates the feasibility and effectiveness of the proposed method.

Keywords

Reading behavior, E-book navigation, educational data, page recommendation

1. Introduction

E-books are rapidly gaining ground in recent years, transforming how we teach in higher education [29, 30]. Beyond their convenience, e-books provide a treasure trove of data through students' interactions with digital texts, something we cannot do with traditional textbooks. Hundreds of students read the same e-textbook in and out of class, and every interaction, from page flip to highlight and annotation, is recorded. Analyzing these data gives us exciting new possibilities to understand how students behave and improve education.

Meanwhile, despite the vast number of students using e-book systems, the system cannot understand students' intentions to offer students personalized learning experiences, and interactivities between the system and the students still need to be improved [38, 39]. One major challenge of such a system is designing "smart" interactions to improve student engagement and learning experience [23]. For example, jump-back is a frequent behavior with strong user intention when students interact with e-books [22]. Many students often return to previous pages to review during the reading since they want to reread the difficult or missed concept that is not understood well when reading later pages or refer to the related content when doing quizzes or practices [23]. However, the lack of navigation function of an e-book significantly influences its usability, for example, previous works found that students usually need to try several times to find the correct page without good navigation aids with the system [11, 23].

Although much effort has been devoted to e-book navigation research, most of them only focus on designing e-book interfaces, ignoring the power of historical data. With the increasing availability of learner-e-book interaction data, one interesting question arises: Can we leverage data-driven techniques to help alleviate the navigation problem? More specifically, our objective is to explore how we can develop a model to understand user intentions and aid them in locating the most relevant previous pages for rereading.

LAK-WS 2024: Joint Proceedings of LAK 2024 Workshops, March 18–19, Kyoto, Japan
boxuan@artsci.kyushu-u.ac.jp (B. Ma); chenli@artsci.kyushu-u.ac.jp (C. Li); lu@ie.akita-u.ac.jp (M. Lu)
0000-0002-1566-880X (B. Ma); 0000-0003-0063-8744 (C. Li); 0000-0001-7503-1301 (M. Lu)

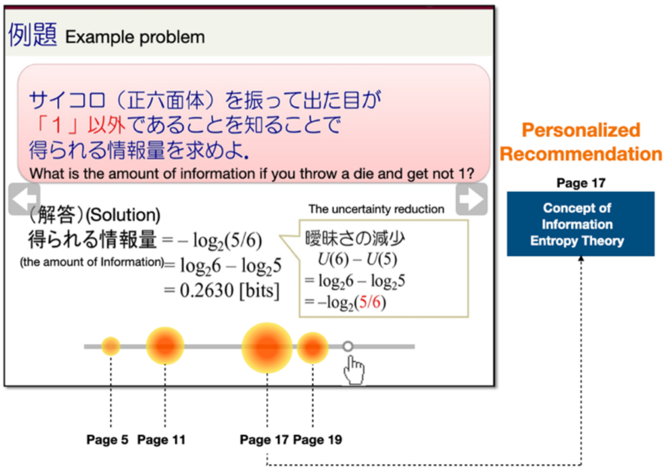

Figure 1: An example of jump-back recommendations. The different-sized circles represent possible end pages, where a larger circle means a more significant probability. The page with the highest probability describes the concept of information entropy theory.

Figure 1 shows a simple example of the navigation problem we are going to deal with. Image a student is reading an e-textbook page, which gives an exercise problem and the solution about calculating the amount of information. To understand the exercise better, he/she attempts to reread the earlier page that explains the relevant concepts. Then, the system automatically detects the student's intention and recommends several relevant pages. For example, page 17, the page with the highest probability, describes the concept of information entropy theory. This problem is referred to as an automated navigation recommendation. The primary challenge here is to develop a data-driven model by considering the lecture content, preferences of the current user, and the historical jump-back behavior of all users for navigation recommendations.

In this paper, we studied the problem of automated navigation recommendations for e-books, which aims to provide more smart interactions between e-book systems and students. Using historical data, we proposed a model to recommend the right page for a jump-back based on user preference. Our experiments validate the effectiveness of the proposed method using real datasets from the courses at our university.

