A Framework for Dynamic Knowledge Modeling in Textbook-Based Learning

Yun Huang
Intelligent Systems Program
University of Pittsburgh
210 S. Bouquet Street
Pittsburgh, PA, USA
yuh43@pitt.edu

Michael Yudelson
Human-Computer Interaction
Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, USA
yudelson@cs.cmu.edu

Shuguang Han
School of Information
Sciences
University of Pittsburgh
135 N. Bellefield Ave.
Pittsburgh, PA, USA
shh69@pitt.edu

Daqing He
School of Information
Sciences
University of Pittsburgh
135 N. Bellefield Ave.
Pittsburgh, PA, USA
dah44@pitt.edu

Peter Brusilovsky
School of Information
Sciences
University of Pittsburgh
135 N. Bellefield Ave.
Pittsburgh, PA, USA
peterb@pitt.edu

ABSTRACT

Various e-learning systems that provide electronic textbooks are gathering data on large numbers of student reading interactions. This data can potentially be used to model student knowledge acquisition. However, reading activity is often overlooked in canonical student modeling. Prior studies modeling learning from reading either estimate student knowledge at the end of all reading activities, or use quiz performance data with expert-crafted knowledge components (KCs). In this work, we demonstrate that the dynamic modeling of student knowledge is feasible and that automatic text analysis can be applied to save expert effort. We propose a data-driven approach for dynamic student modeling in textbook-based learning. We formulate the problem of modeling learning from reading as a reading-time prediction problem, reconstruct existing popular student models (such as Knowledge Tracing) and explore two automatic text analysis approaches (bag-of-words-based and latent semantic-based) to build the KC model. We evaluate the proposed framework using a dataset collected from a Human-Computer Interaction course. Results show that our approach for reading modeling is plausible; the proposed Knowledge Tracing-based student model reliably outperforms baselines and the latent semantic-based approach can be a promising way to construct a KC model. Serving as the first step to model dynamic knowledge in textbook-based learning, our framework can be applied to a broader context of open-corpus personalized learning.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions.acm.org.

UMAP '16, July 13-17, 2016, Halifax, NS, Canada

© 2016 ACM. ISBN 978-1-4503-4370-1/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2930238.2930258>

Keywords

textbook-based learning, reading, learner model, Knowledge Tracing, Additive Factor Model, Performance Factor Analysis, latent topic modeling, text analysis

1. INTRODUCTION

The steadily increasing volume of online educational content makes it harder for learners to find *appropriate content* that matches their individual goals, interests, and knowledge. In the past, adaptive hypermedia (AH) techniques attempted to address this problem by providing personalized adaptive navigation support to help individual students locate, recognize, and comprehend relevant information. Evaluations of adaptive navigation support techniques have demonstrated their ability to increase learning outcomes, retention, and efficiency of student work [7, 14, 24].

Unfortunately, existing adaptive navigation support techniques can only work efficiently within a closed corpus of documents whose domain concepts and other metadata have been manually identified and indexed at the time of the system design. The open corpus educational systems of today, when the whole web could be considered the content base, present new challenges for traditional AH techniques. From these challenges, we want to address the following three: (1) how to determine the knowledge components (KCs, or skills, concepts) behind each unit of open corpus content without human engagement; (2) how to maintain dynamic student knowledge models on the level of automatically identified KCs with only reading interactions; (3) how to apply our inferred student knowledge about KCs underlying the content to make personalization decisions.

In this paper, we present our attempt to construct dynamic student knowledge models for personalized guidance in the context of online textbook-based learning environments. Such a context enables us to readily apply our approaches to a broader context of open-corpus personalized learning and to address the above-mentioned AH challenges with an open corpus. Our key idea is to automatically ex-

tract *elements of meaning* from the text and to adopt these individual elements as knowledge components for student modeling. However, current popular student models rely on in-time quiz performance data, which is often unavailable in our context. To address this, we utilize the available abundant reading interaction data and formulate the problem as a reading time prediction problem, so that existed popular student models can be applied after simple modifications. This student modeling approach allows us to implement an in-time personalized guidance approach that can distinguish between reading and skimming content pages given a student’s current level of knowledge.

Specifically, for content modeling (or knowledge component modeling), we explore both bag-of-words-based approach [36] and latent semantic topic modeling approach [2]. In terms of modeling student learning, we mainly employ a widely-adopted Knowledge Tracing (KT) model [13] that models the learning process by hidden Markov models [34], and we compare it with the state-of-the-art logistic regression-based models of student learning. In the following sections, we will provide detailed descriptions of prior work, our methods and experimental results.

2. BACKGROUND

With our main research focus on tracing students’ knowledge in textbook-based learning applying predictive student models, related research of our study lies in the following three areas: difficulty assessment for learning materials, predictive evaluation for students’ knowledge modeling, and automatic educational content modeling.

2.1 Difficulty Assessment in Adaptive Hypermedia

Adaptive electronic textbooks frequently guide users by distinguishing between “ready to be learned” content, which bears new information without being too difficult to prevent users from understanding, and “not ready to be learned” content, which is too complicated for the current state of the user’s knowledge. With fine-grained domain models that index content pages at the concept level, adaptive navigation support can be provided both easily and reliably by assessing the fraction of concepts that the user has already learned and by checking poorly known prerequisites. For example, the “traffic light” approach using a “traffic light” icon to annotate links as ready/not ready content, was introduced in the ELM-ART system [41]. Since then, this approach has been replicated in numerous educational AH systems. Later, this approach was also extended to cases that lack prerequisite links and thus doesn’t allow the system to reliably judge content as “not ready.” In these cases, links to content are annotated with “knowledge progression” icons presenting to what extent the user might already know the content [8, 21, 31]. Many studies confirmed the effectiveness of these approaches in the context of learning from online textbooks [3, 8, 14, 20, 31, 41].

