The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Exercise-Enhanced Sequential Modeling for Student Performance Prediction

Yu Su,[†§] Qingwen Liu,[§‡] Qi Liu,[‡*] Zhenya Huang,[‡] Yu Yin,[‡]
Enhong Chen,[‡] Chris Ding,[♭] Si Wei,[§] Guoping Hu[§]
[†]School of Computer Science and Technology, Anhui University
[‡] Anhui Province Key Lab. of Big Data Analysis and Application, University of Science and Technology of China
[§]iFLYTEK Research, [♭]CSE Department, University of Texas at Arlington
{yusu, qwliu}@iflytek.com, qiliuql@ustc.edu.cn, {huangzhy, yxonic}@mail.ustc.edu.cn
cheneh@ustc.edu.cn, chqding@uta.edu, {siwei, gphu}@iflytek.com

Abstract

In online education systems, for offering proactive services to students (e.g., personalized exercise recommendation), a crucial demand is to predict student performance (e.g., scores) on future exercising activities. Existing prediction methods mainly exploit the historical exercising records of students, where each exercise is usually represented as the manually labeled knowledge concepts, and the richer information contained in the text descriptions of exercises is still underexplored. In this paper, we propose a novel *E*xercise-*E*nhanced *R*ecurrent *N*eural *N*etwork (EERNN) framework for student performance prediction by taking full advantage of both student exercising records and the text of each exercise. Specifically, for modeling the student exercising process, we first design a bidirectional LSTM to learn each exercise representation from its text description without any expertise and information loss. Then, we propose a new LSTM architecture to trace student states (i.e., knowledge states) in their sequential exercising process with the combination of exercise representations. For making final predictions, we design two strategies under EERNN, i.e., *EERNNM with Markov property* and *EERNNA with Attention mechanism*. Extensive experiments on large-scale real-world data clearly demonstrate the effectiveness of EERNN framework. Moreover, by incorporating the exercise correlations, EERNN can well deal with the cold start problems from both student and exercise perspectives.

1 Introduction

Online education systems, such as massive open online course (MOOC) and intelligent tutoring system (ITS), provide students with open access for self-learning. Their prevalence and convenience have attracted great attentions from both educators and general publics (Anderson et al. 2014).

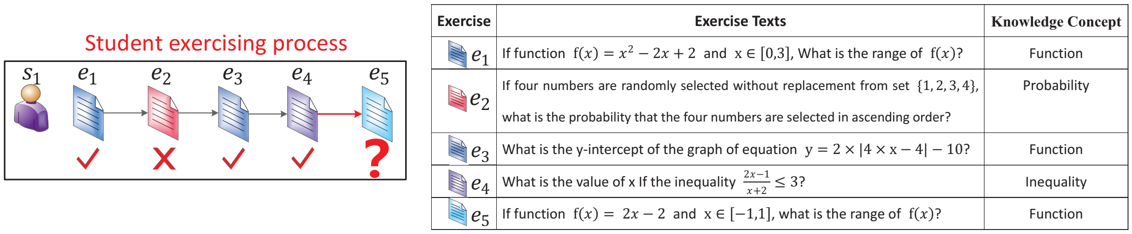
In such education systems, students can get appropriate guidance and acquire knowledge individually in the process of exercising. Figure 1 shows an example of such process of a typical student. Generally, when an exercise is posted (e.g., e_1), the student reads its text and applies knowledge to answer it. Totally, student s_1 has done four exercises during the process. In order to offer students proactive services for their self-improvement, e.g., learning remedy suggestion and personalized exercise recommendation (Kuh et al. 2011), a crucial demand is to predict their performance (e.g., score), i.e., forecasting whether or not a student could answer the exercise (e.g., e_5) correctly in the future (Baker and Yacef 2009).

In the literature, there are many efforts in predicting student performance from both educational psychology and data mining areas, such as cognitive diagnosis (DiBello, Roussos, and Stout 2006), knowledge tracing (Corbett and Anderson 1994), matrix factorization (Thai-Nghe et al. 2010) and deep learning (Piech et al. 2015). Generally, most methods devote efforts to modeling student exercising records for the prediction. However, they just represent each exercise as knowledge concepts, e.g., exercise e_1 in Figure 1 is represented as the concept "Function". These knowledge-specific concepts are usually marked by experts (e.g., teachers) in practice, which may be labor intensive (Desmarais, Beheshti, and Naceur 2012). Meanwhile, these manual representations cannot distinguish individual characteristics (e.g., difficulty) of exercises so that causing server information loss (DiBello, Roussos, and Stout 2006), e.g., exercise e_1 and e_3 are different according to their texts (e_3 is more difficult than e_1) though they are all labeled with "Function". To this end, in this paper, we argue that it is beneficial to combine both student exercising records and the text of each exercise for more precisely predicting student performance.

