A page jump recommendation model and result interpretation based on structured annotation methods

Wenhao Wang
Kobe University
w496886951@hotmail.com

Etsuko Kumamoto
Kobe University
kumamoto@kobe-u.ac.jp

Chengjiu Yin
Kyushu University
yin.chengjiu.247@m.kyushu-u.ac.jp

ABSTRACT
The e-book system, widely used in learning and teaching, has generated a large amount of log data over time. Researchers analyzing these data have discovered the existence of student's jump back behavior, which is positively correlated with academic achievement. However, they also found that this behavior has the disadvantage of low efficiency. To address this issue, book page recommendation systems have been developed. Unfortunately, these systems currently suffer from low accuracy rates and their recommendation results lack interpret-ability. In response to these challenges, we developed an e-book page recommendation model based on natural language processing and student reading data in this study. We then employed structured annotation methods to analyze the recommendation results.

Keywords
Education data, page recommendation, BERTopic model, Natural language process.

1. INTRODUCTION
Online learning, integrated with the internet, has made reading e-learning materials essential [19][15].

Researchers found that students often revisit pages, correlating with academic achievement, but finding desired pages in e-books can be inefficient [28][9][16]. A TF-IDF-based recommendation model was proposed to improve efficiency, but it lacked accuracy and clear explanations [10][23].

This study developed a BERTopic-based e-book page recommendation model, using structured annotations for better interpret-ability [8].

In summary, this study focuses on the following research questions.

RQ 1: Can the introduction of the Bertopic model improve the accuracy of page recommendation?

RQ 2: What factors affecting the recommendation accuracy can be identified from the results of this recommendation?.

2. LITERATURE REVIEW
2.1 Natural Language Processing in education
Natural Language Processing (NLP) has been increasingly applied in education. [1] used it to analyze interview data. [20] discussed trends and challenges in NLP for education feedback analysis. [17] proposed an NLP-based model for ranking higher education institutes. [2] developed an NLP algorithm for feedback in medical education. [24] emphasized introducing NLP concepts to K-12 students. [12] conducted a survey of NLP applications in education. [4] provided tips for implementing NLP in medical education program evaluation. [29] leveraged Large Language Models for concept graph recovery and question answering in NLP education.

In this study, we also process the content of the textbook using natural language processing techniques to generate page recommendations based on semantic connections.

2.2 Recommendation system in e-learning
Recommendation systems in e-learning have evolved significantly. [25] tackled the cold-start problem in paper recommendation systems. [13] personalized courseware recommendations using knowledge discovery techniques. [6] introduced a hybrid filtering method for efficient resource recommendation. [14] combined collaborative filtering and sequential pattern mining for learning item recommendations. [27] proposed a learner-oriented approach based on mixed concept mapping and immune algorithm. [26] explored 'user unknowns' through an interactive process. [21] used personality information to enhance community recommendations. Lastly, [22] presented an agent-based recommendation approach using knowledge discovery and machine learning. While these systems provide students with learning strategies, paths, or resources, there is a noticeable gap in e-book page recommendations.

To fill this void, this study developed a backtracking-based recommendation system for e-book pages, aiming to enhance students' reading efficiency and satisfaction. We also employed structured annotation techniques to increase the interpretability of the outcomes generated by this intelligent recommendation system.

3. METHOD
In this section, we delve into the data sources utilized for building the recommendation model, the method employed for mining the similarity between book pages, and the definition and discovery method of students' complete jump behavior. We will conclude with an introduction to the generation process of the recommendation model and its results

3.1 Datasets
The datasets utilized in this study comprises studying logs collected from a commercial law course in 2017 and 2018. This course aimed to impart legal spirits, legal thoughts, and the application of legal knowledge to 297 university students.

W. Wang, E. Kumamoto, and C. Yin. A page jump recommendation model and result interpretation based on structured annotation methods. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 368–375, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729838

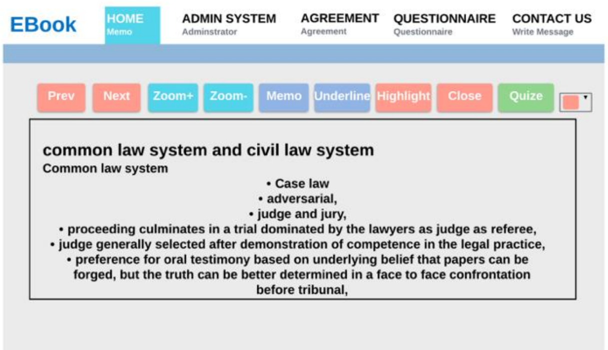

Figure 1. The example of the textbook.