2.2 Predictive Evaluation of Student Models

The approaches to building and evaluating student models and adaptive systems have evolved significantly over the last two decades. The classic approaches to evaluating user models and adaptive systems were mostly empirical (see [12] for a good review). A frequently used approach is A/B testing. It compares user performance and behavior in two versions

of a system: one with and one without adaptation. However, during an empirical evaluation, many factors contribute to the outcome, and the quality of the user model is just one of many. As a result, it is difficult to assess the quality of just the user modeling alone. To address this problem, several researchers advocated the use of the *layered evaluation* of adaptive systems [4, 32, 38] where each *layer* that contributes to the performance of the adaptive system (such as interface or user modeling) is evaluated separately.

In a layered evaluation context, the prediction accuracy evaluation approach emerged as a de-facto standard to isolate and assess the student modeling component of the personalized learning system. Predictive accuracy evaluation replaced live user studies with fully-automatic assessment based on real user data collected during the learning process. The idea is to split user data into two parts (training and testing). The training part of the data is used to build a student model, which is then used to predict student performance in the held-out testing part. The student model is then assessed by comparing the predicted performance to the actual performance obtained from the testing data. The classic work of Knowledge Tracing [13] presented several predictive evaluations in order to demonstrate its useful properties. Many recent studies [44, 16, 25] also compared and evaluated student models by performing a cross-validated evaluation, and they paid attention to the plausibility of the model parameters as well. The most recent internal evaluation framework [22] proposed multifaceted aspects including predictive accuracy, parameter plausibility, and consistency, and the work compared some state-of-the-art student models [17] from these multifaceted aspects.

2.3 Automatic Educational Content Modeling

Automatically modeling the content of documents has been viewed as the basis for various tasks, including user modeling [6] and text classification [37]. There are several methods that can be used for automatic content representation. The simplest approach is the bag-of-words-based approach, where the frequencies of words (terms) in both the documents and the collection are counted [26, 28, 36]. Despite its simplicity, this method has been among the most frequently used approaches for calculating the similarity among different fragments of educational content [19]. However, its limitation of not being able to capture the knowledge components inside the domain, particularly in domains related to education, is also well documented [19].

Therefore, the literature has also explored the automatic representation of educational materials considering semantics. For example, The Conceptual Open Hypermedia Services Environment (COHSE) [1] proposed a representation of online educational documents with metadata, and through integrating with an ontological reasoning service, it can form a conceptual hypermedia educational system. More recently, various latent semantic-based approaches such as topic modeling have gained attention in automatic content modeling [40]. By representing the content of documents as finite mixtures over an underlying set of latent topics, topic modeling moves beyond a bag-of-words representation, and avoids the rather expensive development cost in some earlier semantic modeling methods. In particular, Latent Dirichlet Allocation (LDA) [2] has been explored many times in content modeling [9, 35], including in supporting semantic representation in the fine-grained textbook linking [19].

3. METHODS

3.1 Problem Statement

Traditionally, student knowledge is inferred from performance data, which is collected from students’ activities on questions, problems, quizzes, tests or exams. Such data carries a relatively high certainty for inferring student knowledge. However, textbook-based learning behavior investigated in this paper almost exclusively consists of students’ reading activities. Problems or questions, even if in cases when they are provided as a part of reading-oriented learning, are usually found at the end of relatively large reading sections, and can only serve as a delayed assessment of student knowledge. In this situation, to maintain a highly dynamic student model that is necessary to deliver just-in-time personalization experience, we need to learn how to use students’ online reading interactions to estimate changes in the students’ dynamic knowledge. One benefit of using students’ reading behaviors is that they are the most abundant data obtainable from the system logs in our context.

Prior literature supports our idea of linking students’ reading time to their overall knowledge growth. For example, research on learning curves [29, 27] confirms that with the increases in practice opportunities, students’ problem-solving time decreases, following a power law. Although reading might require less complex cognitive process than problem-solving, we argue that it is also an activity that builds up fluency through “repetition” as a learning process. In this case, “repetition” means reading the content (e.g., document, page) that contains previously covered knowledge components. In other words, students’ knowledge of a knowledge component should grow as students spend time reading the related content. Meanwhile, once a student acquires the knowledge underlying a piece of learning content, they would generally spend little time on the content.

More formally, we have made two assumptions:

- If a student spends time reading a document, the student’s knowledge of the knowledge components associated with the document will grow.
- If a document contains knowledge components that a student has already learned, the student is highly likely to skip or skim the document; if a document contains new knowledge components the student has not yet learned, the student is highly likely to read the document carefully. Note that like most models, our model simplifies real-world experience in order to quantify it in an efficient way. It assumes a “diligent student” who is motivated to learn and reads with few distractions. It also assumes that knowledge is the only factor that drives students’ reading behaviors.

The above-mentioned student learning process is illustrated in Figure 1, where we assume that there are N knowledge components (from KC_1 to KC_n) underlying a set of reading materials (documents). For each student, the knowledge level of a knowledge component can be learned or unlearned. Through consistent content reading that contains the same knowledge component, a student can transition from the unlearned to the learned state with some probability. Also, given the knowledge levels of a set of KCs, the time that a student needs to spend reading a new document can be predicted. This formulation directly translates the problem into what traditional student modeling can address.

