Learning from Non-Assessed Resources: Deep Multi-Type Knowledge Tracing

Chunpai Wang, Siqian Zhao, Shaghayegh Sahebi
Department of Computer Science
University at Albany-SUNY
Albany, NY 12222
{cwang25,szhao2,ssahebi}@albany.edu

ABSTRACT

The state of the art knowledge tracing approaches mostly model student knowledge using their performance in assessed learning resource types, such as quizzes, assignments, and exercises, and ignore the non-assessed learning resources. However, many student activities are non-assessed, such as watching video lectures, participating in a discussion forum, and reading a section of a textbook, all of which potentially contributing to the students' knowledge growth. In this paper, we propose the first novel deep learning based knowledge tracing model (DMKT) that explicitly model student's knowledge transitions over both assessed and non-assessed learning activities. With DMKT we can discover the underlying latent concepts of each non-assessed and assessed learning material and better predict the student performance in future assessed learning resources. We compare our propose method with various state of the art knowledge tracing methods on four real-world datasets and show its effectiveness in predicting student performance, representing student knowledge, and discovering the underlying domain model.

Keywords

Knowledge Tracing, Multiple Learning Resource Types, Non-Assessed Learning Resources, Memory Augmented Neural Networks, Domain Knowledge Modeling, Student Knowledge Modeling

1. INTRODUCTION

As the education landscape shifts toward distance learning, the online learning systems advance in complexity and capacity. They can handle more students, evaluate students through different kinds of assessments, and offer various types of learning resources to them. In such systems, a student can study a reading section, take a quiz, watch a video lecture, and practice programming in an embedded development environment. As a result, students learn from heterogeneous types of activities in modern online learning systems, among which some can be assessed and some cannot.

Despite this heterogeneity in learning resource types, current student knowledge tracing models mostly focus on assessed learning resources, ignoring the non-assessed ones. In the assessed learning resource types, such as quizzes and assignments, students' performance can be evaluated given their answers and solutions. These kinds of learning resources provide a window to student knowledge through observing their performance. Conversely, in the non-assessed learning resources, such as readings and video lectures, such an observation does not exist. Hence, evaluating student knowledge and performance while interacting with these learning resources is a difficult task [4, 11, 10].

Indeed, because current knowledge tracing approaches do not model non-assessed learning resources, identifying their underlying concepts, finding the similarities between these learning resources, and in general domain knowledge modeling for such non-assessed learning materials is still a challenging problem. That is, many modern knowledge tracing models do not rely on a predefined domain knowledge model, such as a Q-matrix, and can identify the "latent concepts" that are being evaluated in problems, quizzes, or assignments [20, 23, 21, 7, 9]. This is particularly useful when annotating learning materials with their concepts is expensive or infeasible. However, discovering such latent concepts in non-assessed learning resources is an under-explored research area. Some recent works have aimed in identifying such latent concepts [19] and similarities [17] between assessed and non-assessed learning materials. However, their findings were according to static student performance, ignoring the sequential learning data of students.

In this paper, we argue that modeling non-assessed learning materials is essential and non-dispensable in tracing student knowledge. Students learn from all types of activities and ignoring a large portion of student activities is a missed opportunity in student knowledge tracing. Especially that previous research has shown that working with various learning activity types has considerable benefits for student learning [15, 2, 1, 12]. Hence, modeling both assessed and non-assessed learning activities should result in a more accurate estimation of student knowledge state and prediction of their performance on future assessed learning resources.

Accordingly, we propose Deep Multi-type Knowledge Tracing (DMKT) model, which not only traces student knowledge states over various learning activity types but also provides a feasible solution to discovering underlying patterns or concepts for both assessed and non-assessed learning resources. To this end, DMKT estimates student knowledge gain between every two consecutive assessed learning activities according to student performance on them. At the same time, it distributes this estimated knowledge gain among the in-between non-assessed learning activities and the latest assessed activity. We use an attention mechanism for this distribution. As a result, DMKT can model the underlying latent concepts for each of the assessed and non-assessed learning resources, evaluate student knowledge after interacting with these learning resources and predict student performance on the assessed ones.

We evaluate our proposed model on four real-world datasets, showing the significant effect of modeling various learning resource types on the task of student performance prediction. Also, we showcase the interpretability of DMKT by visualizing student knowledge while working with various learning resource types. Finally, we demonstrate the power of DMKT in discovering the learning resources' similarities and underlying latent concepts.

2. RELATED WORK

Student knowledge tracing aims to capture the student's knowledge state and knowledge state transition patterns, which could be further used for tasks like students' performance prediction, intelligent curriculum design, and interpretation and discovery of structure in student tasks.

Traditional knowledge tracing methods modeled knowledge transition on assessed learning resources using predefined domain knowledge models (concepts of learning resources). For example, Drasgow et al. proposed IRT that leverages the structured logistic regression to model student's dichotomous responses and estimates the student's ability, learning resource difficulty [8]. BKT uses binary variables for modeling whether student acquires a concept or not, and a Hidden Markov Model is used to update the probability that student answers a question correctly [6, 22]. However, since annotating a domain knowledge model can be expensive and time consuming, in many real-world scenarios, such predefined domain knowledge models are not be provided. To solve this problem, new approaches turn to investigate modeling student knowledge and domain knowledge at the same time. For example, Lan et al. utilized the matrix factorization to model the student knowledge and concept-question association, assuming the sparse association between concepts and questions [14]. As another example, Doan et al. model student learning with a tensor factorization in which the student knowledge is having an increasing trend using a rank-based constraint [7].

At the same time, in the past few years with the advance of deep neural networks, deep knowledge tracing methods have emerged. For example, DKT [18] utilizes LSTM to model students' knowledge transition over time. Recently, transformer-based neural networks have been successfully applied to model the different knowledge transitions of different students' historical interactions on learning resources [5, 9]. SAKT [16] uses the self-attention mechanism to model the interdependencies among interactions on the sequence. In [23], Zhang et al. proposed a Dynamic Key-Value Memory Networks based method (DKVMN), which integrates the memory augmented neural networks with the attention mechanism, to exploit the relationships between underlying concepts for better students' skill acquisition modeling. Yeung et al. extended DKVMN, by integrating the one-parameter logistic item response theory to provide better interpretability [21]. However, none of the deep knowledge tracing models have focused on modeling the non-assessed learning activities and tracing student knowledge on such activities.

Knowledge Tracing using Multiple Learning Resource Types. Previous approaches ignored the effect of learning activities on non-assessed learning resources, none of the methods mentioned above consider both assessed and non-assessed learning resources at the same time. However, in reality, students not only learn from practicing assessed learning resources (such as questions) but also learn by studying the non-assessed one, such as watching video lectures, reading textbooks, and discussing with others. One reason for not modeling the non-assessed activities is that reliable student performance observations are missing in these activities. This makes modeling the knowledge transition from these non-assessed learning activities difficult. To the best of our knowledge, the only existing work that models non-assessed learning activities along with the assessed ones is Multi-View Knowledge Model (MVKM) [25]. MVKM models multiple learning resources jointly using tensor factorization to capture latent students' features and latent learning resource concepts, assuming that latent concepts are shared by different learning resource types. However, this method can only capture the linear dependencies between variables, as the latent students' features and latent learning resource concepts are multiplied via linear matrix and tensor products. On the other hand, due to the large memory cost of tensor factorization, MVKM can not handle the datasets with very large student and learning resource numbers. Unlike MVKM, our proposed method in this paper considers the non-linear relationships between variables, and handles large datasets, while modeling student knowledge gain from multiple learning resource types (both assessed and non-assessed).