A total of 146076 rows of log data were collected, encompassing the student's account number, operation type, occurrence time, and the page number where the operation took place. As illustrated in the table, "NEXT" signifies that the student clicked the "Next" button to proceed to the subsequent page, while "PREV" indicates a click on the "Prev" button to return to the previous page. For instance, the first row of data reveals that student '915' navigated from page 12 to page 13 at 11:42:27.6 a.m. on March 9, 2018.

Table 1. Examples of student log data.

User no	process code	operation name	operation date	eBook no	page no
915	2	NEXT id:39	2018-3-9 11:42:27.6	39	12
956	1	PREV id:39	2018-3-9 11:42:28.2	39	2

3.2 BERTopic model for page recommendation

This study employs the BERTopic model, a cutting-edge topic model in the field of natural language processing, to uncover the hidden semantic associations between pages. BERTopic is a technique that leverages BERT embeddings and utilizes word embedding methods such as word2vec, dimensionality reduction methods like UMAP, HDBSCAN clustering, and class-based TF-IDF (c-tf-idf) to create dense clusters. This process generates topics and retains the important words of topics [18]. The BERTopic model stands out from other topic models as it bridges the gap between density-based clustering and centroid-based sampling [7]

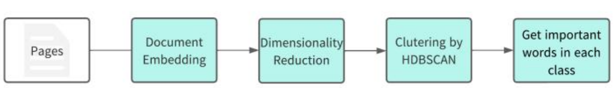

Figure 2. The process of BERTopic model.

The use of HDBSCAN clustering eliminates the need for manual determination of the optimal number of clusters, significantly enhancing the efficiency of clustering. Additionally, c-tf-idf takes into account not only the importance of words in the document but also their significance in the topic [31]. Therefore, this method is adopted in this study for mining textbook content.

$$W_{t,c} = tf_{t,c} \times \log(1 + \frac{A}{f_t})$$

In this study, we adopted the latest topic modeling techniques to uncover the hidden semantic associations between textbook pages. As depicted in the formula, $W_{t,c}$ signifies the importance of term t in class c, $tf_{t,c}$ denotes the frequency of term t in class c, "A" represents the average number of words in each category, and f_t indicates the frequency of term t appearing in all categories. A term t can better represent class c when f_t is smaller and $tf_{t,c}$ is larger, implying that term t frequently appears in class c but seldom appears in other classes.

To discover the association between textbook pages, we conducted topic modeling on 100 pages of the textbook. The process began with preprocessing each page of text, which included word segmentation removal, articles, and other stop words. Subsequently, following the BERTopic process, we employed word2vec, HDBSCAN, and c-tf-idf algorithms for word embedding, text clustering, and extracting topic keywords respectively. The results revealed three topics mined from the 100 pages of the textbook. The composition of each page's topic is illustrated in Figure 4.

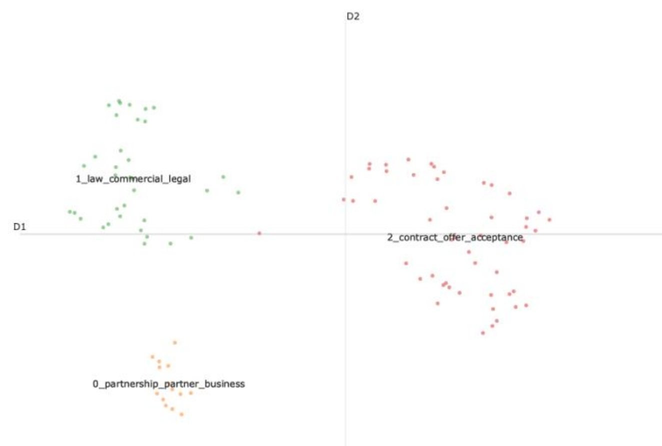

Figure 3. Result of the topic mining. Different colors correspond to different topics, each dot corresponds to a page and the three most important words of each class are marked in the figure.

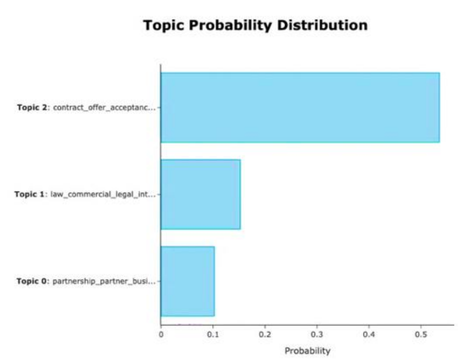

Figure 4. Example of the topic composition of a page.

Through the aforementioned process, each page of the textbook is transformed into a row vector. By calculating the cosine similarity, we can ultimately obtain the correlation matrix R_1 between each page of the textbook.

3.3 Page jump behavior modeling

To ascertain the relevance of pages from students' reading logs, we need a model to identify students' page jump behavior. Given the effectiveness of the Jump back behavior modeling proposed in [15] in identifying students' complete-jump behavior as proven in [32], this study seeks the association between book pages based on this method.