Instead of using performance on problems (quizzes) as observations, we can use reading time. Furthermore, to evaluate the inference of knowledge, we can examine the prediction of reading time on the held-out data. In contrast to traditional classroom studies, this method requires less time and labor, and can provide valuable insights for performing studies. In addition to evaluating predictive performance, we also examine each KC model’s potential impact for real-world adaptive learning by considering its semantics, granularity, and the ability to capture transfer learning.

Note that in this hypothesized learning process, we assume binary states for both observed reading time (*Skim* or *Read*) and hidden knowledge levels (*Learned* or *Unlearned*). As mentioned before, this is a simplification for efficiency, particularly when applying classic student models which have the same binary state assumptions. Here, *Skim* corresponds to *Skim Reading* or *Scanning*, and *Read* corresponds to *Receptive Reading* or *Reflective Reading* from the typology of reading proposed in [30]. Since the logged reading time is a continuous variable, we conduct discretization on the data (see Section 4.1.2 for details). Next, we will explain how we reconstruct traditional probabilistic student models and apply automatic text analysis to model reading.

Figure 1: An illustration of knowledge modeling in textbook reading. KC_n represents the n^{th} knowledge component (KC) and L_i denotes the students’ knowledge state for each KC at i^{th} learning opportunity. ΔI indicates the content that a student reads.

3.2 Student Models for Reading

In this paper, we compare two schools of student models for reading: one uses latent variables empowering latent knowledge estimates, while the other is based on logistic regression models and serves as a high baseline.

We also compare each school of student models with a low baseline, *majority class model* (*Majority*). This model makes predictions for the target variable (reading time) by using the majority class based on its distribution of the data. For example, in our final dataset, there are around 67% activities labeled as *Read*, so this model always predicts *Read* for observations.

3.2.1 Knowledge Tracing-Based Model

Knowledge Tracing (KT) [13] has been the de-facto standard for inferring student knowledge from performance data. The classic KT model relies on decomposing knowledge in a domain into individual pieces of knowledge components (KCs), and traces students’ knowledge acquisition for each KC through a hidden Markov model (HMM) [34].

We use the classic KT paradigm and transform reading modeling into a traditional performance modeling problem. The variables in Knowledge Tracing for each knowledge component and their corresponding variables in our model are:

- **Observed variable:** a binary performance variable that indicates whether a question is answered correctly or not. In our reading context, this corresponds to a binary reading time variable with two possible states - *Skim* or *Read*).
- **Hidden variable:** a binary knowledge state variable with two possible states - *Learned* or *Unlearned*.

A classic KT has four parameters for each knowledge component (assuming no *forgetting* so that the probability of transferring from a Learned to an Unlearned state is zero). These four parameters and their corresponding parameters in our model are as follows:

- **P(L₀):** the probability that a student initially knows the KC, i.e., the student is in the learned state.
- **P(T):** the probability that a student transitions from an unlearned to a learned state.
- **P(G):** the probability that a student “guesses” correctly, given that the student is in the unlearned state. In our case, it is the probability of a student *skims* when being in the unlearned state.
- **P(S):** The probability that a student “slips,” given that the student is in the learned state. In our case, it is the probability of a student *reads* when being in the learned state.

If we assume each reading activity as an opportunity to learn a single KC (i.e., each document only involves one KC), then each student’s reading time on documents related to a KC can be ordered into a sequence, and all such sequences of a KC can serve as the input for training a KT model (an HMM) that is directly used in prediction. However, reading is a complex learning process, since it usually involves multiple KCs at the same time when a student reads a document. It is not clear how much attention each KC is allotted, based on the observation of time on the document (learning content) level. Traditional KT is constructed based on the decomposition of knowledge into units, where each unit’s measurement can be directly collected. As a result, we have to develop some mechanisms in the KT framework to address the attention attribution issue in reading.

As a first attempt, we propose a simple mechanism similar to the one used in [16]. Despite its simplicity, such kind of mechanisms has achieved a relatively good level of predictive accuracy in some previous reported results on the quiz performance data [16, 42]. This mechanism can be explained via the following steps:

Parameter Learning: During the parameter learning (training) process, we assume that each KC has equal responsibility for the observed reading time, and that they are independent of each other. This means we duplicate a single observation multiple times to make sure that each underlying KC has one observation, which forms each KC’s learning sequence. For example, if a student *Skims* a document that contains KCs KC_1 , KC_2 , KC_3 , we duplicate this activity three times (labeled as *Skim*), and assign it to each of the three KCs. Then we can train an HMM for each KC as in the traditional KT.