Unfortunately, there are many technical and domain challenges along this line. First, there are diverse expressions of exercises, which requires a unified way to automatically understand and represent the characteristics of them from a semantic perspective. Second, student performance in the future is deeply relied on their long-term historical exercising, especially on their important knowledge states. How to track the focused information for students is very challenging. At last, student performance prediction task suffers from the "cold start" problem. That is, we usually have to make predictions for new students and new exercises (Wilson et al. 2016). In this scenario, limited information could be exploited, and thus, leading to the poor prediction results.

To address these challenges, we propose a novel *E*xercise-*E*nhanced *R*ecurrent *N*eural *N*etwork (EERNN) framework

[*]Corresponding Author.



Exercise	Exercise Texts	Knowledge Concept
e_1	If function $f(x) = x^2 - 2x + 2$ and $x \in [0,3]$, What is the range of $f(x)$?	Function
e_2	If four numbers are randomly selected without replacement from set $\{1, 2, 3, 4\}$, what is the probability that the four numbers are selected in ascending order?	Probability
e_3	What is the y-intercept of the graph of equation $y = 2 \times \|4 \times x - 4\| - 10$?	Function
e_4	What is the value of x If the inequality $\frac{2x-1}{x+2} \leq 3$?	Inequality
e_5	If function $f(x) = 2x - 2$ and $x \in [-1,1]$, what is the range of $f(x)$?	Function

Figure 1: Example: Left box shows a student exercising process. Right table shows texts and knowledge concepts of exercises.

to predict student performance by taking full advantage of student exercising records and the texts of exercises. Specifically, for student exercising process modeling, inspired by some techniques in nature language processing, we first design a bidirectional LSTM to automatically characterize each exercise semantics by exploiting its text. The learned encodings can be interpreted as exercise-specific embeddings, which capture the individual characteristics of each exercise without any expertise. Then, we propose a new LSTM architecture to trace student states in their sequential exercising process with the combination of exercise representations. For making predictions, leveraged by student states and exercise embeddings, we design two strategies under EERNN framework. The first one is a straightforward yet effective strategy, i.e., *EERNNM with Markov property*, in which students' future performance only depends on current states. Comparatively, the second is a more sophisticated one, *EERNNA with Attention mechanism*, which tracks the focused student states based on similar exercises in the history. In this way, EERNN can naturally predict each student performance on future exercises given her exercising records. Finally, we conduct extensive experiments on a large-scale real-world dataset, which clearly demonstrate the effectiveness of EERNNM and EERNNA. Moreover, by considering the exercise correlations, EERNN framework can effectively deal with the cold start problem when making predictions for new students and new exercises. To the best of our knowledge, this is the first comprehensive attempt to consider both exercising records and exercise texts for student performance prediction.

2 Related Work

The related work can be classified into following categories, i,e., Cognitive Diagnosis, Knowledge Tracing, Matrix Factorization and Deep learning researches.

Cognitive Diagnosis. In the domain of educational psychology, cognitive diagnosis is a technique to predict student performance by discovering student states from their exercising records (DiBello, Roussos, and Stout 2006). Traditional cognitive diagnostic models (CDM) could be grouped into two parts: continuous ones and discrete ones. Among them, item response theory (IRT), as a typical continuous model, characterized each student by a variable from a logistic-like function (Embretson and Reise 2013). Comparatively, discrete models, such as *Deterministic Inputs, Noisy-And gate model* (DINA), represented each student as a binary vector which denoted whether she mastered or not the knowledge concepts required by exercises (De La Torre 2009). To improve prediction results, many variations, such as learning factors analysis (LFA) (Cen, Koedinger, and Junker 2006), performance factors analysis (PFA) (Pavlik Jr, Cen, and Koedinger 2009) and FuzzyCDM (Wu et al. 2015) were proposed by combining other factors.

