Student Modeling with Automatic Knowledge Component Extraction for Adaptive Textbooks

Khushboo Thaker[1], Peter Brusilovsky[1], and Daqing He[1]

University of Pittsburgh, Pittsburgh, PA , USA
k.thaker,peterb,dah44@pitt.edu

Abstract. Online textbooks have become a significant component of online and blended learning environments. Taking this medium one step further, Adaptive online Textbooks (AoT) recommend the most relevant pages and practice activities based on students current knowledge state. AoT use student interaction data to infer the current state of student knowledge through student modeling (SM). The knowledge is inferred on knowledge components (KCs) associated with textbook material (sections/pages, practice activities, and quizzes). However, most of these techniques rely on expert annotated knowledge components. A challenge of student modeling in the context of adaptive textbooks is that traditional student models are constructed based on performance data (question answers or problem solving) Student interaction with online textbooks, however, produces large volume of student reading data, but a very limited amount of question-answering data. This leads to the requirement of annotating reading materials (textbook sections and paragraphs) with related Kcs. However, given large number of textbook sections it becomes impractical and time consuming to annotate these large components with Kcs in practice. To bridge this gap between practical and theoretical SM models in AoTs, we have proposed the use of automatic KC extraction to annotate textbook sections with KCs. This can help us to utilize current student models for AoT.

Keywords: Student Modeling · Automatic Concept Extraction · Adaptive Textbooks.

1 Introduction

AoTs are one of the oldest technologies of personalized web-based learning [23]. Present popularity and easy accessibility of electronic textbooks makes this technology more attractive than ever. State-of-the-art AoTs recommend adaptive content using simple content similarity and learners page visit patterns [10, 12]. Recently SM based approaches on student reading behavior have been proposed to make these models more sophisticated by incorporating students current knowledge state for adaptation [11, 22, 21]. In conventional SM frameworks [16, 6], student knowledge state is measured on the level of individual KCs (domain skills or concepts). The primary goal of KC-level knowledge modeling is to provide effective learning and reduce the total time spent on skill acquisition by

offering adaptive feedback, guiding the student to the most appropriate learning content. To provide adaptive feedback, the system keeps track of students' activities such as student reading time and performance on practice activities. These user interactions are later used by SM systems to distill student knowledge and predict student behavior on possible reading trajectories.

State-of-the-art SM systems require every material (textbook sections/pages, practice activities, and quizzes) to be annotated with an independent set of KCs. Traditionally, subject experts index every section of a digital textbook with a set of domain concepts [6]. Expert-based concept indexing was acceptable in the early days of the web when the volume of digital textbooks and online educational content was low, but it does not scale to abundant digital content. Recently, student models for AoT have tried to annotate these KCs automatically [11], using text mining and text extraction approaches. However, automatic KC extraction techniques output noisy as well as correlated KCs [20], breaking the independent KC assumption of existing SM and degrading their performance. This has leads to a gap between theory and practice, and most of the AoT in practice do not incorporate students' skill statistics [10, 12].

To make this process more efficient in this work we tried to incorporate automatic KC extraction techniques, to obtain better representative KCs and further evaluated them for SM in AoTs.

2 Related Work

2.1 Student Modeling in ITS

Approaches in student modeling in ITS could be classified into two major groups: Logistic Regression models and Knowledge Tracing models [17]. Logistic regression models are motivated by the power law of learning [15], which states that probability of applying a skill correctly decreases by a power function. These models utilize student observation logs as the inputs, and try to predict student performance with a learning activity based on KCs (skills) associated with the activity. One of the earlier models in this group is known as Additive Factor Model (AFM) [5], which computes the odds of a student's success on a particular question based on the number of previous attempts. Performance Factor Analysis [16] improves AFM by separately modeling the student's previous successes and failures on a particular skill.

Knowledge Tracing (KT) model was introduced in 1995 by Corbett and Anderson [6]. KT uses Hidden Markov Models (HMM) to represent student knowledge as binary latent variables. Each latent variable represents student knowledge of a particular KC, which could be either known or unknown. The observed variable is the performance of student at a given step, which is measured as a binary variable representing the correctness of a step or an answer (correct or not correct). KT directly represents KC-level knowledge estimation and allows dynamic knowledge update at each student learning opportunity.