Predicting: After the parameter fitting process, in order to perform prediction on a new document, we have to aggregate the prediction (predicted probability of *Skim*) from each underlying KC of this document. Prior literature provides several options for aggregation [16, 42, 17]: (1) multiplying each KC’s predicted probability; (2) using the lowest prediction probability among all KCs (i.e., choosing the weakest); or (3) computing the average of each KC’s predicted probability. We argue that the first strategy is not suitable, since a reading document typically involves a number of KCs, which easily makes the resulting probability very small. In our preliminary study we found out that the second strategy behaved the same as a majority class model, in that it ended up always predicting *Read*. Therefore, we chose the third strategy for our experiments. Equation 1 shows that the predicted probability of a student skimming a document D_i at the t^{th} learning opportunity is the average of the predicted probabilities from each required KC k underlying this document:

$$P(D_i^t = Skim) = \frac{1}{N} \sum_{k \in D_i} \left(P(L_k^t = Learned) \times (1 - P(S)_k) + P(L_k^t = Unlearned) \times P(G)_k \right) \quad (1)$$

Updating: After performing prediction on a new document, the actual observation of this document can be used to update the belief of each KC’s knowledge. We assume that such new evidence will have impact on all of the KCs underlying the document, so we update all the involved KCs. Equation 2 shows how we compute the posterior probability of a student in the learned state of KC k observing this student *Skimming* a document D_i at t^{th} learning opportunity based on the prior probability. Equation 3 shows how we further compute the prior probability (up-to-date estimate) of the student in the learned state of KC k at the next $(t + 1)^{th}$ learning opportunity, based on the transition probability $P(T)_k$. For updating the knowledge based on an observation of *Read*, the procedure is similar.

$$\begin{aligned} P(L_k^t = Learned)_{post} &= P(L_k^t = Learned | D_i^t = Skim) \\ &= \frac{P(L_k^t = Learned, D_i^t = Skim)}{P(D_i^t = Skim)} \\ &= \frac{P(L_k^t = Learned)_{prior} \times (1 - P(S)_k)}{P(D_i^t = Skim)} \end{aligned} \quad (2)$$

$$\begin{aligned} P(L_k^{t+1} = Learned)_{prior} &= P(L_k^t = Learned)_{post} \\ &\quad + P(L_k^t = Unlearned)_{post} \times P(T)_k \end{aligned} \quad (3)$$

3.2.2 Logistic Regression-Based Models

There are several logistic regression-based models that are traditionally used in modeling student learning. Rasch (1PL IRT) model [39] does not model learning per se, but is used as a basis for other approaches. Rasch model assumes that the correctness of a student’s response on an item (a question or a problem) depends on the student’s ability and the corresponding item’s difficulty. The Additive Factor Model (AFM) [11] and the Performance Factors Analysis (PFA) model [33] extend the Rasch model by actually capturing the process of learning. Instead of considering item difficulties, AFM considers knowledge components (KCs, or skills) that

are linked to the corresponding item. It models student performance as a function of the student ability, the difficulty and the learning from accumulated practices of each underlying KC of the current item. Moreover, the student ability parameters can be treated as random effects [44]. Compared with AFM, traditional PFA removes student ability parameters and differentiates learning from successes and learning from failures.

Traditional AFM and PFA are constructed following the rationality (assumption) that practices should increase (the belief of) the knowledge level, as well as the chance to succeed. As explained in Section 3.1, our hypothesized learning process of reading follows a similar rationality: accessing documents about a particular KC should increase (the belief of) the knowledge level and the likelihood of skimming next documents about the same KC. Also, the discretized reading time as the dependent variable is still binary. As a result, the logistic regression models that are traditionally applied to modeling student quiz (problem) performance are directly applicable to modeling reading. We choose *Skim* as a *success* and *Read* as a *failure*. As explained in Section 3.1, this is a simplification based on assuming that students are diligent, and that they will read and only read a document when there are underlying unlearned KCs.

Equation 4 shows how we reconstruct PFA for modeling reading (similar idea can be used to reconstruct AFM). We compute the predicted probability of a student skimming a document D_i at the t^{th} learning opportunity, based on all required KCs' corresponding counts of this student's previous *skim* activities (S_k) and *read* activities (R_k) on documents that require the same KC k :

$$P(D_i^t = Skim) = logistic(\sum_{k \in D_i} (\alpha_k + \beta_k S_k + \rho_k R_k)) \quad (4)$$

where α_k , β_k , ρ_k are coefficients interpreted as initial easiness, learning rates from previous *Skim*, or *Read* activities of each KC k .

As opposed to KT, AFM and PFA models avoid local optima when fitting the parameters, and they both have a natural mechanism to handle multiple skills per item observations by the logistic regression formulation. Prior studies have shown that PFA outperforms KT in prediction particularly in multiple skills per item cases [33, 16]. So, AFM and PFA models serve as high baselines for prediction in our study. However, AFM and PFA lack the ability to explicitly represent knowledge estimations for each individual knowledge component which can be helpful for personalization. A variety of statistical packages are available to fit mixed-effect logistic regressions (AFM) or generalized linear models (PFA).

3.3 Extracting Knowledge Components

A knowledge component (*KC*) is the basic knowledge unit (e.g., concept, skill) to accomplish steps in a task or a problem. In prior studies about students' knowledge modeling [13, 17], *KC* is often predefined and manually crafted by domain experts; however, this process can be time-consuming in an open corpus, and particularly in our context of textbook-based learning. In this paper, we explore different methods for automatically extracting *KC* from textbooks.

The simplest approach is to treat each single *word* (term) in a document as a *KC* under the bag-of-words assumption.

Although it provides fine-grained granularity and potentially captures transfer learning across a book (since words can be shared across a book), it fails to capture the semantic relationships among different words. As a result, we explore several additional approaches that aim to capture the underlying semantics among different words.

We try two approaches for estimating semantic-based knowledge components – a coarse-grained approach that estimates each *chapter* within a textbook as a KC and a fine-grained approach that mines *latent topics* as KCs using the latent Dirichlet allocation (LDA) [2] model. In addition to the granularity difference between these two approaches, they also differ in their ability to capture transfer learning: the former fails to capture transfer across chapters, since each document is mapped to one chapter only used for estimating the current chapter's knowledge level, while the latter captures transfer, since topic modeling allows different documents to share latent topics in a probabilistic way. A comparison of these approaches can be seen in Table 1 and the details are explained in Section 4.1.3.