Knowledge Tracing. Knowledge Tracing is an essential task for tracing the knowledge states of each student separately so that we can predict their performance on future exercising activities, where the idea is similar to typical sequential behavior mining (Shang et al. 2017). In this task, Bayesian knowledge tracing (BKT) was a popular knowledge-specific model, which assumed students' knowledge states as a set of binary variables followed by Hidden Markov Model (Corbett and Anderson 1994). The exercise-knowledge relationship was usually labeled by experts. Many extensions were proposed by considering other factors, e.g., exercise difficulty (Pardos and Heffernan 2011), multiple knowledge concepts (Xu and Mostow 2010) and student individuals (Yudelson, Koedinger, and Gordon 2013). One step further, to improve the prediction performance, researchers also suggested to incorporate IRT or PFA into traditional BKT (Khajah et al. 2014a; 2014b).

Matrix Factorization. Recently, researchers have attempted to leverage matrix factorizations from the perspective of data mining for student performance prediction (Toscher and Jahrer 2010; Thai-Nghe et al. 2010). Usually, the goal of this kind of research is to predict the unknown scores of students as accurate as possible given a student-exercise performance matrix with some known scores. For example, Thai et al. (2015) proposed a multi-relational matrix factorization for the prediction in online learning systems. To capture the changes of student exercising process, Thai et al. (2011) proposed a tensor factorization approach by adding additional time factors. Chen et al. (2017) noticed the effects of both Learning theory and Ebbinghaus forgetting curve theory and incorporated them into a unified probabilistic framework.

Deep Learning. Deep learning is a family of state-of-the-art techniques, which has achieved great success in many applications, e.g., speech recognition (Graves, Mohamed, and Hinton 2013), image classification (Krizhevsky, Sutskever, and Hinton 2012), natural language processing (Mikolov et al. 2013) and also some educational applications like question difficulty prediction (Huang et al. 2017). Inspired by its remarkable performance, deep knowledge tracing (DKT)

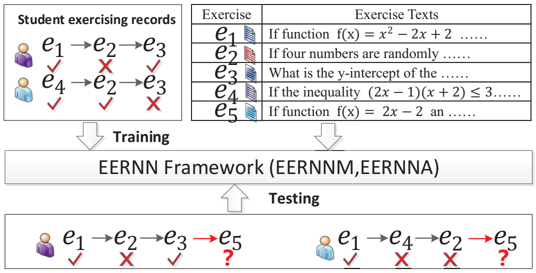

Student exercising records
e1 → e2 → e3
✓ ✗ ✓
e4 → e2 → e3
✓ ✓ ✗
Exercise
Exercise Texts
e1
If function f(x) = x² − 2x + 2 ……
e2
If four numbers are randomly ……
e3
What is the y-intercept of the ……
e4
If the inequality (2x − 1)(x + 2) ≤ 3……
e5
If function f(x) = 2x − 2 an ……
Training
EERNN Framework (EERNNM,EERNNA)
Testing
e1 → e2 → e3 → e5
✓ ✗ ✓ ?
e1 → e4 → e2 → e5
✓ ✗ ✗ ?

Figure 2: Solution Overview.

was the first attempt, to the best of our knowledge, to utilize recurrent neural networks (e.g., RNN and LSTM) to model student exercising process for predicting their performances (Piech et al. 2015). Moreover, by bridging the relationship between exercises and knowledge concepts, Zhang et al. (2017) proposed a dynamic key-value memory network model for student performance prediction.

Our work differs from the previous studies as follows. First, we make use of both student exercising records and exercise texts. Second, the proposed model tracks the focused local effects in the sequential exercising process by attention mechanism, benefiting the prediction. At last, we can well handle the cold start problem by incorporating exercise correlations without any retraining.

3 EERNN Framework

In this section, we first formally introduce student performance prediction problem and give a solution overview. Then we describe the details of EERNN framework. At last, we specify the model learning and testing stage.

Problem and Solution Overview

In an online education system, there are S students and E exercises, where students do exercises individually. We record the exercising process of each student as $s_i = \{(e^i_1, r^i_1), (e^i_2, r^i_2), \ldots, (e^i_T, r^i_T)\}$, where e^i_j represents the j-th exercise solved by student i and r^i_j denotes the corresponding score. Generally, if student i answers exercise j right, r^i_j equals to 1, otherwise it equals to 0. In addition to the logs of student exercising process, we are also given the text descriptions of exercises. Formally, each exercise e_i is combined with a word sequence as $e_i = \{w^i_1, w^i_2, \ldots, w^i_M\}$. For simplicity, we use $s = \{(e_1, r_1), (e_2, r_2), \ldots, (e_T, r_T)\}$ and $e = \{w_1, w_2, \ldots, w_M\}$ to represent each student process s_i and each exercise text e_i, respectively.