Definition 1: complete jump. A complete jump comprises one or more page transitions by a student to locate the desired page for review. Here, (*u, B, s, e*) denotes that student *u* jumps from start page *s* to end page *e* in book *B*.

Definition 2: jumping back (*Jb*). A jumping back event occurs when a student uses the PREV button to jump from the current page *c* to a previous page *e* (*c*>*e*).

Definition 3: jumping forward (*Jf*). A jumping forward event occurs when a student uses the NEXT button to jump from the current page *c* to a subsequent page *e* (*c*<*e*).

Definition 4: short reading (*Sr*). After jumping to a page in the textbook, the student typically reads it for a period to determine whether the page is the desired one. We term this a short reading event. It helps determine the end of a complete jump. For instance, if the time of the first jumping event is at t_1 and the next one is at t_2, then the short reading period is $Sr = t_2 - t_1$.

Based on the definitions provided, a Deterministic Finite Automaton (DFA) is used to construct the complete jump behaviors. Figure 4 illustrates the transitions in the DFA. In essence, it includes four states: Ready, Record, Check, and Dump.

Ready state: The state transitions to Record upon receiving a *Jb* or *Jf* event.

Record state: The DFA maintains a stuck at this state. When a *Jb* or *Jf* event is received, it pushes all these events into the stack. If a Sr event occurs, the state transitions to Check.

Check state: In the Check state, it compares the duration time Sr to the valid reading time. If $t_{min} \leq \text{Sr} \leq t_{max}$, it transitions to the Dump state. Otherwise, it reverts to the Ready state. While previous studies suggested that t_{min}=2s and t_{max}=20min [30], this study calculated t_{min} using the total number of words per page and the English reading speed of college Japanese students (words per minute(wpm)=144.42) [5], given that the number of words per page of the textbook varies.

Dump state: The sequence of events in the stack is aggregated to construct a complete-jump behavior.

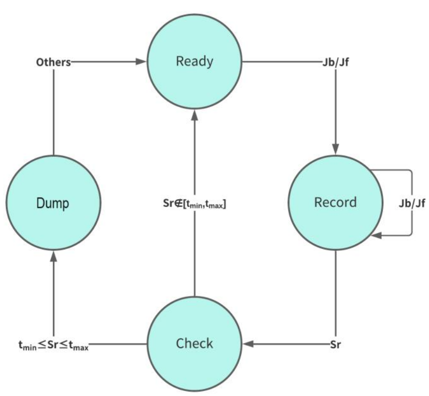

Figure 5. DFA used to judge complete jump behavior.

After extracting complete jump behaviors from logs, the relevance between pages can be obtained and described by probability relevance score matrix R_2:

$$R_2 = \tanh\left(\frac{P_{s,e}}{1 - P_{s,e}}\right)$$

here the $P_{s,e}$ can be defined as:

$$P_{s,e} = \frac{n_{s,e}}{\sum_{e=1}^{N} n_{s,e}}$$

and the $n_{s,e}$ denotes the number of complete jump behaviors (*u, B, s, e*) of all students *u* on textbook *B*.

3.4 Construction of the hybrid model

The recommendation model introduced in section 3.2, which is grounded in textbook content, solely considers the semantic similarity between pages. Conversely, the probability relevance score matrix obtained from log data, as introduced in section 3.3, disregards the relevance of textbook content entirely. Consequently, relying on a single model for page recommendation may not yield satisfactory results. However, in most instances, the amalgamation of multiple recommendation models can enhance performance [11].

Therefore, this paper uses a compiling method based on Guttman's Point Alienation statistic [3] to combine the recommendation models in sections 3.2 and 3.3:

$$R(\alpha, \beta) = \alpha R_1 + \beta R_2$$

Here R is the overall relevance matrix and α and β are the weights of R_1 and R_2 respectively.

The way in this study to find out the optimal values of the parameters is given by this formula [3]:

$$J_s(\alpha, \beta) = -\frac{\sum_{e1 >_{q_s} e2} [R_{s,e1}(\alpha, \beta) - R_{s,e2}(\alpha, \beta)]}{\sum_{e1 >_{q_s} e2} |R_{s,e1}(\alpha, \beta) - R_{s,e2}(\alpha, \beta)|}$$

$$\arg\min J = \frac{1}{card(Q)} \sum J_s(\alpha, \beta)$$

The notation $e1 >_{q_s} e2$ signifies that for a given page s, the user is more inclined to jump to page e1 than to page e2. Here, $R_{s,e1}$ represents the similarity value between page s and page e1, while $R_{s,e2}$ denotes the similarity value between page s and page e2. The set Q={q_s} corresponds to the pre-prepared recommendation list for the textbook, extracted from students' real jumping records, where each q_s is the recommendation list for page s. When the order of the recommendation list generated by the model perfectly aligns with the order of the expert recommendation list, J is set to -1.