4. EXPERIMENTS AND RESULTS

We conduct three comparison studies that demonstrate the ability of the proposed KT-based model in modeling textbook reading. We will start with description of the experiment setup in Section 4.1, then present our studies investigating different KC models under KT-based and logistic regression-based models in Section 4.2, and finally compare these two schools of student models for modeling reading.

4.1 Experiment Setup

4.1.1 System and Dataset

Our dataset was collected from two online reading platforms: *AnnotatED+* [5] and *Reading Circle* [18], which were used for a graduate course about Interactive Systems Design at the University of Pittsburgh from 2007 to 2015. The collection of online readings for the class included five textbooks on human-computer interaction. The online reading systems have many functions for engaging and helping students to learn: for example, students can leave and see comments during reading, and each student's reading progress is visualized. Every week, students were assigned reading assignments; and every other week, they were asked to take a short quiz in the class that contained questions that were highly related to the reading materials. Although it was not mandatory for students to use the reading platform, many students found it to be useful and used the system for reading the textbooks.

This dataset contains students' logged time of accessing each document of a book (we call it an activity). Each document contains one to more pages and typically corresponds to a subsection (the lowest level) of a book. In order to get more reliable information for modeling, we conducted the following filtering of the data: we removed students who read fewer than one unique document; we removed the per-page activities with too much time (> 13 minutes, which is the 95th percentile of the distribution); and we removed the documents with fewer than 10 activities in total (to reduce bias in discretization in Section 4.1.2).

The final dataset contains 10,188 activities on 325 documents from 289 students. There are around 33% activities labeled as *Skim*, and thus the majority class is *Read* (67%).

The mean (median) values of some attributes of the dataset are reported as follows: the total number of documents per student is 35 (10); the total reading time per student is 57 (13) minutes; the number of activities per document is 31 (24); and the reading time per document over all activities totaled 1.6 minutes (22 seconds).

4.1.2 Discretization of Time

We discretized time based on each document’s reading time distribution (across activities), because our documents have a high variance in length ($SD=314$ words), which we considered as the primary factor that affects reading time, similar to [15]. In our preliminary study, we found out that the 33rd percentile of the time distribution per document constitutes a reasonable cutoff to differentiate *Skim* and *Read*. We first computed the reading speed for each activity by dividing the number of words in the corresponding document by the time spent on this document. The distribution of learning speeds across activities was highly skewed, so we obtained the median as 6 words per second, and treated it as the normal reading speed in our context. This speed is above the reported speed of an average-speed reader [23] and below the reported speed of the fastest college graduate readers [10]. We then computed the reading speed by the 33rd percentile time for each document and obtained the median as 20 words per second from the skewed distribution. We considered it as the minimum skimming speed (an activity with a reading speed above this is labeled as *Skim* and an activity with a reading speed below this is labeled as *Read*). We found out that this speed is about three times faster than the normal reading speed, which is consistent with the finding in a classic work [23] that found that speed readers and skimmers were 3 times to 2.5 times faster than normal readers. As a result, we chose this discretization method for our experiments.

Admittedly, our discretization does not consider students’ individual differences, yet incorporating this aspect is non-trivial: measuring a stable reliable learning speed might require tests on students before they use the system, which may reduce students’ engagement or may not be available for existing platforms, and it might require collecting enough observations and then conducting time-consuming online parameter estimations while students are using the system. It is also unclear as to whether the learning speed can be considered stable. In our current document based discretization, the system can immediately perform modeling even if a student has just entered the system, and will avoid time-consuming online parameter estimations. In the future, we plan to further improve our discretization method.

4.1.3 Knowledge Component Extraction

As mentioned in Section 3.3, we adopted three different approaches for knowledge component extraction - the pure word-based method, a coarse-grained semantic-based method (chapter-based) and a fine-grained semantic-based method (latent topic-based). Table 1 compares them.

When applying the word-based approach for knowledge component extraction, we removed stop words¹, excluded non-letter symbols (e.g., brackets and punctuations) and performed stemming. There are 8,076 words identified as KCs. For the chapter-based approach, each book chapter

¹The stop word list is directly downloaded from Mallet (<http://mallet.cs.umass.edu/>).

Table 1: Knowledge component model comparison by number of KCs, whether it captures semantic relations, whether it provides enough fine-grained granularity for adaptation, and whether it captures transfer learning across chapters and books.

KC Model	#KCs	semantics	granularity	transfer
Words	8,076	no	yes	yes
Book Chapter	35	yes	no	no
Latent Topic	250	yes	yes	yes

is treated as a knowledge component directly. In total, our dataset includes 35 chapters from five textbooks. The latent topic-based approach identifies the knowledge components from the textbooks, based on latent semantic mining methods. This paper follows a previous study that works on a similar task [19]. Specifically, our LDA approach have 250 topics (this number was chosen based on our preliminary study). Since the initial document-topic probability is uniformly distributed, a probability that is higher than the cutoff $1/250$ indicates the relative importance of a topic in a document. Therefore, we chose $1/250$ as the cutoff probability, and represented a document using topics with an association probability bigger than $1/250$.