3. DEEP MULTI-TYPE KNOWLEDGE TRACING (DMKT)

3.1 Problem Formulation

A standard knowledge tracing (KT) problem is to predict student performance or response on an upcoming question, given the learner's performance records on previously solved questions. These records typically consist of a sequence of questions and responses at each discrete time step, denoted as a tuple (q_t^s, r_t^s) for student s at time step t. Since we only discuss how to predict future performance for a single student, we omit the superscript s in the following sections. Therefore, given students' past history records up to time $t-1$ as $\{(q_1, r_1), \cdots, (q_{t-1}, r_{t-1})\}$, the goal of KT is to predict their response r_t to question question q_t at the current time step t.

In this paper, we aim to incorporate students' non-assessed learning activities and model student knowledge transition over both assessed and non-assessed learning resources, such as solving quizzes, watching video lectures, viewing annotated examples or hints, and participating discussion forums. Therefore, given student's past historical responses to assessed learning materials as well as past history of non-assessed learning activities, we would like to estimate student knowledge and predict their performance in the next assessed learning resource. To do this, assuming L distinct non-assessed learning resources and Q distinct assessed ones, we represent students' historical records up to time $t-1$ as $\{(q_1, r_1), \mathcal{L}_1, (q_2, r_2), \mathcal{L}_2, \cdots, (q_{t-1}, r_{t-1}), \mathcal{L}_{t-1}\}$, in which $\mathcal{L}_t = \{l_t^1, l_t^2, \cdots, ...l_t^n\}$ at each time step t denotes the sequence of n non-assessed learning activities (e.g., watching video lectures) between the assessed activities (e.g., answering questions) q_t and q_{t+1}. Our goal is to predict student performance on assessed learning material q_t at each time step t, model student knowledge at and between time steps in interaction with q_t and all l_t^is, and discover the underlying latent concepts of assessed qs and non-assessed l^is.

3.2 The Base Model

We base our DMKT model upon a recent successful deep knowledge tracing model: DKVMN [23]. DKVMN is a special type of memory-augmented neural networks (MANN) for knowledge tracing which has one static key matrix to store the knowledge concepts and one dynamic value matrix to store students' updated mastery levels of those corresponding concepts. Assuming that there are N latent concepts $\{c^1, \cdots, c^N\}$ for each learning resource, and each latent concept can be represented by d_h-dimensional embeddings, similar to DKVMN, DMKT has the key matrix $\mathbf{M}^k$ of size $N \times d_h$ to store the N knowledge concepts. Similarly, the value matrix $\mathbf{M}_t^v$ of size $N \times d_h$ stores the student's mastery levels of each concept, at time step t.

However, DKVMN only supports updating knowledge states $\mathbf{M}_t^v$ on assessed learning materials, and lacks the ability to leverage the abundant of data other than student responses on assessed learning materials. To overcome this limitation, our proposed DMKT updates $\mathbf{M}_t^v$ with an additional internal component that employs the attention mechanism to process the non-assessed learning activities between any two assessed ones and use the updated $\mathbf{M}_t^v$ to predict student's performance on upcoming assessed learning resource. This component contains two functionalities, one is to update student knowledge state on non-assessed learning activities, and another is to summarize all activity contexts before an assessed activity to help accurate prediction of student performance.

One may think that a straightforward solution to integrate the non-assessed learning resources would be to consider them as student interaction features. However, since the non-assessed learning activities are not explicitly represented in such models, their contribution to student knowledge could be assessed. Also, such an approach cannot model student's knowledge transition between different non-assessed learning activities. In the following, we introduce our novel updating and summarizing functionalities that help DMKT to model all learning activity types. An overview of DMKT's architecture can be found in Figure 1[1].

3.3 Learning Resource Attention Weights

For the simplicity of illustration, let us assume that there is only one non-assessed learning activity, e.g., watching a video lecture, between solving two problems q_{t-1} and q_t, that is $\mathcal{L}_{t-1} = \{l_{t-1}\}$. DMKT assumes that student knowledge gets updated as the student interacts with l_{t-1} and q_t, weighted by their corresponding attention weights. So, in each step, DMKT uses attention weights from q_t and l_{t-1} to update the student knowledge in the concepts' embeddings, $\mathbf{M}_t^v$.

To compute the attention weights, DMKT first embeds all questions into an embedding matrix $\mathbf{A}^q \in \mathbb{R}^{Q \times d_h}$, and all video lectures in another embedding matrix $\mathbf{A}^l \in \mathbb{R}^{L \times d_h}$. At each time step, DMKT extracts the embedding vector of q_t ($\mathbf{k}_t \in \mathbb{R}^{d_h}$) from $\mathbf{A}^q$, as well as the embedding vector $\mathbf{k}_{t-1}^l \in \mathbb{R}^{d_h}$ of l_{t-1} from $\mathbf{A}^l$. Then, it uses these embedding vectors to query the key memory matrix $\mathbf{M}^k$ to obtain the attention weights $w_t^q(i)$ and $w_t^l(i)$ respectively as follows:

$$w_t^q(i) = \text{Softmax}\left(\mathbf{k}_t^{q\top} \mathbf{M}^k(i)\right) \tag{1}$$

$$w_{t-1}^l(i) = \text{Softmax}\left(\mathbf{k}_{t-1}^l{}^\top \mathbf{M}^k(i)\right) \tag{2}$$

The attention weight in $\mathbf{w}_t^q$ and $\mathbf{w}_{t-1}^l$ can be viewed as respectively the correlation between question q_t and lecture l_{t-1} with each of the N latent concepts. Notice that, $w_t^q(i)$ and $w_{t-1}^l(i)$ are the i-th element in the attention weight vectors $\mathbf{w}_t^q$ and $\mathbf{w}_{t-1}^l$ respectively, and for interpretability purposes the attention weights sum to one ($\sum_{i=1}^N w_t^q(i) = \sum_{i=1}^N w_{t-1}^l(i) = 1$).

3.4 Student Performance Prediction

At each time step t, DMKT aims to predict the student's performance on q_t. Since the predicted performance is a result of student knowledge that is gained by interacting with both problems and lectures, it is intuitive to aggregate these knowledge gains and predict the student performance accordingly. Remember that the memory value matrix $\mathbf{M}_t^v \in \mathbf{R}^{N \times d_h}$ is used to represent student's knowledge state on each concept embedding. So, to summarize the student's mastery level of question q_t and lecture l_{t-1} in the N concepts, we compute the weighted sum of all memory slots in the value matrix using attention weight vectors $\mathbf{w}_t^q$ and $\mathbf{w}_{t-1}^l$, respectively.

$$\mathbf{r}_t^q = \sum_{i=1}^N w_t^q(i) \mathbf{M}_t^v(i) \tag{3}$$

$$\mathbf{r}_{t-1}^l = \sum_{i=1}^N w_{t-1}^l(i) \mathbf{M}_t^v(i) \tag{4}$$

Then, we concatenate the latent knowledge states or mastery levels $\mathbf{r}_t^q$ and $\mathbf{r}_{t-1}^l$ on question q_t and lecture l_{t-1} with question embedding $\mathbf{k}_t^q$ as well as lecture embedding $\mathbf{k}_{t-1}^l$

[1] The source code is provided at: `https://github.com/persai-lab/EDM2021-DMKT`

vertically and pass them into a fully connected layer with a Tanh activation to obtain a summary vector $\mathbf{f}_t$

$$\mathbf{f}_t = \text{Tanh}\left(\mathbf{W}_1^\top \left[\mathbf{r}_t^q, \mathbf{r}_{t-1}^l, \mathbf{k}_t^q, \mathbf{k}_{t-1}^l\right] + \mathbf{b}_1\right) \tag{5}$$

where $[\cdot]$ denotes concatenation. This summary vector $\mathbf{f}_t$ contains a summary of all information, such as student ability and the relationship between question q_t and lecture l_{t-1}, to predict student response at time t accurately. Finally, the student's performance in query question q_t is calculated by passing the feature vector $\mathbf{f}_t$ through another fully connected layer with a Sigmoid activation as follows:

$$p_t = \text{Sigmoid}\left(\mathbf{W}_2^\top \mathbf{f}_t + \mathbf{b}_2\right) \tag{6}$$

3.5 Student Knowledge Update

DMKT tracks the student knowledge states by updating the memory value matrix $\mathbf{M}_t^v$ after each learning activity on q_t and l_t so as to predict student performance on q_{t+1} using the updated $\mathbf{M}_{t+1}^v$.