2. Related Work

2.1. Analyzing E-book Reading Behaviors

Many researchers focus on analyzing learners' behaviors when they interact with e-textbook learning materials, aiming to understand how students learn and what they need when reading learning materials [1, 15, 19, 32].

Considerable research efforts have been devoted to mining students browsing patterns based on their log data [26, 31]. For example, Majumdar et al. [24] analyzed e-textbook logs during specific critical reading tasks to represent learners' reading behaviors with learning strategies such as critical reading. Ma et al. [22] modeled and analyzed the differences in e-textbook reading behavior patterns between traditional face-to-face classes and online classes during the pandemic. Their results show that online lectures lead to more off-task behaviors. Some works pay attention to specific reading behaviors. Yin et al. [37] grouped students into four clusters using k-means clustering, and their reading behavioral patterns were analyzed. Ma et al. [21] extracted the jump-back behaviors from the e-textbook reading stream data and then systematically studied the behaviors from different perspectives. Following this line, they identified six content categories that give rise to jump-back incidents: explanation of concept and theorem, example problem and solution, assignments and in-class exercise, learning objectives, beginning of a new unit, and tutorial steps [23].

On the other hand, some researchers focus on the relationship between student learning behavior and their score, and e-book log data are used to predict student performance. Okubo et al. propose a recurrent neural network based method for predicting student performance using e-book log data [27]. Junco et al. [10] conducted linear regression analyses to determine whether e-book usage metrics predicted final course grades. In addition, the correlation between students' reading behaviors in an e-book system and their academic achievement is investigated to identify the key e-book features that may affect students' performance [6] and engagement [35]. Students' e-textbook interaction data can also be used to model student knowledge acquisition. Huang et al. [9] proposed a knowledge tracing model that measures students' level of knowledge of the underlying concept by looking at the amount of time she/he has spent on the related pages of the e-book. Following these works, Akçapınar et al. [2] analyzed students' e-textbook interaction data and developed an early warning system for students at risk of academic failure. Additionally, their other work explored students' reading approaches from e-textbook data using theory-driven and data-driven approaches [1]. The results identified three different reading approaches: deep, strategic, and surface.

Although e-book reading behaviors have been widely explored, most existing works aim at analyzing or visualizing data rather than studying historical data in depth to facilitate e-book interaction. These works inspire us to select useful interaction activities from e-book log data. We use these data further to build interaction techniques for navigation recommendations, which prior work has not done.

2.1. Facilitating Navigation of Learning Material

2.1.1. Facilitating Video Navigation

A large number of works focus on the specific research of video navigation, as watching course videos is the most important activity for online learning platforms such as MOOCs [14]. Some researchers focus on studying user behavior patterns and their implications [13, 17]. They find strong correlations between user behaviors and video content [8], interesting video segments can be detected through users' collective interactions (e.g., seek/scrub, play, pause) with the video [3, 7]. Based on these works, many video interfaces provide navigation distribution according to the historical data, and then detect user intention and recommend potential positions to go automatically. For example, Yadav et al. [34] designed a system that provides nonlinear navigation in educational videos, which utilizes features derived from a combination of the audio and visual content of a video. Carlier et al. [4] collect viewing statistics as users view a video and use these data to reinforce the recommendation of viewports for users. Kim et al. [12] present a 2D video timeline with an embedded visualization of collective navigation traces and a visual summary representing points with frequent learner activity. Zhang et al. [39] introduced an approach to segment videos and used a factorization machine model to provide navigation suggestions in MOOCs.

In general, the literature has pointed to interesting findings and inspired our work. However, these works are limited only to videos, which rely on visual and auditory elements and provide a more immersive passive viewing experience. At the same time, e-books focus on text and images, offer a one-way content consumption by reading, and need more user interaction. Therefore, methods designed for video cannot be applied directly to e-textbook systems in university environments.