4.1.4 Cross-Validation and Evaluation

Following the conventions of traditional KT, we constructed KC-specific models, which means we fit parameters for each knowledge component, and evaluated our models by their ability to be generalized from trained students to test students. We conducted a 10-fold student stratified cross-validation where the data was first randomly split into 10 groups of students, and in each fold, 90% of students (from 9 groups) were used as a training set and the remaining 10% were used as the test set. Although trained models only capture KC characteristics, by the online update mechanism explained in Section 3.2.1, student models are able to maintain individual knowledge estimations for each student as their interaction evidence is accumulated.

We reported two popular prediction metrics used in evaluating student models, *root-mean-squared error (RMSE)* and *area under the receiver operating characteristic curve (AUC)*. For RMSE, a lower value is preferred; for AUC, a higher value is preferred. We computed the average RMSE or AUC across 10 folds and reported a 95% confidence interval, based on a t-distribution. In addition, we also compared the models’ potential impact on real-world adaptive tutoring by considering the semantics, granularity, and the ability to capture transfer of each KC model.

4.1.5 Tools

When building the KT-based and logistic regression-based models, we used two tools that allowed us to cope with the scale of the data. The first tool, which was used for KT, is **hmm-scalable**. It is a command line utility implemented in C/C++. It targets large datasets in the order of tens of millions of records and implements a suite of solver algorithms. **Hmm-scalable** can also fit KT models with per-student parameters (individualized KT) [43]. It has been extensively tested on different datasets and is freely available². We used the default values in the tool for our exper-

²<https://github.com/IEDMS/standard-bkt>

iments ($p(L_0)=0.5$, $p(T)=0.4$, $p(G)=0.2$, $p(S)=0.2$, maximum iteration=200, tolerance=0.01, solver using Gradient Descent). The other tool is **LIBLINEAR**, which we used for building logistic regression-based models. This tool also targets large datasets. **LIBLINEAR** was developed by a team at the National Taiwan University that won first prize at the 2010 Knowledge Discovery and Data Mining Cup. Although **LIBLINEAR** supports logistic regression, it does not support random factors. To fix that, we have developed a modified version of **LIBLINEAR** that implements random factors via grouped penalties. This modified version is also freely available³.

4.2 Experimental Results

In this section, we summarize our modeling results for KT-based and logistic regression-based models under different KC extraction methods.

4.2.1 Knowledge Tracing-based Student Model with Different Knowledge Component Models

Figure 2: KT-based student model with different KC models. The reported values are averaged across 10-fold cross-validation. 95% CI is also plotted for each method.

The summary of the results of KT models with varying KC models is given in Figure 2. Here, we see that in terms of both RMSE and AUC across all KC extraction methods, our proposed KT-based models consistently outperform the majority class baseline (which predicts *Read* for all observations). Even the simple chapter model performs quite well. This shows that students’ reading behaviors in our dataset can be reasonably well described under our hypothesis (assumptions) of the learning process in Section 3.1. Also, our framework of using a KT-based student model seems to be an effective approach to model learning from textbooks. Using the majority class model will suggest that students read all documents, risking the waste of a considerable amount of time that could be better used for learning new material.

Because the difference in predictive power among the three KC extraction methods is not statistically significant, it shows the robustness of our KT-based student model varying granularity and semantics of the underlying KC models. In terms of the absolute value, the KT model based on latent topics has a slight advantage over a word-based or chapter-based KC model. Despite the predictive accuracy being similar among the three models, we argue that the latent topic-based model might offer the highest benefit for personalization by examining semantics, granularity, and transfer learning (Table 1). Although the chapter-based model has clear semantic and pedagogic meaning, its knowledge modeling is too coarse-grained and it fails to capture transfer learning

³<https://github.com/IEDMS/liblinear-mixed-models>

across chapters. Although words (terms) can potentially capture transfer learning across chapters, the word-based model has weak semantic ground by treating words (terms) as knowledge components. In this context, the latent topic-based model provides a reasonable granularity level with its 250 topics operating on a sufficient semantic level, and it also maintains the ability to model transfer learning across chapters (and books). We foresee that combining latent topic modeling with textbook structure or expert knowledge can further increase the modeling capability. We plan to investigate this combination in future work.

4.2.2 Logistic Regression-based Student Models with Different Knowledge Component Models

Figure 3: Logistic regression-based student models with different KC models. The reported values are averaged across 10-fold cross-validation. 95% CI is also plotted for each method.

As shown in Figure 3, we also find that logistic regression-based student models all consistently outperform the majority class baseline model. This again provides evidence to support our hypothesis (assumptions) of the learning process in reading and the effectiveness of our framework to model this learning process.

There is no clear winner in terms of the KC units used (chapters, words, or latent topics); however, the PFA model based on simple chapters does stand out in terms of AUC. This suggests the potential benefit of using a textbook structure (as an exposition of domain expert knowledge) to extract KC models. However, as mentioned in Section 3.3, 4.2.1 and Table 1, chapter-based modeling provides a too coarse-grained knowledge estimation, and even if we replace it with a more fine-grained section or subsection based representation, it still fails to capture transfer learning across units (sections/subsections). This is not the case for word-based or latent topic based models, where words or latent topics can be shared across chapters or books so that knowledge (estimations) can be transferred. Combining this with the results from the KT-based model, we further strengthen our belief that a promising future direction is to combine latent topic modeling with a textbook structure or with expert knowledge.