For assessed learning activities, we first retrieve an embedding vector of (q_t, r_t), denoted by $\mathbf{v}_t^q \in \mathbb{R}^{d_h}$, from a response embedding matrix $\mathbf{B}$ of size $2Q \times d_h$. This embedding $\mathbf{v}_t$ contains the information about how much student knowledge should be updated after working on question q_t with outcome r_t^q. We also use the *erase-followed-by-add* mechanism to update the memory value matrix, that is to erase the memory first using erase vector $\mathbf{e}_t^q \in [0,1]^{d_h}$ before adding new information with the add vector $\mathbf{a}_t^q \in \mathbb{R}^{d_h}$. This update of each value memory slot could be summarized as an erase step and an add step as follows:

Erase Step:

$$\begin{aligned} \mathbf{e}_t^q &= \text{Sigmoid}\left(\mathbf{E}^\top \mathbf{v}_t^q + \mathbf{b}_\mathbf{e}^q\right) \\ \tilde{\mathbf{M}}_t^v(i) &= \mathbf{M}_{t-1}^v(i) \otimes [\mathbf{1} - w_t^q(i)\mathbf{e}_t^q] \end{aligned} \tag{7}$$

Add Step:

$$\begin{aligned} \mathbf{a}_t^q &= \text{Tanh}\left(\mathbf{D}^\top \mathbf{v}_t^q + \mathbf{b}_\mathbf{a}^q\right)^T \\ \mathbf{M}_t^v(i) &= \tilde{\mathbf{M}}_{t-1}^v(i) + w_t^q(i)\mathbf{a}_t^q \end{aligned} \tag{8}$$

where $\mathbf{1}$ is a vector of all ones, and $\otimes$ represents the element-wise multiplication.

For each non-assessed activity, we follow a similar erase-followed-by-add steps in Eq.(7) and Eq.(8), except that we use $\mathbf{k}_t^l$ directly instead of a new response embedding.

Erase Step on Non-assessed Resources:

$$\begin{aligned} \mathbf{e}_t^l &= \text{Sigmoid}\left(\mathbf{H}^\top \mathbf{k}_t^l + \mathbf{b}_\mathbf{e}^l\right) \\ \tilde{\mathbf{M}}_t^v(i) &= \mathbf{M}_{t-1}^v(i) \otimes \left[\mathbf{1} - w_t^l(i)\mathbf{e}_t^l\right] \end{aligned} \tag{9}$$

Add Step on Non-assessed Resources:

$$\begin{aligned} \mathbf{a}_t^l &= \text{Tanh}\left(\mathbf{G}^\top \mathbf{k}_t^l + \mathbf{b}_\mathbf{a}^l\right)^T \\ \mathbf{M}_t^v(i) &= \tilde{\mathbf{M}}_{t-1}^v(i) + w_t^l(i)\mathbf{a}_t^l \end{aligned} \tag{10}$$

3.6 Network Architecture and Extension

The neural network architecture of DMKT is shown in Figure 1. For illustration simplicity, this figure assumes that there is only one non-assessed learning resource l_t between q_t and q_{t+1}. This architecture mainly contains two components: *read* component for making a prediction on input question q_t and *write* component for updating the value matrix after interacting with l_t and q_t.

When there are multiple non-assessed learning activities between q_t and q_{t+1}, that is $\mathcal{L}_t = \{l_t^1, \cdots, l_t^n\}$, we can simply extend the model by looping over each activity to generate $\mathbf{k}_t^{l^i}$ as well as $\mathbf{r}_t^{l^i}$ using equation (4) for $i \in \{1, \cdots, n\}$. When making predictions, we use $\sum_{i=1}^n \mathbf{k}_t^{l^i}$ to represent $\mathbf{k}_t^l$ and $\sum_{i=1}^n \mathbf{r}_t^{l^i}$ to represent $\mathbf{r}_t^l$ in the architecture. When updating the knowledge, the value matrix is updated sequentially over all activities as described in the previous subsection.

3.7 Training

All learnable parameters , i.e. $\mathbf{A}^q, \mathbf{A}^l, \mathbf{B}$, in the entire DMKT model are trained in end-to-end manner by minimizing the binary cross-entropy loss of all students' assessed responses, i.e.,

$$\ell_{BCE} = -\sum_t \left(o_t \log p_t + (1 - o_t) \log (1 - p_t)\right) \tag{11}$$

where o_t denotes the observation of correctness on assessed response at time t and p_t denotes the prediction of correctness of DMKT at time t.

3.8 Knowledge State Calculation

DMKT is capable of tracing and depicting knowledge concept mastery level for each student. A student's knowledge state before each assessed or non-assessed learning activity can be obtained in the *read* process using the following steps.

Assume that there are N dummy query questions q^is, each of them only using one concept, for the purpose of knowledge state calculation. Each of dummy questions can obtain a designed embedding $\mathbf{k}^i$ such that the correlation weight vector $\mathbf{w}^i$ is "one-hotted", that is $\mathbf{w}^i = [0, \cdots, w_i, \cdots, 0]$ where w_i of concept c^i is equal to 1. Then, we can use each of these one-hot correlation weight vectors to access value matrix state on each slot $\mathbf{M}_t^v(i)$ to obtain $\mathbf{r}_t^i$ for each concept c^i. In other words, $\mathbf{r}_t^i = \mathbf{M}_t^v(i)$ for q^i.

Then, we can predict the student knowledge purely based on $\mathbf{r}_t^i$ by masking the weight of the input content embedding in Eq. (5), which ends up as:

$$\mathbf{f}_t^i = \text{Tanh}\left(\left[\mathbf{W}_1^{r^i}, \mathbf{0}, \mathbf{0}, \mathbf{0}\right]^\top \left[\mathbf{r}_t^i, \mathbf{r}_{t-1}^l, \mathbf{k}_t^i, \mathbf{k}_{t-1}^l\right] + \mathbf{b}_1\right) \tag{12}$$

where $\mathbf{W}_1$ is split into four parts including $\mathbf{W}_1^{r^i}$, $\mathbf{W}_1^{k^i} = \mathbf{0}$, $\mathbf{W}_1^{r^l} = \mathbf{0}$, and $\mathbf{W}_1^{k^l} = \mathbf{0}$. Finally, a scalar value p^i is output as in Eq.(6) to be the predictive mastery level of concept c^i. We repeat this process N times with N numbers of one-hot correlation weight vectors to obtain student's knowledge state vector with size $1 \times N$ after each learning activity.