2.1.2. Facilitating E-book Navigation.

There are also some works for the specific research of e-book navigation. One line of these works toward designing navigation functions of the interface, including book-marking, reading dashboard, and concept map. For example, Yoon et al. [38] introduced Touch-Bookmark, a multitouch navigation technique for e-books. It enables users to bookmark a page in a casual manner and return to it quickly when required. However, the bookmark still needs to be set manually. Lu et al. [20] developed a reading path dashboard to support the visualization of the reading path for students. Such a dashboard makes it easier for students to find their interests and confusion while a course is offered. However, it is insufficient for personalized navigation aid purposes. In contrast, technology-supported concept maps are also used for e-book navigation [5, 16, 18, 28, 33]. In these approaches, pre-constructed concept maps are presented to learners, and they can use the map as a navigational aid by clicking on concept map nodes to move to the related page. While the problem is the pre-constructed concept maps are expensive as they need human expert labor. Even though some works try to generate concept maps automatically, they are difficult for students to comprehend, minimizing gains from using them as navigational tools [5, 18, 33].

Another line of these works is more similar to ours, which focuses on designing smart interactions by providing automated navigation recommendations for students. Yang et al. [36] propose an e-book page ranking method to rank e-book pages automatically. The top-ranked e-book pages are then selected to form the reading recommendation. However, their method is used for preview purposes and only provides pre-class reading recommendations. Recently, Kang and Yin [11] developed a recommendation system combining the TF-IDF model and page jumping model to recommend students the desired pages when reading, but their model tends to be simple and ignores user preferences.

In summary, previous works made a great success for e-book navigation. However, most works focus on interface design, and there was little work solving this problem using neural networks. Therefore, our work contributes to the current research by providing personalized navigation recommendations leveraging interaction history data and combining content and user preferences to predict and recommend pages for e-textbook navigation automatically when users touch the e-book progress bar.

3. Automated Navigation Framework

3.1. Problem Formulation

Our goal is to make personalized recommendations to a student when he/she is planning to jump back to the previous page for review. More specifically, when a student clicks the cursor on the progress bar of an e-textbook, it triggers navigation recommendations automatically. Formally, given a learning material l, a student s, the start page p_s, the objective is to train a model to maximize the probability that student s would jump back to the end page p_e ($p_e < p_s$) of the e-textbook l. Given enough log data, we aim to build a model to recommend the relevant pages for each learner at each page.

3.2. Model

As shown in Figure 2, we propose a framework with deep learning. Specifically, for each jump-back log, we use the preferences of the corresponding student, the start page, and the end page with learning material characteristics as input. Then, the model learns the interaction function among the variables and outputs the correct probability of the end page.

3.2.1. Features

Inspired by previous studies, we extract the following features from the interaction data in our work, including user preference features, interaction features, and learning material features.

User Preference Features. Previous research showed that different students would have different jump-back patterns, such as the frequency of the jump-back, jump span, and the length of their read time after jumping to their desired page [23]. These features are also helpful in predicting student performance [36]. Therefore, we use the following features to identify specific student's personal preference $\boldsymbol{p}$.

• Stay Time: the reading time after jumping to the student's desired page of the specific jump-back.

• Jump Span: the number of pages between the start and end pages of the specific jump-back.

Interaction Features. These are the basic features that indicate the start and end positions for a specific jump-back behavior:

• Start Page: the corresponding page number of the start page.

• End Page: the corresponding page number of the start page.

Learning Material Features. Previous works showed that features of the learning material also significantly affect students' jump-back behaviors [23, 39]. We first extract some basic features that describe the learning material's attributes and identify specific learning material's characteristics $\boldsymbol{l}$. We also want to include some content-related features. However, they are hard

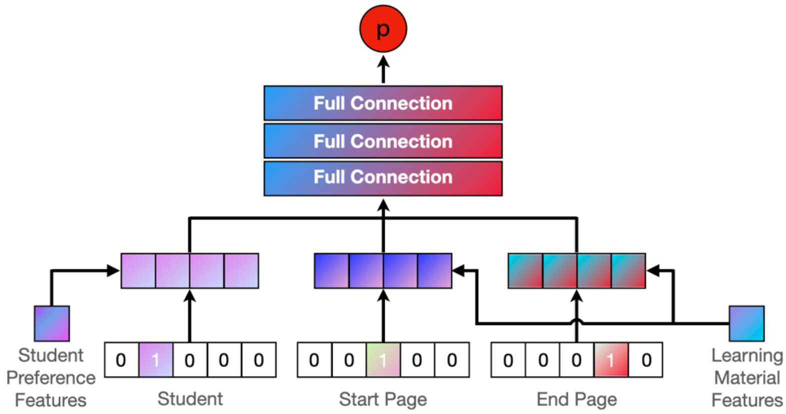

Figure 2: Architecture for the proposed model.

to extract as the learning material includes texts and figures and is mainly in Japanese, so we leave it as our future work.