In this work, we would like to utilize both regression based models for automatic concept extraction (ACE). We would leave it for future work to extend these models for KT framework.

2.2 Adaptive Online Textbooks

The research on adaptive textbooks has been motivated by the increasing popularity of World Wide Web (WWW) and the opportunity to use this platform for learning. The hypertext nature of early WWW made an online hypertext-based textbook a natural media for learning while the increased diversity of Web users stressed the need for adaptation. The first generation of adaptive textbooks [3, 7, 10, 12] focused on tracing student reading behavior to guide students to most relevant pages using adaptive navigation support [3, 7, 10, 23] or recommendation [12]. These types of personalization were based on a sophisticated knowledge modeling: each textbook page was associated with a set of concepts *presented* on the page as well as concepts *required* to understand the page [3, 7]. On the other hand, SM was relatively simple: these systems treated each visit to a page as a contribution to learning all presented concepts.

A significant trend of modern online textbooks is the increased inclusion of interactive content "beyond text". While the attempts to integrate online reading with problem solving have been made in the early days of online textbooks [23], it was a rare exception. Modern textbooks, however, routinely integrate a variety of "smart content" such as visualizations, problems, and videos. In this context, the ability to integrate data about student work with all these components and use it for a better-quality SM becomes a challenge for modern online textbooks.

2.3 Reading for Adaptive Textbooks

Reading is a cognitive process whereby the reader builds a situation model of text to comprehend[13] the text. Several computational models are being studied to understand reading behavior[13, **?**], which try to infer readers comprehension. A recent trend in student modeling research is to incorporate student reading behavior [11, 22, 8] to incorporate student comprehension. Eagle et al. [8] were among the first to incorporate student reading rate in a knowledge tracing model. Their study depicted the positive effect of integrating students' reading rate to provide individualization. Huang et al. [11] also modeled student reading behavior using a knowledge tracing model for online adaptive textbooks, by learning students skimming and reading behavior. Across these efforts, the key idea is to provide content adaptation based on the student's knowledge state. The model has a strict assumption that students' reading rate is positively correlated with their performance. However, this assumption does not hold for all students [1]. Thaker et al. [22] addressed this limitation by integrating both practice activities and reading interactions to deal with students' noisy reading behavior. Furthermore, recently, Carvalho et al. [4] investigated the effect of attempting optional reading exercises in MOOCs. Their study suggested that attempting optional

reading activities helps to boost students' performance and learning[4]. In a recent study Thaker et. al. incorporated student reading behavior in regression based models [16, 5] to model student activity performance [21]. The model again relied on expert annotated concepts both for reading text and practice activities. Our attempt in this work is to try different possibilities of ACE and analyze the possibility of using them as KCs.

3 Automatic Concept Extraction

Concept-based textbook indexing was introduced by early projects focused on adaptive textbooks [10, 24]. In these systems every section of a digital textbook was associated with a set of domain concepts (called outcomes) that are present in that section. This concept-level indexing of educational content was used to model student progress and to recommend sections to read. However, these methods depend on manual concept identification, which is performed by domain experts. For ACE, we explored phrase based extraction techniques and tried to incorporate them to Student Models. These methods were evaluated against gold concept dataset which was annotated with experts. In this section we will discuss the techniques used for ACE

3.1 Term-based (TFIDF)

We started with a simple approach based on words importance in a reading unit. Here, we applied the traditional TF*IDF (Term Frequency - Inverse Document Frequency) approach[19]. For each textbook section (reading unit) top-N TF*IDF-weighed words were extracted and considered as KCs for that section. Note that before TF*IDF weighting and KC extraction, each document is tokenized by stop-word removal, excluding non-letter symbols (e.g. punctuation marks and digits) and finally stemmed by Porter stemmer [18].

3.2 Noun Phrase Chunking(TFIDF-NP)

In this approach, we use a two-step automatic indexing of textbook sections with concepts. The first step generates a list of candidate concepts by applying noun phrase chunking[14]. The assumption here is all the concepts will occur as noun phrases in the text. Next the candidate noun phrases are ranked based on their TF-IDF score[19] and then top N high scoring noun phrase are selected as concepts representing the document. We will refer to this concept extraction methods as TFIDF-NP.