4. EXPERIMENTS

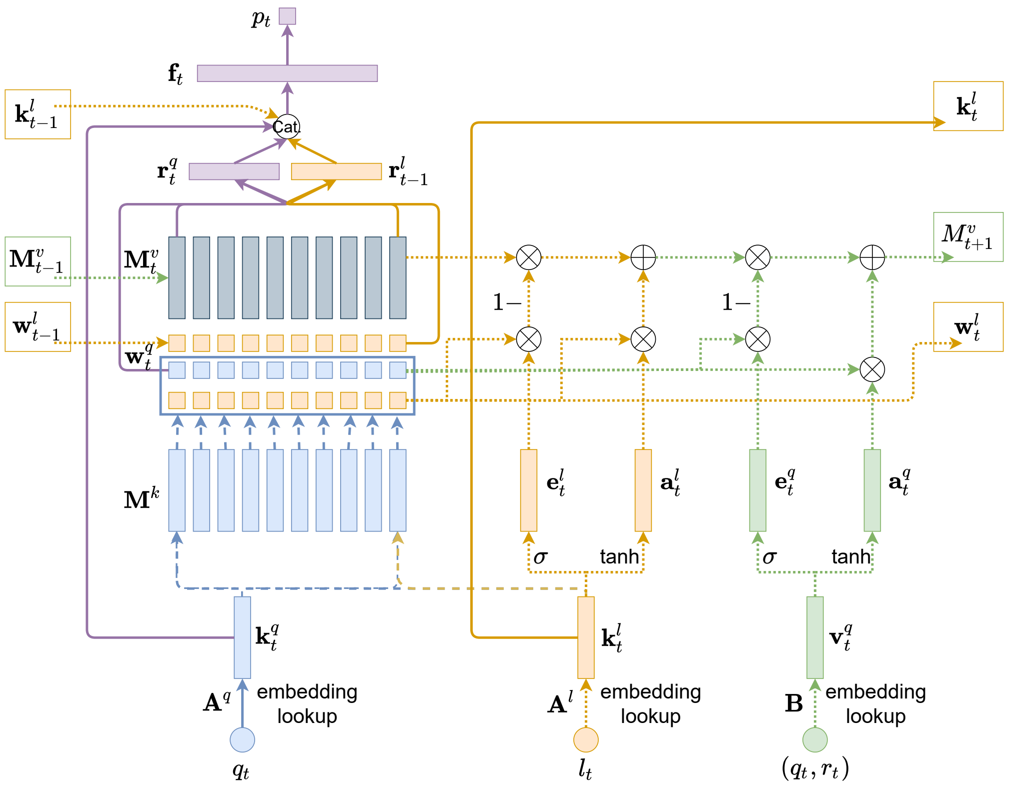


Figure 1: Neural Network Architecture of DMKT.

To evaluate our proposed model, we conduct three kinds of experiments. First, we compare it with state-of-the-art baselines in the student performance prediction task. Second, we analyze the discovered student knowledge transition patterns in terms of assessed and non-assessed learning activities. Last but not least, we validate the non-assessed learning resources' latent concepts discovered by the proposed method.

4.1 Datasets

We use three real-world datasets to evaluate the proposed model:

MORF[2] is an open online course dataset from Coursera [3]. In this course, students can watch lecture videos and work on problems. Each problem is a full complex course assignment. These video lectures and assignments are published in sequential order in this dataset, but students can have multiple attempts on each assignment and watch any video at any time. Students' scores are normalized into $[0, 1]$.

EdNet[3] is collected by Santa[4], a multi-platform AI tutoring service for students to prepare TOEIC English testing. We use the problem explanation documents as the non-assessed learning resources. There are $297,915$ user records in the full dataset, and we randomly extract $1,000$ users' records for experiments.

Junyi[5] is a dataset that comes from a Chinese e-learning website. Students work on problems from 8 math areas. Each problem has several hints, students can request hints when solving problems. We consider the problems as the assessed learning resources and the associated problem hints as the non-assessed learning resources. There are $25,925,922$ records in total from $247,606$ users in the full dataset. We extract two subsets of this full dataset for experiments. One is called Junyi2063, which contains 2063 users' records on 3760 questions and 1432 hints. A smaller dataset named Junyi1564, which consists of 1564 users' records on 142 questions and 116 hints, is extracted to serve the purpose of visualization on concept discovery results. The descriptive statistics of these four datasets are shown in the table 1.

4.2 Baseline Methods

In experiments of performance prediction, we compare with 13 baseline methods on the task of student performance prediction on assessed learning resources, including six state-of-the-art deep learning based knowledge tracing models, one existing tensor factorization based knowledge tracing model supporting multiple learning resource types, and seven extended deep learning based models utilizing non-assessed learning resources as additional input features. These methods are:

[2] https://educational-technology-collective.github.io/morf/

[3] https://github.com/riiid/ednet

[4] https://aitutorsanta.com/intro

[5] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1275

Table 1: Descriptive Statistics of 3 Datasets.

Dataset	Users	Questions	Question Records	Mean Question Responses	STD Question Responses	Correct Question Responses	Incorrect Question Responses	Non-gradable Materials	Non-gradable Records
MORF	686	10	12031	0.7763	0.2507	N/A	N/A	52	41980
EdNet	1000	11249	200931	N/A	N/A	118767	82184	8324	150821
Junyi1564	1564	142	120984	N/A	N/A	86654	34328	116	16389
Junyi2063	2063	3760	290754	N/A	N/A	193664	97090	1432	69050

- DKT [18]: is a pioneer deep learning based knowledge tracing method that uses LSTM to model students' knowledge transition over time.

- DKVMN [23]: is a variant of memory augmented neural networks that model the latent knowledge concept and dynamic student knowledge state over time.

- DeepIRT [21]: is an extension of DKVMN that integrates the one parameter logistic item response theory (1PL-IRT) to provide better interpretability, which could reduce the overfitting issue.

- SAKT [16]: is an attention-based method that leverages the self-attention mechanism to model the interdependencies among interactions on the sequence.

- SAINT [5]: is a transformer-based deep knowledge tracing method, two multi-head attention mechanisms are used to model exercise and response separately.

- AKT [9]: is a variant of transformer-based deep knowledge tracing method that using a monotonic attention mechanism to model the different knowledge transition of students' each historical performance on questions.

In addition to those baselines that support assessed learning materials, we also compare our method with some baselines that either can leverage additional students' non-assessed learning activities by design, or we modify them to consider such non-assessed activities as features of the assessed ones and predict students' future performance. These methods are:

- MLP-M: is a simple multi-layer perceptron that could take query question ID, user ID, and user's 3 past historical records on current query question, as well as 3 most recent non-assessed learning activities as input, and output a probability of user's mastery level on query question.

- DKT-M [24]: is an enhanced DKT model that could incorporate additional question features by concatenating the feature embeddings with exercise response embedding as the input of vanilla DKT.

- SAINT-M [5]: is a variant of SAINT that summing over all embeddings of non-gradable activities along with position encoding as the input of SAINT.

- MVKM [25]: is state of the art method on modeling student knowledge transition over multiple learning resource types based on multiview tensor factorization.

Inspired by the DKT-M [24], we apply the same strategy to DKVMN to incorporate additional non-assessed learning activities as features to end up with method DKVMN-M. Also, inspired by the way of SAINT-M [5] to incorporate rich features into transformer-based model, we apply same strategy as described in the paper into SAKT and AKT to incorporate additional non-assessed learning activities as response features that ends up with baseline methods SAKT-M and AKT-M, respectively.