By comparing the order of the recommendation list produced by the model with that of the generated jump record list, we can determine the optimal parameter values which minimize J.

4. RESULT

In this section, we initially evaluate the performance of the recommendation model on the validation set, analyze the recommendation results, and subsequently employ structured annotation by examining the textbook content and student types.

4.1 Recommendation model's performance evaluation

The training sets used here comes from 177 students who participated in the law course in 2017. The testing sets used to evaluate the recommendation model's performance comes from 120 students who participated in the course in 2018.

The performance of the recommendation system is evaluated using the F1 score, coverage, and NDCG. As shown in the table, there is a significant difference in the performance of the recommendation results on the test set when only recommending 1 page and recommending 3 pages, with the performance being superior when only recommending 1 page. At the same time, compared with the system in [10], although it only uses recall for evaluation, the optimal recall value on the test set in this paper is 0.64, but the system in this paper has reached a maximum recall performance of 0.81.

Table 2. The performance of the recommendation system

Number of recommended pages	F1 SCORE	Coverage	NDGC@3
1 page	0.9398	0.97	0.955
3 pages	0.6153	0.51	0.5273

Result 1: The model's recommendation results experience a decline in quality as the number of recommended pages increases.

Figure 6 presents the recall rate of the recommendation results. As depicted in the figure, when the recommendation system suggests only a single, most likely page to jump to, the recall rate can attain a relatively high level of 0.81. However, with an increase in the number of recommended pages, the recall rate of the recommendation model on the test set rapidly decreases to 0.3.

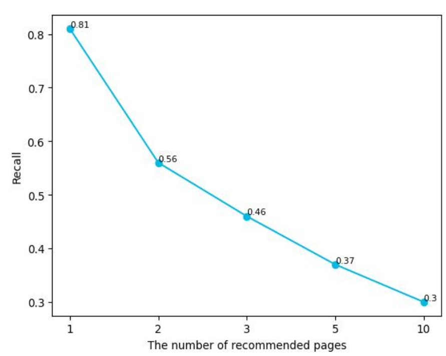

Figure 6. The change in recall rate as the number of recommended pages increases.

Result 2: The recommendation model's performance diminishes for the end sections after segmenting the textbook.

Figure 7 illustrates the performance of the recommendation model on the data set when the textbook is divided into three parts: beginning (p.1-p.40), middle (p.40-p.60), and end (p.60-p.100). From Figures 13, 14, and 15, the students' page-turning behavior shows significant changes around page 40 and page 60. Therefore, we simply divide this textbook at page 40 and page 60 to try to find possible reasons. Similar to the results in the previous figure, as the number of recommended pages increases, the recall rate of the model on the test set rapidly declines to around 0.3. Additionally, it can be observed from the figure that the results for the beginning and middle sections of the textbook are quite similar, both significantly higher than that of the end section.

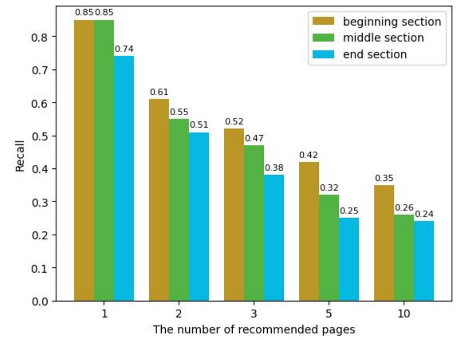

Figure 7. The performance of each section on the test set.

4.2 Textbook page and students' structured annotations

To identify the cause of the results in section 4.1, we categorized each page based on the textbook content. Furthermore, we clustered the students in the test set based on the number of jumps and reading time.

Table 3. Page types and their definitions.

Page type	Definition
Type1: concept explanation	Textual explanation of laws, legal terms, etc.
Type2: Concept comparison and examples	Comparison and specific cases of laws, legal terms, etc.
Type3: Questions with analysis	Questions with analysis or answers
Type4: Questions without analysis	Questions without answers
Type5: The structure of the knowledge points	The structure of the knowledge points in the section.
Type6: Title page	A page that only contains the title content

Figure 8 presents the clustering results of the students. It reveals three distinct student types: Type A students have less total reading time and fewer jumping records, Type C students have a longer reading time and more jumping records, while Type B students fall between Type A and Type C in terms of total reading time and number of jumping records.

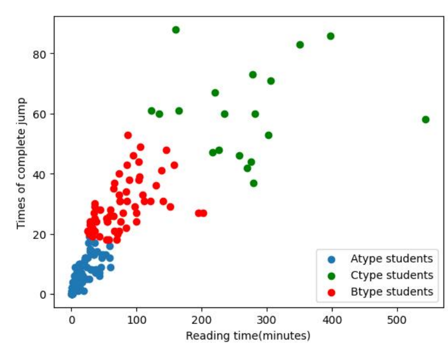

Figure 8. The result of student clustering.