3.3 Wikipedia[1] based filtering(Wiki-NP)

In this approach, we use a two-step automatic indexing of textbook sections with concepts. The first step is similar to *TFIDF-NP* and generates a list of

[1] http://www.wikipedia.org

candidate concepts by applying noun phrase chunking[14]. The next step filters these noun-phrase using a Wikipedia page names. Here the assumption is that concepts/topics taught in the page will have a corresponding Wikipedia page.

3.4 Latent Dirichlet Allocation (LDA)

Term-based similarity, is not able to capture the semantic relations in the text. To test proposed models against semantic representation baseline we considered using LDA. This paper implements the vanilla version of LDA as proposed by Blei et al. [2]. LDA is a unsupervised approach which models each text unit as distribution over 'k' latent topics. LDA was trained on all the textbook sections of the course. The value of $k = 200$ (number of topics), which performed best among the models trained on $k = 10, 20, 50, 100, 150, 200, 250$ topics. Trained LDA was used to annotate each textbook section as well as practice activities with corresponding LDA topic.

4 Experiments

4.1 System and Dataset

The dataset used for the experiment is collected from online reading platform Reading Circle [9] in Spring 2016. This system was used for graduate level course on Information Retrieval at University in North America. The system provides an active reading environment to the student where they read the assigned textbooks material to prepare for the next class. To keep students motivated to use the system for reading, the system provides feedback about students reading progress as well as average class reading progress. Each section of the assigned textbook reading is followed by a quiz with several questions, which allow students to assess how well they learned the content. There is no restriction on the number of attempts to the questions, Reading Circle logs each and every attempt made by the student. The final dataset contains 22,536 interactions from 22 students (see more details in Table 1).

Table 1. Dataset Statistics

Number of documents (sections)	394
Number of questions	158
Number of students	22
Median per student of reading time (minutes)	104
Average per student questions attempted	126
Median Reading Speed (words per minutes)	773
Percentage of skimming Activities	33%
Percentage of reading Activities	67%

4.2 Reading Data Pre-processing

The reading logs are noisy. A student can open a course content, start reading and then leave for some personal work, as the system will remain open until time out, this will generate a log that suggests the student was reading that content. Similarly, students might open the page and immediately try the activities or open another page. To handle this noise, we took calculated reading speed for each page and adjusted the records which were beyond the student reading limit. The general student reading speed was considered between 400 to 800 words per minute (wpm). To calculate reading speed we divided the number of words on the page by the minutes student spend on the page.

4.3 Model details

To incorporate students reading behaviour we used the existing Comprehensive Factor Analysis *CFM* model [21]. *CFM* is an extension of *PFA*, with the addition of student reading activities as a predictor of student's success in the step. The model assumes that students skill mastery improves with the opportunities the student have to read materials associated with the skill. One reading opportunity is a duration for which a student has the text page opened. Thus reading opportunity starts when the student visits a particular page and it ends when the student starts performing practice activities on that page or leaves the page to visit another page. The Below equation defines *CFM* model.

$$\textbf{CFM-RO:} \quad \ln \frac{p_{ij}}{1-p_{ij}} = \alpha_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj}(\mu_k S_{ik} + \rho_k F_{ik} + \zeta_k RO_{ik}) \tag{1}$$

where, i is a student, j is a step. k is a *Skill*. α_i is a coefficient associated with student i (regression intercept) and represents the proficiency of student i. Q is a Qmatrix and Q_{kj} is Qmatrix cell associated with item j and *Skill* k. β_k represents the difficulty of skill k. S_{ik} and F_{ik} as number of success and failure attempts respectively of student i on skill k. ζ_k is the coefficient which measures the learning rate of a skill from reading opportunities and RO_{ik} is the number of reading opportunity student i has on skill k.