• Learning Material ID: identify a specific learning material in the dataset. Each learning material has a unique ID.

• Learning Material Length: total number of pages for a specific learning material.

3.2.2. Model Input and Predict Output

The student's characteristic is represented by the one-hot student embedding vector $\boldsymbol{s}$ and student preference features $\boldsymbol{p}$ we extracted before. It can be formulated as:

$$\boldsymbol{sp} = \boldsymbol{s} \oplus \boldsymbol{p}, \tag{1}$$

where $\oplus$ is the concatenation operation. The start page and end page are specific pages in specific learning material. To capture the unique characteristic of each page, the start page is represented by integrating the one-hot page embedding vector $\boldsymbol{p_s}$ and learning material preference $\boldsymbol{l}$. It can be formulated as:

$$\boldsymbol{p_{sl}} = \boldsymbol{p_s} \oplus \boldsymbol{l}. \tag{2}$$

Likewise, the end page is represented by integrating the one-hot page embedding vector $\boldsymbol{p_e}$ and learning material preference $\boldsymbol{l}$:

$$\boldsymbol{p_{el}} = \boldsymbol{p_e} \oplus \boldsymbol{l}. \tag{3}$$

After obtaining representations of the student, the start page, and the end page, we input them into the prediction layer, which includes two full connection layers and an output layer, to output the probability $y\hat{}$ that the student jumps to the end page. It can be formulated as:

$$\hat{y} = \sigma\,(Pred\,(\boldsymbol{s_p}, \boldsymbol{p_{sl}}, \boldsymbol{p_{el}})), \tag{4}$$

where σ is the sigmoid function.

The objective function is a binary cross-entropy loss function. For a specific student s and the start page p_s, the ground truth is a jump-back behavior that truly happens, which means a student has jumped back to the end page p_s from the start page p_e. Let y be the ground truth, and $\hat{y}$ be the predicted probability. Using Adam optimization, all parameters are learned by minimizing the objective function given by:

$$L = \textstyle\sum_t y log\hat{y} + (1 - y) log(1 - \hat{y})\,. \tag{5}$$

4. Evaluation

4.1. Dataset

The datasets used in this study were reading logs collected in two information science courses offered to first-year undergraduate students at our university, one for the 1st-semester and another for the 2nd-semester. The instructor is the same, but the students are different, and the content is slightly changed. There are 16 e-textbook learning materials for the 1st-semester course, and 155 students attended the course, resulting in 580,510 log data. As for the 2nd-semester course, there are 23 e-textbook learning materials, and a total of 230 students attended the course, resulting in 884,327 log data. The length of each learning material is ranging from 7 pages to 61 pages. The instructors and each student used their computers and access to an e-book system to access learning materials. The basic operation of the system is to flip the page. Students can click the previous button to move to the previous page and click the next button to move to the subsequent page, and they could also use a slider to change pages. Students can also use markers and memo annotating for learning. However, we found these operations are rare in the dataset, so we focused on operations related to page flipping.