Figure 9, figure 10 and figure11 present the analysis of the similarity between the jumping target page and the current page for each type of student. We consider there to be a strong semantic connection between two pages when the similarity exceeds 0.95. Conversely, we believe there is no conceptual connection between the two pages when the similarity is less than 0.5. The figure reveals that Type C students initially tend to jump to pages with conceptual associations, but this tendency significantly diminishes as the number of selected pages increases. On the other hand, Type A students exhibit a different jumping tendency from Type C students. As the number of selected pages increases, they increasingly prefer to jump to pages without obvious semantic associations.

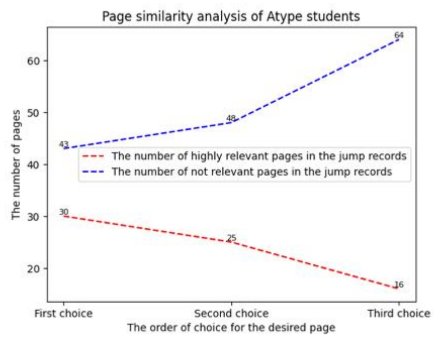

Figure 9. The change in the number of high relevant pages and not relevant pages for A type of student.

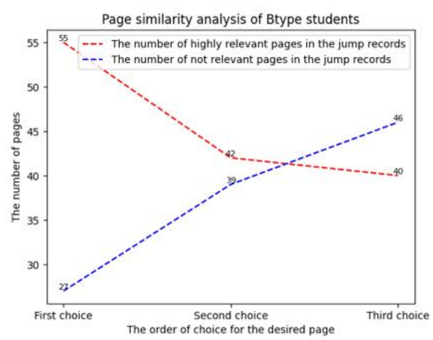

Figure 10. The change in the number of high relevant pages and not relevant pages for B type of student.

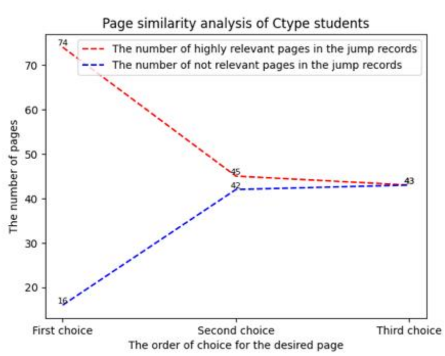

Figure 11. The change in the number of high relevant pages and not relevant pages for C type of student.

Table 4 presents the type analysis of the target pages that each student type jumps to. The concept explanation page is the most frequent type of page that Type C and B students jump to. As the number of selected pages increases, the proportion of Type2 pages in the jumping records of Type C students also increases. Conversely, for Type A students, the most frequent type of page they jump to is the Type2 page. However, as the number of selected pages increases, the proportion of Type1 pages rises while that of Type2 pages decreases.

Table 4. The change in the number of target page types in the jump records of various types of students.

	Ctype students		
Page types	First choice	Second choice	Third choice
Type1	47	44	42
Type2	20	34	28
Type3/Type4	15	10	12
Type5/Type6	17	11	17
	Btype students		
Page types	First choice	Second choice	Third choice
Type1	44	40	49
Type2	22	23	20
Type3/Type4	11	10	7
Type5/Type6	22	26	23
	Atype students		
Page types	First choice	Second choice	Third choice
Type1	29	34	54
Type2	40	25	15
Type3/Type4	8	0	4
Type5/Type6	22	40	26

Table 5 presents the type analysis of each section of the textbook after division. The figure reveals that the composition of page types in the beginning and middle sections are strikingly similar. However, in the composition of the end section pages, the proportion of Type2 pages has significantly increased.

Table 5. The number of each type of page in each section.

Page type	**Beginning section**	**Middle section**	**End section**
Type1	23	12	16
Type2	5	2	17
Type3	5	5	4
Type4	0	0	1
Type5	4	1	1
Type6	3	0	1

5. DISCUSSION

In this section we will try to explain the results shown in section 4.1 by combining the content of section 4.2.

Explanation of Result 1: The recommendation model struggles to meet diverse jump needs as the number of recommended pages increases, due to varying jump patterns among different types of students.

From Figure 9, Figure 10, Figure 11 and Table 4, it's evident that different types of students exhibit distinct preferences when choosing the target page to jump to. Type A students tend to seek textual explanations of concepts, while Type C students are more inclined towards comparisons of concepts or actual cases. Figure

12 further illustrates that there is minimal overlap in the target pages among the three types of students.