4.4 Conclusion

In this paper we proposed few possible solutions to extract KCs automatically given a large text. We have also formulated a possible way to evalute these KC extraction techniques for student performance prediction task. In future, we would like to discuss the results we obtained and more exploratory analysis to understand different methods proposed

References

1. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z.: Off-task behavior in the cognitive tutor classroom: When students "game the system". In: Proc. ACM Conf. on Human Factors in Computing Systems. pp. 383–390. CHI '04 (2004)

2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**, 993–1022 (2003)
3. Brusilovsky, P., Eklund, J.: A study of user-model based link annotation in educational hypermedia. J. of Universal Computer Science **4**(4), 429–448 (1998)
4. Carvalho, P.F., Gao, M., Motz, B.A., Koedinger, K.R.: Analyzing the relative learning benefits of completing required activities and optional readings in online courses. In: The 11th Int. Conf. on Educational Data Mining. vol. 34, p. 68
5. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis: A general method for cognitive model evaluation and improvement. In: Intelligent Tutoring Systems, Lecture Notes in Computer Science, vol. 4053, pp. 164–175. Springer Berlin / Heidelberg (2006)
6. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modelling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction **4**(4), 253–278 (1995)
7. De Bra, P.: Teaching through adaptive hypertext on the www. Int. Journal of Educational Telecommunications **3**(2/3), 163–180 (1997)
8. Eagle, M., Corbett, A., Stamper, J., McLaren, B.M., Wagner, A., MacLaren, B., Mitchell, A.: Estimating individual differences for student modeling in intelligent tutors from reading and pretest data. In: Micarelli, A., Stamper, J., Panourgia, K. (eds.) Intelligent Tutoring Systems. pp. 133–143. Springer International Publishing, Cham (2016)
9. Guerra, J., Parra, D., Brusilovsky, P.: Encouraging online student reading with social visualization. In: The 2nd Workshop on Intelligent Support for Learning in Groups at the 16th Conf. on Artificial Intelligence in Education. pp. 47–50 (2013)
10. Henze, N., Naceur, K., Nejdl, W., Wolpers, M.: Adaptive hyperbooks for constructivist teaching. Künstliche Intelligenz **13**(4), 26–31 (1999)
11. Huang, Y., González, J., Kumar, R., Brusilovsky, P.: A framework for multifaceted evaluation of student models. In: Proc. the 8th Int. Conf. on Educational Data Mining. pp. 203–210 (2015)
12. Kavcic, A.: Fuzzy User Modeling for Adaptation in Educational Hypermedia. IEEE Transactions on Systems, Man and Cybernetics **34**(4), 439–449 (2004)
13. Kintsch, W., Welsch, D.M.: The construction-integration model: A framework for studying memory for text. (1991)
14. Kudo, T., Matsumoto, Y.: Yamcha: Yet another multipurpose chunk annotator (2005)
15. Newell, A., Rosenbloom, P.S.: Mechanisms of skill acquisition and the law of practice. Cognitive skills and their acquisition **1**, 1–55 (1981)
16. Pavlik, P., Cen, H., Koedinger, K.: Performance Factors Analysis–A New Alternative to Knowledge Tracing. In: Proc. the 2009 Conf. on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. pp. 531–538. IOS Press (2009)
17. Pelánek, R.: Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. User Modeling and User-Adapted Interaction **27**, 313–350 (2017)
18. Porter, M.: An algorithm for suffix stripping. Program **14**(3), 130–137 (mar 1980)
19. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management **24**(5), 513–523 (1988)
20. Thaker, K., Brusilovsky, P., He, D.: Concept enhanced content representation for linking educational resources. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). pp. 413–420. IEEE (2018)

21. Thaker, K., Carvalho, P., Koedinger, K.: Comprehension factor analysis: Modeling student's reading behaviour: Accounting for reading practice in predicting students' learning in moocs. pp. 111–115. LAK19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3303772.3303817
22. Thaker, K., Huang, Y., Brusilovsky, P., Daqing, H.: Dynamic knowledge modeling with heterogeneous activities for adaptive textbooks. In: The 11th Int. Conf. on Educational Data Mining. pp. 592–595 (2018)
23. Weber, G., Brusilovsky, P.: ELM-ART: An adaptive versatile system for web-based instruction. Int. Journal of Artificial Intelligence in Education **12**(4), 351–384 (2001)
24. Weber, G., Brusilovsky, P.: Elm-art: An adaptive versatile system for web-based instruction. Int. Jour. of Artificial Intelligence in Education **12**, 351–384 (2001)