4.3 Implementation Details

For binary response datasets, including EdNet and Junyi datasets, we convert the response tuple (q_t, r_t) into a single value $z = q_t + r_t \times Q \in \{1, \cdots 2Q\}$ as the lookup key of embedding layer. For numerical response MORF dataset, we feed the tuple (q_t, r_t) into a linear layer to get the embedding. For the question q_t and non-assessed learning resource l_t, we feed their ID into the embedding layers.

For evaluation purpose, we perform the 5-fold user stratified cross-validation for all models and all datasets. Hence, for each fold, 60% users are used as the training set, 20% are validation set, and the rest 20% as test set. For each fold and every method, we use the validation set to tune the hyper-parameters and record the optimal training loss as the condition of early stopping.

We utilize the Gaussian distribution with 0 mean and 0.2 standard deviation to initialize the values of $\mathbf{M}^k$ and $\mathbf{M}^v_0$. We learn the model using the Adam optimization with a learning rate of 0.01 and reduce the learning rate by half once the training loss increases, with the minimal learning $1e$-5 for all methods in 200 max epochs. We also utilize the norm clipping threshold to 50.0 to avoid gradient exploding for all methods. In addition, we follow the general processing steps for knowledge tracing that truncate long sequence and pad short sequence with 0s. The length of sequence is considered as a hyper-parameter of all models which needs to tune. In addition, we also tune the max sequence length of non-assessed learning activities between two assessed learning activities $\mathcal{L}_t$. If the length of non-assessed learning activities is over the maximum size, then we take the most recent ones. Similarly, if the length is less than the required sequence length, we pad with 0s. The table 2 shows the best hyper-parameters of our DMKT on 4 datasets.

We implement the models using PyTorch on a computer with a single NVIDIA Tesla-K80 GPU. For DKT and DKT-M, our implementation is different from the original paper [18], and we follow the same idea suggested by [23] that use norm clipping and early stopping, which could ease the gradient exploding as well as overfitting issues of LSTM. Xavier

Table 2: Hyperparameters of DMKT

Dataset	d_h	N	seq. len.	$\lvert\mathcal{L}_t\rvert$
MORF	128	8	50	8
EdNet	128	8	50	2
Junyi1564	256	8	50	2
Junyi2063	256	32	50	2

initialization is also used to initialize the parameters in DKT and DKT-M. All the baseline methods are implemented in PyTorch and tested to achieve similar performance as reported in the original paper except the SAINT ans SAINT-M. For SAINT, we borrow the implementation from github[6] and extend it to the SAINT-M, since the authors did not release the code.

4.4 Student Performance Prediction

The results of predicting students' performance in the assessed learning resources, including their 95-percentile confidence intervals, are shown in Table 3. The RMSE is measured to evaluate the prediction performance on MORF dataset due to numerical user responses, and the AUC is measured on EdNet and Junyi datasets. A low RMSE score indicates a high prediction performance. An AUC of 0.5 represents the model's performance is equivalent to a random guess model. A high AUC score accounts for a high prediction performance. As you can see, our proposed method, DMKT, achieves the best performance over all baseline methods on all four datasets. This shows that explicitly modeling non-assessed learning materials, along with the assessed ones, is essential in capturing the variations in student performance data.

We can also see that by simply incorporating the non-assessed activities between two assessed activities as additional input features (the "-M" models) the prediction performance is improved in some methods, such as AKT-M on MORF, EdNet, and Junyi datasets. However, unlike attention-based methods which could learn interaction correlation in a long sequence, this kind of simple integration strategy does not improve and may harm the prediction performance in other methods, such as in DKT-M and DKVMN-M, which tend to summarize past historical records as context embeddings. The reason we believe is this trivial integration of non-assessed activities not only loses a large amount of sequential information to model student knowledge transition over time, but also could introduce more noisiness on the data. We conclude that *simply adding the non-assessed learning activities as features, without modeling them explicitly is not enough and may even harm the prediction performance in some models.*

SAINT and SAINT-M have transformer based architecture, which can stack multiple encoders and decoders. However, in our EdNet dataset that contains only $1,000$ users with $200,931$ records on $11,249$ questions, and without additional constraints or regularization as proposed in AKT (another transformer based model), SAINT and SAINT-M can easily overfit the data. MVKM is the only existing baseline method that can explicitly model multiple learning resource types. We can see that it can outperform the deep knowledge tracing methods that uses non-assessed learning materials as features in MORF, which is a mid-size datasets. However, it cannot efficiently run in the larger datasets as the memory usage and linear time complexity over number of interaction records in MVKM limits its applicability on large datasets, such as EdNet and Junyi. Therefore, due to long running time on EdNet and Junyi datasets, we only report its performance in the MORF dataset.

It is worth noting that when our model is fed with assessed learning resources only, it will be equivalent to DKVMN. However, as presented in the table, our proposed model DMKT achieves a better performance over DKVMN as well as DKVMN-M, because DMKT explicitly models the student knowledge transition on non-assessed learning activities, which provides a more accurate encoded information to make the predictions accurately.

4.5 Student Knowledge State Visualization

To see how intractable the discovered student knowledge states are, we visualize the students' knowledge states. Basically, knowledge state visualization shows student's knowledge mastery level on each concept before each attempt on a non-assessed or an assessed learning activity. This provides a useful tool to monitor student knowledge coverage over different concepts and helps instructors to analyze the student's lacking concepts so as to provide tailored instructions for each student. To visualize student knowledge states, we follow the steps in section 3.8 to calculate knowledge state values over all concepts across the student sequence for each student. We show visualization of one example student's knowledge states in the MORF dataset in figure 2. As you can see in the figure, the top x-ticks are labeled with student learning activities. Assessed learning materials (assignments) start with A and non-assessed ones (lecture videos) are annotated by the week they are scheduled and the sequence of video lecture within the week. For example W4V0 means the student has watched week 4 video lecture 0 and A1B denotes the Assignment-1B in week 1. The bottom x-ticks are labeled by either student performance (grade) in the assessed learning materials, or an icon indicating the non-assessed learning resource type. Each row represents one latent concept. In the figure, this student starts with a randomly initialized value memory matrix $\mathbf{M}_0^t$ at time step 0 before working on $A1B$. After finishing the A1B, student's knowledge is updated and increased a little on concept 3 and 6 before working on A3. Student's knowledge grows gradually by working on assignments $A3$ and watching video lectures in week 4. However, student's knowledge drops a little before working on assignment $A4$ and it explains the reason why that student only receives a score 0.3 at the first attempt. Student's knowledge on all concepts grow by working on the assignments until the student started watching video lecture $W6V1$. We can see a slight drop in student's knowledge of some of the concepts (e.g., 7) and increase in other concepts (e.g., 1) while they are watching these videos. One potential reason for the decrease on concept 7 could be the lack of practice with assignments. Watching video lectures indeed improve student knowledge on concept 1 and 2. Another reason for the drop in concept 7 could be related to the student's problem solving ability which results

[6] `https://github.com/Shivanandmn/Knowledge-Tracing-SAINT`

Table 3: Student Performance Prediction Results on 3 Real-World Datasets. Root Mean Square Error (RMSE) and Area Under Curve (AUC) are used to evaluate performance on datasets with numerical feedback and binary feedback, respectively. The average performance over 5 folds as well as 95% confidence interval are reported.