Definition 1 ***(Student Performance Prediction Problem).*** *Given the exercising logs of each student and the text descriptions of each exercise from* 1 *to* T*, our goal is to train a unified model* $\mathcal{M}$ *which can be used to predict the scores* $\tilde{r}_{T+1}$ *on the next exercise* e_{T+1} *of each specific student.*

Figure 2 shows the solution overview of our study. From the figure, in the training stage, we train EERNN framework by modeling all student exercising processes with the exercise texts. After that, in the testing stage, EERNN could predict each student performance on future exercises given her individual sequential exercising record.

Specifically, EERNN is a general framework where we can predict student performance based on different strategies. As the details shown in Figure 3, we propose two implementations under EERNN, i.e., *EERNNM with Markov property* and *EERNNA with Attention mechanism*. Both models have the same process for modeling student exercising records yet follow different prediction strategies.

Modeling process of EERNN

The goal of modeling process in EERNN framework is to model the each student exercising sequence with the input s. From Figure 3, this process contains two main components, i.e., *Exercise Embedding* and *Student Embedding*.

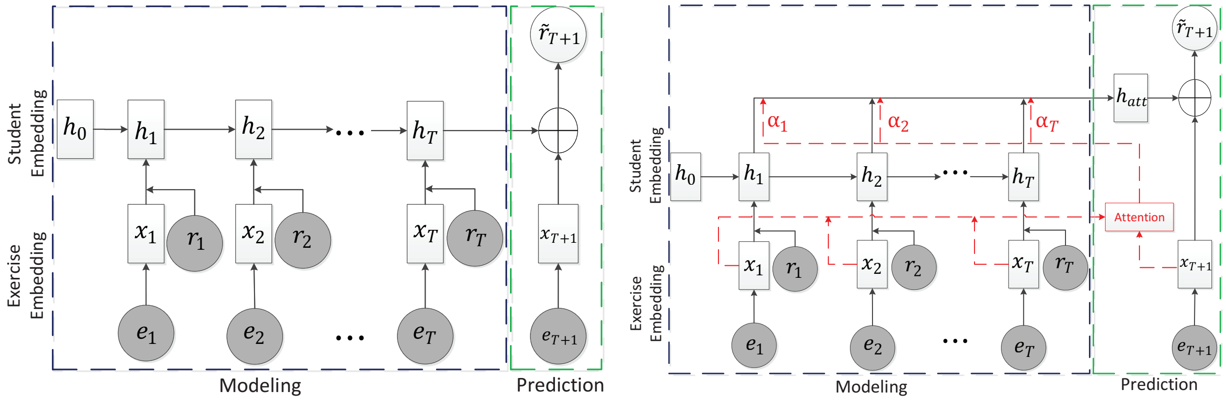
Exercise Embedding. As shown in Figure 3, given student exercising process $s = \{(e_1, r_1), (e_2, r_2), \ldots, (e_T, r_T)\}$, Exercise Embedding learns the semantic representation of each exercise x_i from its text input e_i automatically.

Figure 4 shows the detailed techniques of Exercise Embedding. It is an implementation of a recurrent neural network, which is inspired by the typical one called *Long Short-Term Memory* (LSTM) (Graves, Mohamed, and Hinton 2013) with minor modifications. Specifically, given the exercise's text description with the M words sequence $e_i = \{w_1, w_2, \ldots, w_M\}$, we first take *Word2vec* (Mikolov et al. 2013) to transform each word w_i in exercise e_i into a d_0-dimensional pre-trained word embedding vector. After the initialization, Exercise Embedding updates the hidden state $v_m \in \mathbb{R}^{d_v}$ of each word w_m at m-th word step with the previous hidden state v_{m-1} in a recurrent formula as:

$$\begin{aligned}
i_m &= \sigma(\mathbf{Z^E_{wi}} w_m + \mathbf{Z^E_{vi}} v_{m-1} + \mathbf{b^E_i}),\\
f_m &= \sigma(\mathbf{Z^E_{wf}} w_m + \mathbf{Z^E_{vf}} v_{m-1} + \mathbf{b^E_f}),\\
o_m &= \sigma(\mathbf{Z^E_{wo}} w_m + \mathbf{Z^E_{vo}} v_{m-1} + \mathbf{b^E_o}),\\
c_m &= f_m \cdot c_{m-1} + i_m \cdot \tanh(\mathbf{Z^E_{wc}} w_m + \mathbf{Z^E_{vc}} v_{m-1} + \mathbf{b^E_c}),\\
v_m &= o_m \cdot \tanh(c_m),
\end{aligned} \tag{1}$$