Moreover, Figure 13, Figure 14, Figure 15 are about the jump records for each page, specifically for various types of students. In these records, we’re focusing on the top five pages that have the most jumps. The distribution we’re discussing refers to the number of jumps to these top five pages.And the results show that there is almost no difference in the number of jumps to the target pages after the second-choice page. This suggests that even within the same type of students, the motivations behind their jumping behavior differ.

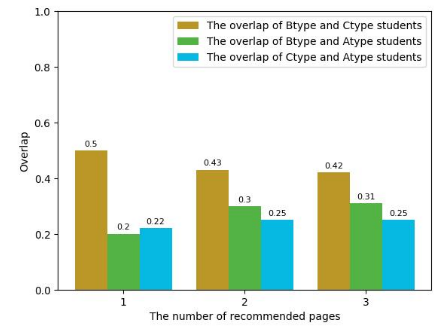

Figure 12. The overlap of the target pages jumped by three types of students.

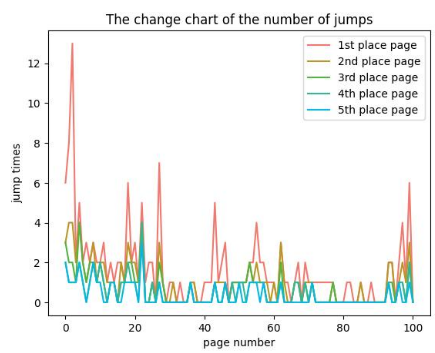

Figure 13. The distribution of the number of jumps to the top five pages with the most jumps in the jump records of each page of C type students.

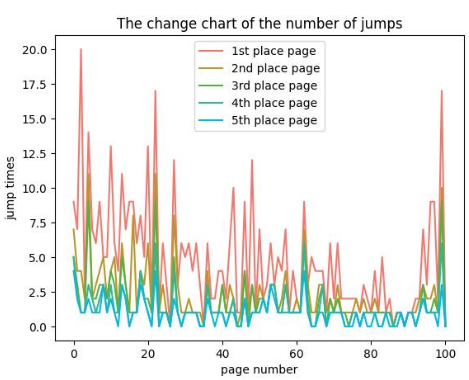

Figure 14. The distribution of the number of jumps to the top five pages with the most jumps in the jump records of each page of B type students.

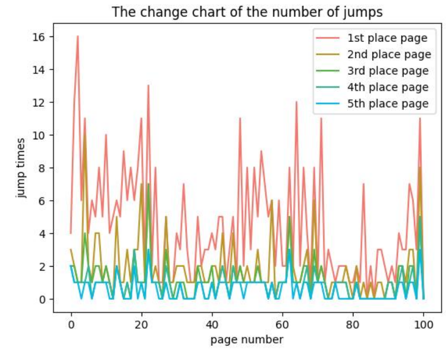

Figure 15. The distribution of the number of jumps to the top five pages with the most jumps in the jump records of each page of A type students.

Explanation of Result 2: The immediate cause is the decrease in the number of jump records in the end section, while the underlying reason lies in the distinct content structure of the end section compared to other sections.

As depicted in Figure 16, in the end section, both the average number of words per page and the average reading time of students have increased compared to other sections. Conversely, the number of recorded complete jumps has decreased. This phenomenon can be attributed to the significant increase in the proportion of Type 2 pages in the end section, as shown in Table 5. [15] highlighted that students primarily jump back to find explanations of concepts. Since Type 2 pages extensively explain concepts, it diminishes students’ need to look forward for concept explanations. Consequently, there are fewer jump records in the end section, and the model cannot obtain optimal parameter values for this part during training.

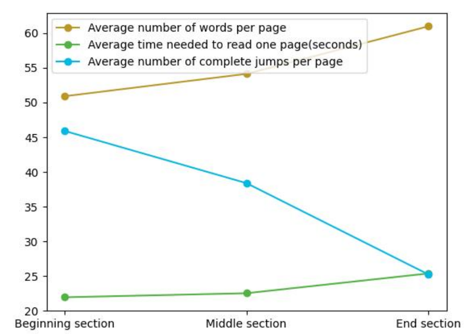

Figure 16. The average number of words, the required reading time, and the number of complete jumps in each section after segmentation.

Additionally, this paper’s recommendations focus on providing navigation suggestions to new users. When using a specific user’s data as the training set, the system can obtain personalized recommendation results. However, the data volume of a single user is inevitably small, which may lead to poorer recommendation results.

Looking ahead, to achieve personalized recommendations for e-book pages and to better implement the principles of fairness and diversity in Human-AI Interaction, it is crucial to consider more diverse learning data when building a recommendation system. For instance, using log data from different types of students at various stages of learning could help explore relevant jump needs.

Furthermore, our data revealed that an increase in concept comparison and example content reduced instances of jump back behaviors. This suggests that reading this type of content can produce an effect similar to the page jump reading strategy. Therefore, enhancing comparative content in textbook design could potentially improve students' learning experience.