	MORF	EdNet	Junyi1564	Junyi2063
Methods	RMSE	AUC	AUC	AUC
DKT	0.1870 ± 0.0191	0.6393 ± 0.0137	0.8877 ± 0.0050	0.8635 ± 0.0059
DKVMN	0.2042 ± 0.0136	0.6296 ± 0.0104	0.8843 ± 0.0065	0.8558 ± 0.0068
DeepIRT	0.1946 ± 0.0080	0.6290 ± 0.0105	0.8749 ± 0.0053	0.8498 ± 0.0069
SAKT	0.2113 ± 0.0275	0.6334 ± 0.0125	0.8623 ± 0.0047	0.8053 ± 0.0075
SAINT	0.2019 ± 0.0077	0.5205 ± 0.0064	0.8454 ± 0.0096	0.7951 ± 0.0119
AKT	0.2420 ± 0.0155	0.6393 ± 0.0104	0.8311 ± 0.0102	0.8093 ± 0.0091
MVKM	0.1936 ± 0.0096	–	–	–
MLP-M	0.2433 ± 0.0350	0.6102 ± 0.0088	0.7055 ± 0.0191	0.7290 ± 0.0150
DKT-M	0.1927 ± 0.0194	0.6372 ± 0.0120	0.8885 ± 0.0048	0.8652 ± 0.0069
DKVMN-M	0.2251 ± 0.0128	0.6343 ± 0.0074	0.8948 ± 0.0054	0.8513 ± 0.0059
SAKT-M	0.2084 ± 0.0272	0.6323 ± 0.0109	0.8305 ± 0.0071	0.7911 ± 0.0107
SAINT-M	0.1977 ± 0.0055	0.5491 ± 0.0068	0.8454 ± 0.0096	0.7741 ± 0.0139
AKT-M	0.2239 ± 0.0151	0.6404 ± 0.0067	0.8296 ± 0.0093	0.8099 ± 0.0098
DMKT	$\mathbf{0.1369 \pm 0.0195}$	$\mathbf{0.6675 \pm 0.0082}$	$\mathbf{0.9440 \pm 0.0061}$	$\mathbf{0.8714 \pm 0.0069}$

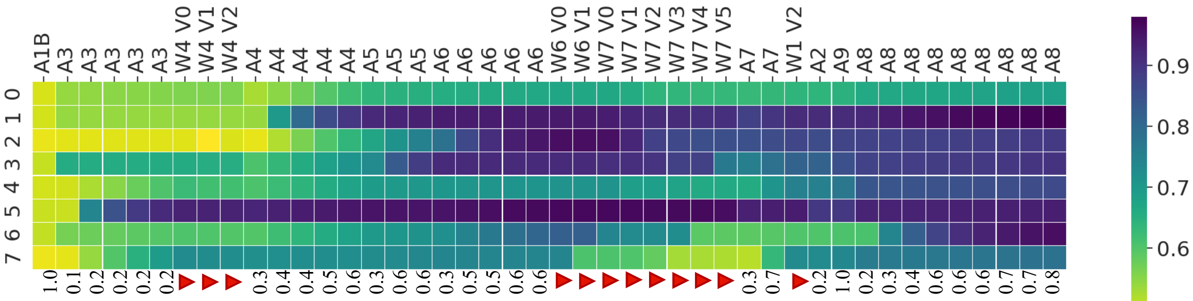


Figure 2: An Example of Student Knowledge State Visualization on MORF Dataset.

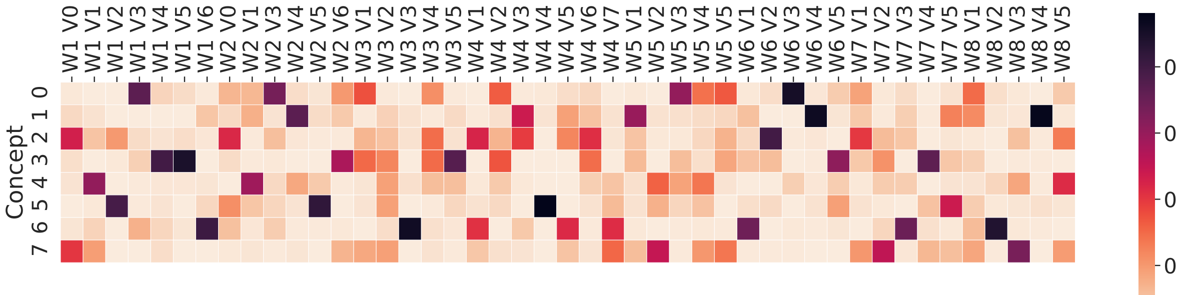
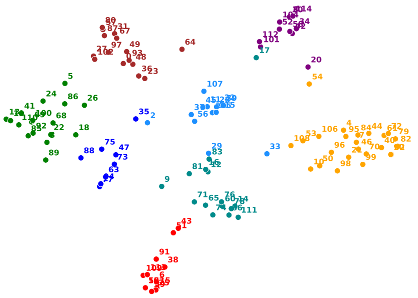


Figure 3: Concepts Matrix of Video Lectures in MORF Dataset.

in their first attempt on Assignment $A7$ to have a score of 0.3. Once the student's first attempt on $A7$ is done, this student quickly masters concept 7 again and their knowledge on all concepts continues to grow along different activities. In this example, it seems *the assessed learning material improves student knowledge more than watching video lectures, which is inline with the previous literature* [10, 13]. Another observation is that this student skips watching video lectures in weeks 1, 2, and 3 before working on assignment $A3$. Similarly, they did not watch videos in week 5 and 6 before trying $A5$ and $A6$. This may explain that this student is not interested in watching video lectures and may not be fully present during watching video lectures which results in tiny knowledge growth over watching them.

4.6 Concept Discovery

In addition to tracing student knowledge over various types of learning activities, DMKT can provide a feasible solution to discovering underlying patterns or concepts for both assessed and non-assessed learning resources. In other words, the correlation weights $\mathbf{w}$ and $\mathbf{w}^l$, can be interpreted as the importance of latent concepts in each assessed, and non-assessed learning activity respectively. Meaning that, since the key matrix $\mathbf{M}^k$ is used to model the knowledge concepts on the full course, the correlation weight between the learning resources and the concepts implies the strength of



6 telling_time--analog
15 subtraction_1--0
25 subtraction_2--subtraction-2
38 adding_and_subtracting_negative_numbers--MINUS
43 limits_1--3
51 same_side_interior_angles--0
55 reading_tables_1--0
59 measuring_segments--0
91 measuring_and_comparing_angles--compare_4
103 logical_arguments_deductive_reasoning--detachment
109 independent_probability--0
113 conditional_statements_and_truth_value--deduce

0 representing_numbers--0
23 number_line_2--where-is-point
27 adding_negative_numbers--PLUS
31 axis_of_symmetry--0
36 congruency_postulates--0
48 evaluating_expressions_1--0
49 angle_types--obtuse
57 reading_tables_2--0
64 congruent_segments--0
67 points_lines_and_planes--line-name
80 reading_bar_charts_2--reading
87 areas_of_kites--kite
93 measuring_and_comparing_angles--measure_2
97 quotient_rule--0
102 logical_arguments_deductive_reasoning--syllogism

8 comparing_absolute_values--0
20 recognizing_rays_lines_and_line_segments--line-segment
30 telling_time_2--set-hands
34 multiplication_2--0
42 multiplying_decimals--0
52 counting_1--fences
58 reading_pictographs_1--read-chart
101 plotting_the_line_of_best_fit--linear
104 reading_bar_charts_3--midrange
112 inverse_trig_functions--0
114 variance--population