4.2. Data Preprocessing

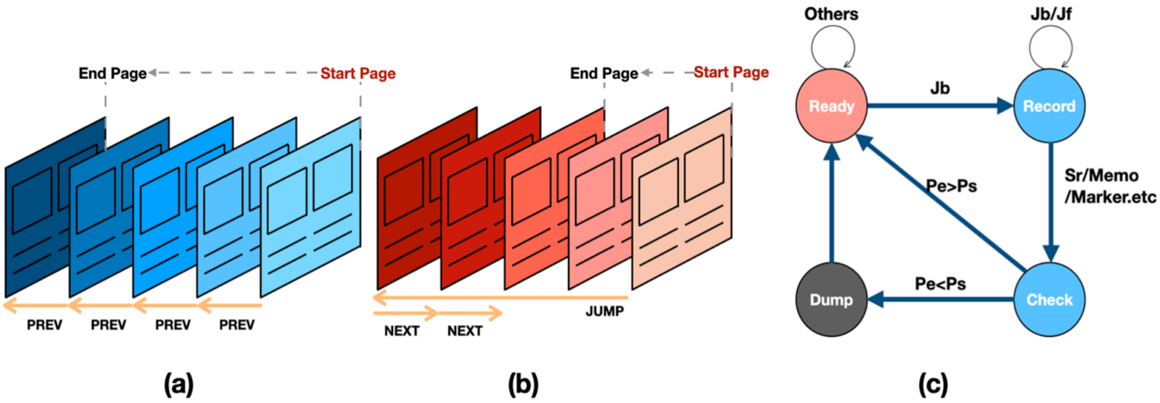

Figure 3: (a) and (b) Two basic jump back patterns. (c) Overview of the DFA.

As we mentioned before, students usually need a series of page-flip actions to find the right page when they jump back. Let (s, l, p_s, p_e) denote a jump-back, which means student s jumps back from the start page p_s to the end page p_e in learning material l ($p_e < p_s$). There are two basic scenarios of students going back to the previous page [21, 23]. The first illustrates a pattern in which the student goes back to a previous page of no interest and continues to look for the correct page that she/he desires to review [25]. As shown in Figure 3(a), the student turns over many pages to jump back to a previous page (p_e). Another pattern is the student jumps back far away from the desired page, and then she/he jumps forward to go to the correct page. As shown in Figure 3(b), the student uses the slider to jump back to an early page first and then clicks the next button two times to jump to the correct page (p_e). To extract the correct start page and end page, and filter the unnecessary page-flip behaviors, we use a deterministic finite automaton to construct the jump behaviors from the data based on previous work [23]. Figure 3(c) shows the overview of the DFA. There are four states: Ready, Record, Check, Dump. At the Ready state, it stays until it receives a jump back event (Jb), then the state goes to Record. When the state is Record, it maintains a stack. When there are jump back events (Jb) or jump forward (Jf) events, it pushes all the events into the stack. After jumping to the desired page in the slide, the student would usually read for seconds. We name it a short-read event. Once there comes a short-read event (Sr) or some other operations (e.g., the student uses maker or memo function) that indicates the student is seriously reading the page, the state transforms to Check state. According to the previous study [23], we tentatively set $2s \leq Sr \leq 20min$. When the state is Check, it compares the start page (ps) of the event at the bottom of the stack and the end page (pe) of the event at the top of the stack. If $pe > ps$, the sequence of events in the stack constitutes a jump forward behavior, then the state goes back to Ready. Otherwise, the state transforms to Dump, where we aggregate the sequence of events in the stack to construct a complete jump behavior that filters the unnecessary page-flip behaviors and extracts the correct start page and end page.

4.3. Experiment Settings

4.3.1. Evaluation Metrics

Like previous works [11, 36, 39], the experiment is conducted in the prediction and recommendation stages. First, the probability of each end page from the same start page will be predicted in the prediction stage. Then, the n end pages with the highest probability will be used for navigation recommendations. For the prediction experiment, we evaluate the performance in terms of Area Under Curve (AUC), Recall, Precision, and F1-score. For the recommendation experiment, we use hits@n to measure the recommendation performance. The full dataset was used in the experiment, using the first 80% for training and 20% for test purposes, respectively.

4.3.2. Generating Negative Samples

In our experiment, a positive sample corresponds to a jump-back that truly occurs, denoted as (s, l, p_s, p_e), which means student s jumps back from the start page p_s to the end page p_e in learning material l ($p_e < p_s$). Following previous work [39], we generated several negative samples by choosing different end pages for each positive sample. Given a start page p_s, there is a list of end pages that all students have jumped back to. While for a specific user's specific jump-back, there is only one end page that the student truly jumps back to. In the remaining list of end pages, we randomly select n end pages to generate negative samples. We set n as 2 in the following experiments to avoid imbalanced data.

4.3.3. Parameter Settings

The model was implemented in PyTorch and was trained with a batch size of 256. We used Adam optimizer with a learning rate of 0.001. The dropout rate is set to 0.2, and early stopping is applied to reduce overfitting.

5. RESULTS

5.1. Prediction Performance

Table 1
Prediction Performance.