4.2.3 Knowledge Tracing-based Student Model Vs. Logistic Regression-based Student Models

As we can see from Figure 4, the KT-based model on average significantly outperforms logistic regression-based models for both RMSE and AUC, as well as across all knowledge component modeling units. This is different from our expectation that logistic regression-based models should have a higher predictive power, since they naturally have the ability to handle multiple skills (as features) and do not suffer

Figure 4: KT-based vs. logistic regression-based student models varying KC models. The reported values are averaged across 10-fold cross-validation. 95% CI is also plotted for each method.

from the local optimum problem. We intend to further investigate the reason for this result. Given this result, the advantage of a KT-based model over logistic regression-based models stands out: it not only has significantly higher predictive performance, but also provides additional knowledge-estimation power.

So far, among all the model variants, the best performance is achieved by the latent topic-based and KT-based model, with the average RMSE as 0.46 and AUC as 0.67 across 10 folds. While these values are not the most ideal, they are very similar to the predictive performance reported in classic student model comparisons [16].

5. DISCUSSION AND CONCLUSION

This paper proposes a novel framework for dynamic modeling of student knowledge in textbook-based learning. We summarize the main contributions and limitations as follows.

We innovatively formulated the problem of modeling knowledge in reading as a reading time prediction problem. We argue that traditional quiz performance-based student knowledge modeling is not suitable in textbook-based learning environments, but instead, that using student reading activities allows us to obtain enough data and perform in-time knowledge estimation to empower in-time personalization. Our results showed that the proposed student models significantly outperform the majority baseline with a reasonable degree of predictive power. It provides evidence that our hypothesized learning process for reading modeling is plausible and that our modeling mechanism is feasible and effective. Moreover, the quality of models could now be evaluated using prediction, which is less time- and labor-consuming than traditional classroom study-based evaluation, and provides important insights for such studies. However, we are aware that such internal data-driven evaluation is still not enough, and we plan to further examine external validity as in [13] and to conduct classroom studies in the end to examine our model’s real world impact on personalization.

We built novel algorithms to model student knowledge learning from reading by reconstructing existing student models. The conventional student model, Knowledge Tracing (KT), is designed for tracing individual knowledge components (KCs) using quiz performance data. Moreover, it does not directly support situations where multiple KCs are associated with a single observed evidence. To address these limitations, we first mapped a time-prediction problem to traditional performance-prediction problems, and then we further constructed credit and blame assignment mechanisms for handling multiple KCs in the complex reading process. In our experiments, we found out that this pro-

posed KT-based model significantly outperforms not only the majority baseline model, but also state-of-the-art logistic regression-based models for student learning (which are expected to have higher predictive ability), with additional knowledge estimation power. This demonstrates that our blind assignment mechanism is simple but effective. We foresee that with improvement in the multiple KC handling mechanisms, applying the KT-based model is promising for modeling learning in reading.

Also, in order to readily apply existing classic student models, we made simplifications in terms of the hidden and observed variables (including the relation between them). We assume that students are diligent, devoted readers who will read whenever there is unlearned knowledge; that knowledge level is the only factor that affects reading time ignoring other factors (such as reading goals or strategies); and that reading time is the only observed behavior, ignoring rich behaviors such as mouse movements and specific focus on parts of a page. We also assume binary states for both knowledge levels and reading time. In particular, we follow classic student models’ KC-specific perspectives without explicitly considering individual student differences. We intend to investigate such issues in the future and improve our student modeling approach.

As a next step for implementing personalized guidance based on our dynamic knowledge model, we need to distinguish content pages with suitable difficulty levels that should be read from those that are too simple or too complicated that can be skimmed or skipped, given a student’s knowledge levels on KCs underlying the content. Our current model’s prediction of *Read* or *Skim* needs to incorporate this aspect of suitable difficulty levels in order to be translated into effective recommendation actions.

We also explored two automatic text-analysis approaches to extract knowledge component models. By using automatically extracted KCs, we address the open corpus problem in adaptive hypermedia research. It makes the process of KC extraction less time- and labor-consuming than in traditional expert-based domain knowledge engineering. We conducted extensive studies comparing different KC extractions varying semantics, granularity, and knowledge transferring ability. We found out that using the latent topic-based KC model provides the highest predictive ability than using simple book chapters or words (for a KT-based model). Although the advantage of using latent topics over others is not significant, we argue that its level of granularity, semantic relation modeling, and transferring ability will offer significant benefits to real-world personalization. We anticipate a promising future direction of combining this approach with textbook structure or domain expert knowledge.

Overall, our work could be considered as the first step to model dynamic knowledge in textbook-based learning. We believe that our framework is promising and that its application lies beyond textbook-based learning. This framework can be applied to a broader context of open-corpus personalized learning, empowering learners with the ability to access the right reading content at the right moment, despite the huge volume of online educational content.

Acknowledgement

This research was supported by the National Science Foundation Cyber-Human Systems (CHS) Program under Grant IIS-1525186.