4 telling_time_0.5--analog
7 divisibility_intuition--0
10 conditional_statements_2--category
21 addition_2--0
40 prime_numbers--0
44 limits_1--2
46 limits_1--1
50 angle_types--right
53 counting_1--page-numbers
54 counting_1--slices
61 square_roots--square
62 similar_triangles_1--similar
70 converse_inverse_and_contrapositive--equations-and-equations
72 vertical_angles--0
79 reading_line_charts_1--0
82 creating_bar_charts_1--0
84 linear_equations_2--0
90 midpoint_of_a_segment--0
95 converse_inverse_and_contrapositive--elephants-and-peanuts
96 converse_inverse_and_contrapositive--vertical-and-congruent-angles
98 surface_area_of_rectangular_boxes--int-box-T
99 conditional_statements_2--event
106 limits_2--n-lt-d
108 derivative_intuition--polynomial

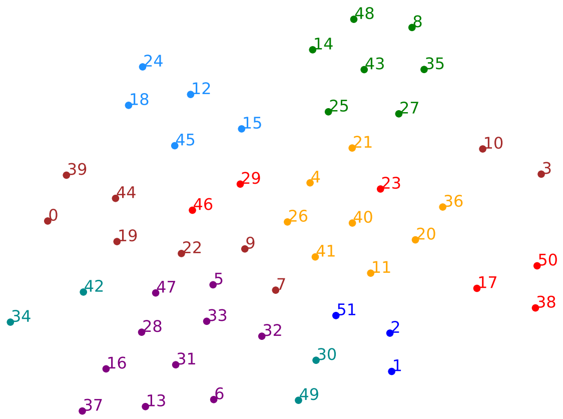
9 converse_inverse_and_contrapositive--dessert-and-pie
12 exponents_1--0
14 multiplication_1--0
16 dividing_decimals_0.5--decimal
17 pythagorean_theorem_1--find-a-leg
60 triangle_types--knownSides
65 measuring_angles--0
66 points_lines_and_planes--name-plane
71 multiplying_and_dividing_scientific_notation--divide
74 converse_inverse_and_contrapositive--rain-and-soccer-practice
76 reading_stem_and_leaf_plots--greater-than
78 reading_bar_charts_1--0
81 reading_bar_charts_2--most-improved
83 linear_equations_1--0
111 triangle_angles_sum--scalene

1 number_line--0
35 expressing_ratios_as_fractions--0
47 limits_2--a-o-xmk
63 conditional_statements_2--adjective
73 complementary_angles--find-hypotenuse
75 converse_inverse_and_contrapositive--sun-and-hat
77 reading_stem_and_leaf_plots--less-than
88 lhopitals_rule--polynomial
94 radius_and_diameter--circle-r-to-d

2 addition_1--0
11 area_of_parallelograms--par-bh-to-K
28 combining_like_terms_1--none
29 exponents_2--0
32 perimeter_of_squares_and_rectangles--rectangle-lw-to-P
33 decimals_on_the_number_line_2--0
37 ordering_negative_numbers--0
39 adding_and_subtracting_negative_numbers--PLUS
45 limits_1--0
56 exploring_mean_and_median--0
105 limits_2--n-gt-d
107 derivatives_1--quadratic
115 sides_and_angles_of_simple_shapes--polygonshape_vertex

3 multiplication_0.5--0
5 absolute_value--0
13 division_0.5--0
18 recognizing_rays_lines_and_line_segments--line
19 recognizing_rays_lines_and_line_segments--ray
22 number_line_2--what-is-point
24 number_line_3--0
26 triangle_types--knownAngles
41 composite_numbers--0
68 angle_types--acute
69 combining_like_terms_2--none
85 radians_to_degrees--1
86 degrees_to_radians--degrees-to-radians
89 special_derivatives--0
92 measuring_and_comparing_angles--measure_w_90
100 counting_2--every-n
110 creating_box_and_whisker_plots--0

Figure 4: Cluster Graph of Non-gradable Learning Materials (Hints) in Junyi1564 Dataset Using t-SNE. The question name corresponding to each hint is shown in the right table. (Best viewed in color)



8 W3 V3: Feature Engineering
14 W1 V6: Case Study - San Pedro
25 W4 V5: Advanced BKT
27 W4 V7: Other Structures
35 W6 V1: Learning Curves
43 W7 V3: Advanced Clustering Algorithms
48 W8 V2: Discovery with Models - Case Study

0 W1 V4: Classifiers Part 2 - RapidMiner 5.3
3 W1 V5: Classifiers Part 3
7 W3 V2: Data Synchronization and Grain Sizes
9 W3 V4: Automated Feature Generation
10 W3 V5: Knowledge Engineering
19 W2 V6: Types of validity
22 W4 V2: Bayesian Knowledge Tracing
39 W6 V5: Other Awesome EDM Visualizations
44 W7 V4: Clustering Examples

5 W3 V1: Ground Truth for Behavior Detection
6 W3 V0: Intro Video
13 W1 V3: Classifiers Part 1
16 W2 V3: Diagnostic Metrics, part 2
28 W5 V0: Intro Video
31 W5 V3: Association Rule Mining
32 W5 V4: Sequential Pattern Mining
33 W5 V5: Network Analysis
37 W6 V3: Scatter Plots
47 W8 V1: Discovery with Models

12 W1 V2: Regressors
15 W2 V0: Intro Video
18 W2 V5: Detector Confidence, part 4
24 W4 V4: Item Response Theory
45 W7 V5: Factor Analysis

30 W5 V2: Casual Mining
34 W6 V0: Intro Video
42 W7 V2: Validation and Selection of K
49 W8 V3: Text Mining

4 W2 V2: Diagnostic Metrics, part 1
11 W1 V0 Introduction
20 W4 V0: Intro Video
21 W4 V1: Knowledge Inference
26 W4 V6: Q Matrix
36 W6 V2: Moment by Moment Learning Graphs
40 W7 V0: Intro Video
41 W7 V1: Clustering

17 W2 V4: Detector Confidence, part 3
23 W4 V3: Performance Factors Analysis
29 W5 V1: Correlation Mining
38 W6 V4: State Space Diagrams
46 W8 V0: Intro Video
50 W8 V4: Hidden Markov Models

1 W2 V1: Detector Confidence
2 W1 V1: Big Data in Education
51 W8 V5: Conclusions and Further Directions

Figure 5: Cluster Graph of Video Lectures using t-SNE and Titles of Video Lecture of MORF Dataset. Lectures under the same concept are labeled in the same color in the left picture and also are put in the same block in the right table. (Best viewed in color)

their inner relationship. *Not only we can use the correlation weight as latent concepts, we can also use them to find similar learning resources by clustering them over these correlation weights.*

For example, in Figure 3, we visualize the importance of each concept in each of the MORF dataset video lectures. The X-axis ticks show the video lecture weeks and numbers and the Y-axis shows the latent concepts. *As we can see, the concept matrix is relatively sparse, showing that most video lectures strongly belong to 2-3 concepts, while they do have a soft memberships in other concepts too. Many video lectures in the same week have similar concept structures.* For example videos 3, 4, and 5 of week 5 all have a strong representation of concept 0 and videos 0, 1, and 2 of week 1 all are having high correlation weights with concept 2. Given that the course schedule is designed by the instructor, such similarities between the concepts in videos of the same week are expected. *Another interesting observation is the strong appearance of some concepts in videos of different weeks.* For example, concept 1 can be seen in both video 4 of week 8 and video 4 of week 6. This shows that these two video lectures share some similarities that are not represented in class schedule. Looking at the video titles from this course (right-hand side of Figure 5) we can see that the video titles are *State Space Diagrams* and *Hidden Markov Models*, respectively, which are two very closely-related topics To better understand such similarities, we look at grouping of videos according to their discovered concepts in the following.