Dataset	Method	ACC	AUC	Precision	Recall	F1-score
1st Semester	SVM	0.612	0.512	0.467	0.701	0.561
	LR	0.759	0.788	0.699	0.430	0.532
	NPR-U	0.799	0.848	0.740	0.558	0.637
	NPR-LM	0.846	0.929	0.778	0.712	0.744
	NPR	**0.850**	**0.933**	**0.785**	**0.718**	**0.750**
2nd Semester	SVM	0.616	0.522	0.447	0.778	0.568
	LR	0.756	0.785	0.704	0.442	0.543
	NPR-U	0.792	0.841	0.732	0.570	0.641
	NPR-LM	0.840	0.928	0.764	0.734	0.749
	NPR	**0.852**	**0.934**	**0.768**	**0.780**	**0.774**

We named our model Neural Page Recommendation (NPR) and compared our model to models that were proposed in previous work [36, 39], including models based on Logistic Regression (LR) and Support Vector Machine (SVM). To gain a deeper understanding of our model, we also add variations of our model. NPR-U is the variation that takes out user related features from the model, and NPR-LM is the variation that takes out learning material related features from the model.

Table 1 shows the performance of all baseline methods and our model. Overall, our model performs better than other machine learning methods, demonstrating that leveraging deep learning could model student interactions more accurately than other models. Moreover, we observed a significant performance drop for our model when taking out user-related features. This result aligns with previous work that students have their personal preferences when they jump back [23] and indicates that incorporating user preference could effectively boost the model. Also, the performance of the NPR-LM model, which takes out learning material related features, only drops slightly. The reason may be that each learning material's content structure design is similar, and the relevant page is not so distant from the current one. Previous works suggested that the content information on each page may help find the right pages [11, 39], and we will consider including such features in our model in the future.

5.2. Ranking Performance

In the predicting experiment, given a specific start page, we get the probability of each end page for a given jump-back. Then, we rank these end pages by their probabilities produced in the prediction stage, and a ranked list of n end pages with the highest probability will be used for navigation recommendations. We compare our method with Random and Frequency-based recommendations [39] for the ranking experiment. The Random approach randomly selects previous pages of the given start page. The Frequency-based approach recommends the most frequent end-pages of all students for a given start page. Table 2 shows the result of the ranking experiment. It indicates that our model based on deep learning outperforms the other methods. The performance of Random recommendation is very low since there are many pages, and it is difficult to recommend the right end page. The Frequency-based approach performs better since the end pages that students reread frequently are usually related and considered important. However, it does not consider student preference, making it hard to provide personalized recommendations.

Table 2
Ranking Performance.

Dataset	Method	hits@1	hits@2	hits@3	hits@4	hits@5
1st Semester	Random	0.090	0.168	0.241	0.306	0.362
	Frequency	0.536	0.680	0.750	0.803	0.840
	NPR	**0.624**	**0.718**	**0.773**	**0.810**	**0.844**
2nd Semester	Random	0.082	0.169	0.238	0.299	0.359
	Frequency	0.511	0.650	0.726	0.781	0.820
	NPR	**0.598**	**0.705**	**0.761**	**0.800**	**0.828**

6. Conclusion

In this paper, we studied the problem of automated navigation recommendations for e-books, which aims at more smart interactions between e-book systems and students. We proposed a model to recommend the right pages for a jump-back. Our experiments validate the effectiveness of the proposed method using real datasets. Also, our results highlight the need for recommendation models that consider student's personal preferences. For future work, we consider including actual content information of each page in our model, which may include the rich information hidden within the texts and underlying topics. Also, we want to consider the dynamic behaviors of students before jump-backs to understand student intentions better and provide recommendations more intelligently.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number: JP20H00622, JP23K1136, JP22H00552, JP21K18134.