6. CONCLUSION AND FUTURE WORK

In this study, we developed an e-book page recommendation model that processes textbook content through the BERTopic model and generates page jump recommendations in conjunction with student log data.

For RQ1: On a test set comprising 120 students' data, this model can provide students with a relatively accurate first choice of jump pages. However, a long tail effect emerges as the number of recommended pages increases.

For RQ2: we tried to find possible reasons in the discussion by analyzing the recommended results. In explaining the recommendation results, we explored the model's long tail effect through the recommended content and structured annotation of students. We discovered that differences in student types, diversity of student jump needs, and types of textbook content are factors contributing to the decline in model performance and the emergence of the long tail effect.

For future work, we plan to use this page recommendation system in different courses to verify the effectiveness of the recommendations. At the same time, we also plan to introduce multi-dimensional data to achieve more intelligent personalized recommendations.

7. REFERENCES

[1] Berdanier, C. G., McComb, C. M., & Zhu, W. (2020, October). Natural language processing for theoretical framework selection in engineering education research. In 2020 IEEE Frontiers in Education Conference (FIE) (pp. 1-7). IEEE. DOI=https://doi.org/10.1109/FIE44824.2020.9274115.

[2] Booth, G. J., Ross, B., Cronin, W. A., McElrath, A., Cyr, K. L., Hodgson, J. A., ... & Jardine, D. (2023). Competency-based assessments: leveraging artificial intelligence to predict subcompetency content. Academic Medicine, 98(4), 497-504. DOI=https://doi.org/10.1097/ACM.0000000000005115.

[3] Bartell, B. T., Cottrell, G. W., & Belew, R. K. 1994. Automatic combination of multiple ranked retrieval systems. In SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University. 173-181. London: Springer London. DOI=https://doi.org/10.1007/978-1-4471-2099-5_18.

[4] Costa-Dookhan, K. A., Maslej, M. M., Donner, K., Islam, F., Sockalingam, S., & Thakur, A. (2024). Twelve tips for Natural Language Processing in medical education program evaluation. Medical Teacher, 1-5. DOI=https://doi.org/10.1080/0142159X.2024.2316223.

[5] Chiba, Katsuhiro. (2017). The effect of extensive reading on English reading speed. Bulletin of the Faculty of International Studies at Bunkyo University, 28(1), 57-65.

[6] Ding, L., Liu, B., & Tao, Q. (2010, March). Hybrid filtering recommendation in e-learning environment. In 2010 Second International Workshop on Education Technology and Computer Science (Vol. 3, pp. 177-180). IEEE. DOI=https://doi.org/10.1109/ETCS.2010.378.

[7] Egger, R., & Yu, J. 2022. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. Frontiers in sociology, 7, 886498. DOI=https://doi.org/10.3389/fsoc.2022.886498.

[8] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. DOI=https://doi.org/10.48500/arxiv.2203.05794.

[9] Huang, A. Y., Lu, O. H., Huang, J. C., Yin, C. J., & Yang, S. J. 2020. Predicting students' academic performance by using educational big data and learning analytics: evaluation of classification methods and learning logs. Interactive Learning Environments, 28(2), 206-230. DOI=https://doi.org/10.1080/10494820.2019.1636086.

[10] Kang, Z., & Yin, C. 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. DOI=https://doi.org/10.1109/ICAIE56796.2023.00026.

[11] Kulkarni, P. V., Rai, S., & Kale, R. 2020. Recommender system in elearning: a survey. In Proceeding of International Conference on Computational Science and Applications: ICCSA 2019. 119-126. Singapore: Springer Singapore. DOI=https://doi.org/10.1007/978-981-15-0790-8_13.

[12] Lan, Y., Li, X., Du, H., Lu, X., Gao, M., Qian, W., & Zhou, A. (2024). Survey of Natural Language Processing for Education: Taxonomy, Systematic Review, and Future Trends. arXiv preprint arXiv:2401.07518. DOI=https://doi.org/10.48550/arXiv.2401.07518.

[13] Liang, G., Weining, K., & Junzhou, L. (2006). Courseware recommendation in e-learning system. In Advances in Web Based Learning–ICWL 2006: 5th International Conference, Penang, Malaysia, July 19-21, 2006. Revised Papers 5 (pp. 10-24). Springer Berlin Heidelberg. DOI=https://doi.org/10.1007/11925293_2.

[14] Li, Y., Niu, Z., Chen, W., & Zhang, W. (2011). Combining collaborative filtering and sequential pattern mining for recommendation in e-learning environment. In Advances in Web-Based Learning-ICWL 2011: 10th International Conference, Hong Kong, China, December 8-10, 2011. Proceedings 10 (pp. 305-313). Springer Berlin Heidelberg. DOI=https://doi.org/10.1007/978-3-642-25813-8_33.