To this end, we follow the clustering procedures as in [23] to group the learning materials according to the discovered latent concepts. At the same time, we compare these groupings by looking at the problem name associated with each hints and lecture titles for Junyi1564 and MORF datasets in Figures 4 and 5, respectively. To do the clustering, we first assign each learning resource with the concept ID that contains the largest correlation weight as the cluster label. Since there are 8 concepts in total, it results in 8 clusters. Then, we use t-SNE to visualize the clusters, which are shown in the left sides of Figures 4 and 5 for Junyi1564 and MORF datasets, respectively.

As we can see, the resulting t-SNE clusters are more distinct in the Junyi1564 dataset compared to MORF. *In other words, most of clusters in Junyi1564 dataset could be easily separated and distinguished. This implies that the discovered concept matrix of the Junyi1564 dataset is more sparse than the one from the MORF dataset, leading to more outstanding clusters than in MORF, as shown in Figure 4*. Indeed we have seen from Figure 3 that each video lecture in MORF is associated with two to three latent concepts rather than having only one distinct concept. This finding matches these datasets' properties: in the MORF dataset, each assessed learning material is a full complex course problem set which is assigned to students every week, and each non-assessed learning resource is a video lecture that covers multiple knowledge concepts. On the contrary, the assessed learning materials in Junyi1564 dataset are simple math problems, with close-to atomic concept coverage, and the non-assessed resources are hints associated with these problems.

As another result of this clustering, and similar to our findings in Figure 3, we can see that *the more similar or related non-assessed learning materials are clustered together*. For example, in Figure 5, video lectures from week 5 are clustered together, showcasing the similarity between latent concepts in video lectures that are scheduled to be presented together in week 5 of the course. Additionally, video lectures that are conceptually similar to each other can be found grouped together. For example, video lectures from week 6 (*V4 - State Space Diagrams*) and week 8 (*V4 - Hidden Markov Models*), from week 1 (*V2 - Regressors*) and week 7 (*V5 - Factor Analysis*), and from week 1 (V6 - Case Study - San Pedro) and week 8 (*V2 - Case Study - Discovery with models*) are grouped together which are conceptually similar.

These findings are also in accordance with the previous findings in the literature on the MORF dataset [25] and show that DMKT can efficiently discover the underlying concepts presented in the non-assessed learning materials, even though student performance on them is not observable.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed DMKT, the first deep learning based knowledge tracing model that can model and trace student knowledge in both assessed and non-assessed learning resources, find the underlying connects and similarities between learning resources, and predict student performance in the assessed ones. We evaluated DMKT extensively, on four real world datasets and demonstrated that because of its explicit modeling of non-assessed learning materials, its ability in representing non-linear relationships and its capacity in handling larger amounts of data, it outperforms all the baselines, in accurately predicting student performance. We further showcased DMKT's ability in meaningfully tracing student knowledge over assessed and non-assessed learning resources, and the potential effect that each of them can have on student knowledge. In our particular example, we showed that solving problems is a more effective way to learn for our selected student, compared to watching video lectures. Finally, we presented that DMKT can find interpretable latent concepts of non-assessed learning materials, that can be used to group them into meaningful clusters. In the future work, we would like to explore this model on various of learning activities to learn hidden patterns on different learning resources so as to provide tailored learning resource recommendations.

Acknowledgements. This paper is based upon work supported by the National Science Foundation under Grant No. 1755910.

6. REFERENCES

[1] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan. Mining videos from the web for electronic textbooks. In *Proceedings of the 12th International Conference on Formal Concept Analysis*, pages 219–234, Berlin, Heidelberg, 2014. Springer.

[2] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1847–1856, New York, NY, USA, 2011. ACM.

[3] J. M. L. Andres, R. S. Baker, G. Siemens, D. Gašević, and C. A. Spann. Replicating 21 findings on student success in online learning. *Technology, Instruction, Cognition, and Learning*, pages 313–333, 2016.

[4] J. E. Beck, K.-m. Chang, J. Mostow, and A. Corbett. Does help help? introducing the bayesian evaluation and assessment methodology. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, pages 383–394, Berlin, Heidelberg, 2008. Springer.

[5] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the 7th ACM Conference on Learning at Scale*, pages 341–344, New York, NY, USA, 2020. ACM.

[6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

[7] T.-N. Doan and S. Sahebi. Rank-based tensor factorization for student performance prediction. In *Proceedings of the 12th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2019.

[8] F. Drasgow and C. L. Hulin. Item response theory. *Handbook of Industrial and Organizational Psychology*, pages 577–636, 1990.

[9] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2330–2339, New York, NY, USA, 2020. ACM.

[10] R. Hosseini, T. Sirkiä, J. Guerra, P. Brusilovsky, and L. Malmi. Animated examples as practice content in a java programming course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 540–545, New York, NY, USA, 2016. ACM.

[11] Y. Huang, J. P. González-Brenes, and P. Brusilovsky. Challenges of using observational data to determine the importance of example usage. In *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, pages 633–637, Berlin, Heidelberg, 2015. Springer.

[12] H. Khosravi, G. Demartini, S. Sadiq, and D. Gasevic. Charting the design and analytics agenda of learnersourcing systems. In *Proceedings of the 11th International Learning Analytics and Knowledge Conference*, page 32–42, New York, NY, USA, 2021. ACM.

[13] K. R. Koedinger, J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a mooc. In *Proceedings of the 2nd ACM Conference on Learning at Scale*, page 111–120, New York, NY, USA, 2015. ACM.

[14] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *The Journal of Machine Learning Research*, 15(1):1959–2008, 2014.

[15] A. S. Najar, A. Mitrovic, and B. M. McLaren. Adaptive support versus alternating worked examples and tutored problems: Which leads to better learning? In *Proceedings of the 22nd International Conference on User Modeling, Adaptation, and Personalization*, pages 171–182, Berlin, Heidelberg, 2014. Springer.

[16] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 384–389. International Educational Data Mining Society, 2019.

[17] R. Pelánek. Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*, 13(2):354–366, 2019.

[18] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, page 505–513, Cambridge, MA, USA, 2015. MIT Press.

[19] S. Sahebi and P. Brusilovsky. Student performance prediction by discovering inter-activity relations. *International Educational Data Mining Society*, 2018.

[20] S. Sahebi, Y.-R. Lin, and P. Brusilovsky. Tensor factorization for student modeling and performance prediction in unstructured domain. *International Educational Data Mining Society*, 2016.

[21] C. K. Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 683–686. International Educational Data Mining Society, 2019.

[22] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *Proceedings of the 18th International Conference on Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer.

[23] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 765–774, New York, NY, USA, 2017. ACM.

[24] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan. Incorporating rich features into deep knowledge tracing. In *Proceedings of the 4th ACM Conference on Learning at Scale*, pages 169–172, New York, NY, USA, 2017. ACM.

[25] S. Zhao, C. Wang, and S. Sahebi. Modeling knowledge acquisition from multiple learning resource types. In *Proceedings of The 13th International Conference on Educational Data Mining*, pages 313–324. International Educational Data Mining Society, 2020.