References

[1] G.khan Ak.apinar, Mei-Rong Alice Chen, Rwitajit Majumdar, Brendan Flanagan, and Hiroaki Ogata. 2020. Exploring student approaches to learning through sequence analysis of reading logs. In Proceedings of the tenth international conference on learning analytics & knowledge. 106–111.
[2] G.khan Ak.apınar, Mohammad Nehal Hasnine, Rwitajit Majumdar, Brendan Flanagan, and Hiroaki Ogata. 2019. Developing an early-warning system for spotting at-risk students by using eBook interaction logs. Smart Learning Environments 6, 1 (2019), 4.
[3] Markos Avlonitis and Konstantinos Chorianopoulos. 2014. Video pulses: User-based modeling of interesting video segments. Advances in Multimedia 2014 (2014).
[4] Axel Carlier, Guntur Ravindra, Vincent Charvillat, and Wei Tsang Ooi. 2011. Combining content-based analysis and crowdsourcing to improve user interaction with zoomable video. In Proceedings of the 19th ACM international conference on Multimedia. 43–52.
[5] Mary J Carnot, Alberto J Ca.as, and Jason Muldoon. Concept maps vs. web pages for information searching and browsing.

[6] Cheng-Huan Chen and Chien-Yuan Su. 2019. Using the BookRoll e-book system to promote self-regulated learning, self-efficacy and academic achievement for university students. Journal of Educational Technology & Society 22, 4 (2019), 33–46.
[7] Konstantinos Chorianopoulos. 2013. Collective intelligence within web video. Human-centric Computing and Information Sciences 3, 1 (2013), 1–16.
[8] Konstantinos Chorianopoulos, Ioannis Leftheriotis, and Chryssoula Gkonela. 2011. SocialSkip: pragmatic understanding within web video. In Proceedings of the 9th European Conference on Interactive TV and Video. 25–28.
[9] Yun Huang, Michael Yudelson, Shuguang Han, Daqing He, and Peter Brusilovsky. 2016. A framework for dynamic knowledge modeling in textbook-based learning. In Proceedings of the 2016 conference on user modeling adaptation and personalization. 141–150.
[10] Reynol Junco and Candrianna Clem. 2015. Predicting course outcomes with digital textbook usage data. The Internet and Higher Education 27 (2015), 54–63.
[11] Zichen Kang and Chengjiu Yin. 2023. Personalized page-transition recommendation model based on contents and students' logs in digital textbooks. In 2023 International Conference on Artificial Intelligence and Education (ICAIE). IEEE, 62–66.
[12] Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Li, Krzysztof Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In Proceedings of the 27th annual ACM symposium on User interface software and technology. 563–572.
[13] Juho Kim, Philip J Guo, Daniel T Seaton, Piotr Mitros, Krzysztof Z Gajos, and Robert C Miller. 2014. Understanding in-video dropouts and interaction peaks in online lecture videos. In Proceedings of the first ACM conference on Learning@ scale conference. 31–40.
[14] Dimitrios Kravvaris and Katia Lida Kermanidis. 2019. Automatic point of interest detection for open online educational video lectures. Multimedia Tools and Applications 78 (2019), 2465–2479.
[15] Effie Lai-Chong Law and Marta Krist.n L.rusd.ttir. 2015. Whose experience do we care about? Analysis of the fitness of scrum and kanban to user experience. International Journal of Human-Computer Interaction 31, 9 (2015), 584–602.
[16] Liang-Yi Li, Gwo-Dong Chen, and Sheng-Jie Yang. 2013. Construction of cognitive maps to improve e-book reading and navigation. Computers & Education 60, 1 (2013), 32–39.
[17] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. 2015. MOOC video interaction patterns: What do they tell us? In Design for teaching and learning in a networked world. Springer, Spain, 197–210.
[18] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScape: Collaborative concept mapping for video learning. In Proceedings of the 2018 CHI conference on human factors in computing systems. 1–12.
[19] Stephan Lorenzen, Niklas Hjuler, and Stephen Alstrup. 2018. Tracking Behavioral Patterns among Students in an Online Educational System. International Educational Data Mining Society (2018).
[20] Min Lu, Li Chen, Yoshiko Goda, Atsushi Shimada, and Masanori Yamada. 2020. Development of a learning dashboard prototype supporting meta-cognition for students. In Companion proceedings of the 10th international conference on learning analytics & knowledge (LAK20). 104–106.
[21] Boxuan Ma, Jiadong Chen, Chenhao Li, Likun Liu, Min Lu, Yuta Taniguchi, and Shin'ichi Konomi. 2020. Understanding jump back behaviors in e-book system. In Comanion Proceedings of the 10th International Conference on Learning Analytics & Knowledge. 623–631.
[22] Boxuan Ma, Min Lu, and Shin'ichi Konomi. 2021. Understanding student slide reading patterns during the pandemic. In 18th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA 2021. IADIS Press, 87–94.
[23] Boxuan Ma, Min Lu, Yuta Taniguchi, and Shin'ichi Konomi. 2022. Exploring jump back behavior patterns and reasons in e-book system. Smart Learning Environments 9, 1 (2022).
[24] Rwitajit Majumdar, Geetha Bakilapadavu, Reek Majumder, Mei-Rong Alice Chen, Brendan Flanagan, and Hiroaki Ogata. 2021. Learning analytics of humanities course: reader profiles in critical reading activity. Research and Practice in Technology Enhanced Learning 16, 1 (2021), 25.
[25] Catherine C Marshall and Sara Bly. 2005. Turning the page on navigation. In Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries. 225–234.
[26] Kousuke Mouri, Fumiya Suzuki, Atsushi Shimada, Noriko Uosaki, Chengjiu Yin, Keiichi Kaneko, and Hiroaki Ogata. 2021. Educational data mining for discovering hidden browsing patterns using non-negative matrix factorization. Interactive Learning Environments 29, 7 (2021), 1176–1188.