[15] Ma, B., Lu, M., Taniguchi, Y., & Konomi, S. I. 2022. Exploring jump back behavior patterns and reasons in e-book system. Smart Learning Environments, 9(1), 1-23. DOI=https://doi.org/10.1186/s40561-021-00183-6.

[16] Myrberg, C. 2017. Why doesn't everyone love reading e-books?

[17] Prabadevi, B., Deepa, N., Ganesan, K., & Srivastava, G. (2023). A decision model for ranking Asian Higher Education Institutes using an NLP-based text analysis approach. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(3), 1-20. DOI=https://doi.org/10.1145/3534562.

[18] Purificato, E., Manikandan, B. A., Karanam, P. V., Pattadkal, M. V., & De Luca, E. W. 2021. Evaluating Explainable Interfaces for a Knowledge Graph-Based Recommender System. In IntRS@ RecSys. 73-88.

[19] Saul Carliner. 2004. An overview of online learning.

[20] Shaik, T., Tao, X., Li, Y., Dann, C., McDonald, J., Redmond, P., & Galligan, L. (2022). A review of the trends and challenges in adopting natural language processing methods for education feedback analysis. IEEE Access, 10, 56720-56739. DOI=https://doi.org/10.1109/ACCESS.2022.3177752.

[21] Sun, J., Geng, J., Cheng, X., Zhu, M., Xu, Q., & Liu, Y. (2020). Leveraging personality information to improve community recommendation in e-learning platforms. British Journal of Educational Technology, 51(5), 1711-1733. DOI=https://doi.org/10.1111/bjet.1301.

[22] Shahbazi, Z., & Byun, Y. C. (2022). Agent-based recommendation in E-learning environment using knowledge discovery and machine learning approaches. Mathematics, 10(7), 1192. DOI=https://doi.org/10.3390/math10071192.

[23] Tsai, S. C., Chen, C. H., Shiao, Y. T., Ciou, J. S., & Wu, T. N. 2020. Precision education with statistical learning and deep learning: a case study in Taiwan. International Journal of Educational Technology in Higher Education, 17, 1-13. DOI=https://doi.org/10.1186/s41239-020-00186-2.

[24] Tian, X., & Boyer, K. E. (2023). A Review of Digital Learning Environments for Teaching Natural Language Processing in K-12 Education. arXiv preprint arXiv:2310.01603. DOI=https://doi.org/10.48550/arXiv.2310.01603.

[25] Tang, T., & McCalla, G. (2004, August). Utilizing artificial learners to help overcome the cold-start problem in a pedagogically-oriented paper recommendation system. In International conference on adaptive hypermedia and adaptive web-based systems (pp. 245-254). Berlin, Heidelberg: Springer Berlin Heidelberg. DOI=https://doi.org/10.1007/978-3-540-27780-4_28.

[26] Teng, S. Y., Li, J., Ting, L. P. Y., Chuang, K. T., & Liu, H. (2018, November). Interactive unknowns recommendation in e-learning systems. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 497-506). IEEE. DOI=https://doi.org/10.1109/ICDM.2018.00065.

[27] Wan, S., & Niu, Z. (2016). A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm. Knowledge-Based Systems, 103, 28-40. DOI=https://doi.org/10.1016/j.knosys.2016.03.022.

[28] Yin, C., Okubo, F., Shimada, A., Oi, M., Hirokawa, S., Yamada, M., ... & Ogata, H. 2015. Analyzing the features of learning behaviors of students using e-books. In International Conference on Computer in Education (ICCE 2015), 617-626.

[29] Yang, R., Yang, B., Ouyang, S., She, T., Feng, A., Jiang, Y., ... & Li, I. (2024). Leveraging Large Language Models for Concept Graph Recovery and Question Answering in NLP Education. arXiv preprint arXiv:2402.14293. DOI=https://doi.org/10.48550/arXiv.2402.14293.

[30] Yin, C., Ren, Z., Polyzou, A., & Wang, Y. 2019. Learning behavioral pattern analysis based on digital textbook reading logs. In Distributed, Ambient and Pervasive Interactions: 7th International Conference, DAPI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings 21. 471-480. Springer International Publishing. DOI=https://doi.org/10.1007/978-3-030-21935-2_36.

[31] Zhang, X. 2023. MPTopic: Improving topic modeling via Masked Permuted pre-training. DOI=https://doi.org/10.48550/arXiv.2309.01015.

[32] Zhang, H., Sun, M., Wang, X., Song, Z., Tang, J., & Sun, J. 2017. Smart jump: Automated navigation suggestion for videos in moocs. In Proceedings of the 26th international conference on world wide web companion. 331-339. DOI=https://doi.org/10.1145/3041021.3054166.