6. REFERENCES

- [1] S. Bechhofer, C. Goble, L. Carr, W. Hall, S. Kampa, and D. De Roure. Cohse: Conceptual open hypermedia service. In *Annotation for the Semantic Web*, pages 193–210. IOS Press, 2003.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] P. Brusilovsky and J. Eklund. A study of user-model based link annotation in educational hypermedia. *Journal of Universal Computer Science*, 4(4):429–448, 1998.
- [4] P. Brusilovsky, C. Karagiannidis, and D. Sampson. Layered evaluation of adaptive learning systems. *International Journal of Continuing Engineering Education and Lifelong Learning*, 14(4/5):402 – 421, 2004.
- [5] P. Brusilovsky and J. Kim. Enhancing electronic books with spatial annotation and social navigation support. In *Proc. 5th Int. Conf. Universal Digital Library (ICUDL)*, 2009.
- [6] P. Brusilovsky and E. Millán. User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web*, pages 3–53. Springer-Verlag, 2007.
- [7] P. Brusilovsky and L. Pesin. Adaptive navigation support in educational hypermedia: An evaluation of the isis-tutor. *Journal of Computing and Information Technology*, 6(1):27–38, 1998.
- [8] P. Brusilovsky, S. Sosnovsky, and M. Yudelson. Addictive links: The motivational value of adaptive link annotation. *New Review of Hypermedia and Multimedia*, 15(1):97–118, 2009.
- [9] K. R. Canini, L. Shi, and T. L. Griffiths. Online inference of topics with latent dirichlet allocation. In *Int. Conf. Artificial Intelligence and Statistics*, pages 65–72, 2009.
- [10] R. P. Carver. *The causes of high and low reading achievement*. Routledge, 2000.
- [11] H. Cen, K. R. Koedinger, and B. Junker. Comparing two irt models for conjunctive skills. In *Proc. 10th Int. Conf. Intelligent Tutoring Systems*, pages 796–798, Berlin/Heidelberg, 2008. Springer-Verlag.
- [12] D. Chin. Empirical evaluations of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, 11(1-2):181–194, 2001.
- [13] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.
- [14] A. Davidovic, J. Warren, and E. Trichina. Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education*, 46(2):241–251, 2003.
- [15] R. Farzan and P. Brusilovsky. Social navigation support in e-learning: What are real footprints. In *IJCAI’05 Workshop on Intelligent Techniques for Web Personalization*, pages 49–56, 2005.
- [16] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proc. 10th Int. Conf. Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.
- [17] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proc. of the 7th Int. Conf. on Educational Data Mining*, pages 84–91, 2014.
- [18] J. Guerra, D. Parra, and P. Brusilovsky. Encouraging online student reading with social visualization. In *The 2nd Workshop on Intelligent Support for Learning in Groups at the 16th Conference on Artificial Intelligence in Education*, pages 47–50, 2013.
- [19] J. Guerra, S. Sosnovsky, and P. Brusilovsky. When one textbook is not enough: Linking multiple textbooks using probabilistic topic models. In *Scaling up Learning for Sustained Impact*, pages 125–138. Springer, 2013.
- [20] N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 12(4):325–350, 2001.
- [21] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 26(4):270–283, 2010.
- [22] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In *Proceedings of the 8th International Conference on Educational Data Mining*, pages 203–210, 2015.
- [23] M. A. Just and P. A. Carpenter. *The psychology of reading and language comprehension*. Allyn & Bacon, 1987.
- [24] A. Kavcic. Fuzzy user modeling for adaptation in educational hypermedia. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(4):439–449, 2004.
- [25] K. Koedinger, J. Stamper, E. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *Proceedings of the 16th International Conference on Artificial Intelligence in Education*, pages 412–430, 2013.
- [26] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.
- [27] B. Martin, A. Mitrovic, K. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283, 2011.
- [28] J. T. Mayes, M. R. Kibby, and H. Watson. Strathtutor: The development and evaluation of a learning-by-browsing on the macintosh. *Computers and Education*, 12(1):221–229, 1988.
- [29] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, 1:1–55, 1981.
- [30] K. O’Hara. Towards a typology of reading goals. Technical Report EPC-1996-107, Rank Xerox Research Centre Cambridge Laboratory, 1996.
- [31] K. A. Papanikolaou, M. Grigoriadou, H. Kornilakis, and G. D. Magoulas. Personalising the interaction in a web-based educational hypermedia system: the case of inspire. *User Modeling and User Adapted Interaction*, 13(3):213–267, 2003.

- [32] A. Paramythi and S. Weibelzahl. A decomposition model for the layered evaluation of interactive adaptive systems. In L. Ardissono, P. Brna, and A. Mitrovic, editors, *10th International User Modeling Conference*, volume 3538 of *Lecture Notes in Artificial Intelligence*, pages 438–442. Springer Verlag, 2005.
- [33] P. I. Pavlik Jr., H. Cen, and K. R. Koedinger. Performance factors analysis – a new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pages 531–538. IOS Press, 2009.
- [34] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.
- [35] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.
- [36] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
- [37] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
- [38] S. Sosnovsky and P. Brusilovsky. Evaluation of topic-based adaptation and student modeling in quizguide. *User Modeling and User-Adapted Interaction*, 25(4):371–424, 2015.
- [39] W. J. van der Linden and R. K. Hambleton. *Handbook of Modern Item Response Theory*. Springer Verlag, New York, NY, 1997.
- [40] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.
- [41] G. Weber and P. Brusilovsky. Elm-art: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, 12(4):351–384, 2001.
- [42] Y. Xu and J. Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best? In *Proceedings of the 5th International Conference on Educational Data Mining*, pages 41–48, Chania, Greece, 2012.
- [43] M. Yudelson, K. Koedinger, and G. Gordon. Individualized bayesian knowledge tracing models. In *Proceedings of 16th International Conference on Artificial Intelligence in Education (AIED 2013)*., pages 171–180, Berlin/Heidelberg, 2013. Springer-Verlag.
- [44] M. Yudelson, P. Pavlik, and K. Koedinger. User modeling – a notoriously black art. In *Proceedings of the 19th International Conference on User Modeling Adaptation and Personalization (UMAP 2011)*, pages 317–328, 2011.