[27] Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, and Hiroaki Ogata. 2017. A neural network approach for students' performance prediction. In Proceedings of the seventh international learning analytics & knowledge conference. 598–599.
[28] Sadhana Puntambekar and Agni Stylianou. 2005. Designing navigation support in hypertext systems based on navigation patterns. Instructional science 33 (2005), 451–481.
[29] Lee Rainie, Kathryn Zickuhr, Kristen Purcell, Mary Madden, and Joanna Brenner. 2012. The Rise of E-Reading. Pew Internet & American Life Project (2012).
[30] James A Shepperd, Jodi L Grace, and Erika J Koch. 2008. Evaluating the electronic textbook: Is it time to dispense with the paper text? Teaching of psychology 35, 1 (2008), 2–5.
[31] Atsushi Shimada, Fumiya Okubo, and Hiroaki Ogata. 2016. Browsing-Pattern Mining from e-Book Logs with Non-negative Matrix Factorization.. In EDM. 636–637.
[32] Alistair Sutcliffe and Jennefer Hart. 2017. Analyzing the role of interactivity in user experience. International Journal of Human–Computer Interaction 33, 3 (2017), 229–240.
[33] Shang Wang and Erin Walker. 2021. Providing adaptive feedback in concept mapping to improve reading comprehension. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 1–11.
[34] Kuldeep Yadav, Kundan Shrivastava, Mohana Prasad, Harish Arsikere, Sonal Patil, Ranjeet Kumar, and Om Deshmukh. 2015. Content-driven multi-modal techniques for non-linear video navigation. In Proceedings of the 20th International Conference on Intelligent User Interfaces.
[35] Albert Yang, YL Chen, Brendan Flanagan, and Hiroaki Ogata. 2020. Applying Key Concepts Extraction for Evaluating the Quality of Students' Highlights on e-Book. In 28th International Conference on Computers in Education Conference Proceedings, Vol. 1. Asia-Pacific Society for Computers in Education (APSCE), 284–288.
[36] CC Yang, G.khan Ak.apinar, Brendan Flanagan, and Hiroaki Ogata. 2019. Developing e-book page ranking model for pre-class reading recommendation. In Proceeding of 27th International Conference on Computer in Education (ICCE 2019). 360–362.
[37] Chengjiu Yin, Fumiya Okubo, Atsushi Shimada, Misato Oi, Sachio Hirokawa, and Hiroaki Ogata. 2015. Identifying and analyzing the learning behaviors of students using e-books. In Proceedings of the 23rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education Hangzhou, China, 118–120.
[38] Dongwook Yoon, Yongjun Cho, Kiwon Yeom, and Ji-Hyung Park. 2011. Touch-Bookmark: A lightweight navigation and bookmarking technique for e-books. In CHI'11 Extended Abstracts on Human Factors in Computing Systems. 1189–1194.
[39] Han Zhang, Maosong Sun, Xiaochen Wang, Zhengyang Song, Jie Tang, and Jimeng Sun. 2017. Smart jump: Automated navigation suggestion for videos in moocs. In Proceedings of the 26th international conference on world wide web companion. 331–339.