where i_m, f_m, o_m represent the three gates, i.e., input, forget, output, respectively. c_m is a cell memory vector. $\sigma(x)$ is the non-linear *sigmoid* activation function and $\cdot$ denotes the element-wise product between vectors. Besides, the input weighted matrices $\mathbf{Z^E_{w*}} \in \mathbb{R}^{d_v \times d_0}$, recurrent weighted matrices $\mathbf{Z^E_{v*}} \in \mathbb{R}^{d_v \times d_v}$ and bias weighted vectors $\mathbf{b^E_*} \in \mathbb{R}^{d_v}$ are all the network parameters in Exercise Embedding.

Traditional LSTM model learns each word representation by a single direction network and can not utilize the contextual texts from the future word token (Tan et al. 2015). To make full use of the contextual word information of each exercise, we build a bidirectional LSTM taking the word sequence in both forward and backward directions, respectively. As illustrated in Figure 4, at each word step m, the forward layer with hidden word state $\overrightarrow{v}_m$ is computed based on both the previous hidden state $\overrightarrow{v}_{m-1}$ and the current word w_m; while the backward layer updates hidden word state $\overleftarrow{v}_m$ with the future hidden state $\overleftarrow{v}_{m+1}$ and the current word w_m. Therefore, each word hidden representation



(a) EERNNM with Markov property

(b) EERNNA with Attention mechanism

Figure 3: The architectures of two implementations based on EERNN framework, where the shaded and unshaded symbols denotes the observed and latent variables, respectively.

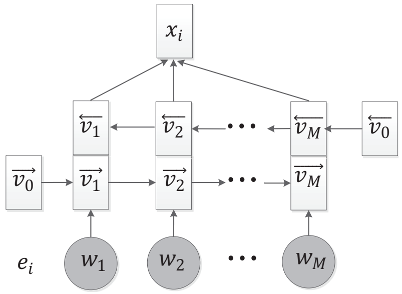


Figure 4: Exercise Embedding of exercise e_i.

v_m can be calculated with the concatenation of the forward state and backward state as $v_m = concatenate(\overrightarrow{v}_m, \overleftarrow{v}_m)$.

After that, to obtain the whole semantic representation of exercise e_i, we exploit the element-wise max pooling operation to merge M word contextual presentations into a global embedding $x_i \in \mathbb{R}^{2d_v}$ as $x_i = \max(v_1, v_2, \ldots, v_M)$.

It is worth mentioning that Exercise Embedding directly learns each exercise semantic representation from its text description without any expert encoding. It can also automatically capture the characteristics (e.g., difficulty) of each exercise and distinguish their individual differences.

Student Embedding. After obtaining each exercise representation x_i from its text e_i by Exercise Embedding, Student Embedding aims at modeling the whole student exercising process and learning the hidden representations of students, which we called *student states*, at different exercising steps combined with the influence of student performance in the history. As shown in Figure 3, EERNN framework relies on two basic assumptions: (1) The student states are influenced by both the exercises and the corresponding scores she got. (2) Students usually learn and forget in their long term sequential exercising process.

Along this line, we exploit a variant of LSTM network for Student Embedding with the input of each specific student exercising process $s = \{(x_1, r_1), (x_2, r_2), \ldots, (x_T, r_T)\}$. Specifically, at each exercising step t, the input to the network is a combined encoding with both exercise embedding x_t and the corresponding score r_t. Since the exercise with right score (i.e., 1) and wrong score (i.e., 0) have different influences on student states during the exercising process, we need to find a appropriate way to distinguish these different effects for a specific student.

Methodology-wise, we first extend the score value r_t to a feature vector $\mathbf{0} = (0, 0, \ldots, 0)$ with the same $2d_v$ dimensions of exercise embedding x_t and then learn the combined input vector $\widetilde{x}_t \in \mathbb{R}^{4d_v}$ as:

$$\widetilde{x}_t = \begin{cases} [x_t \oplus \mathbf{0}] & \text{if} \quad r_t = 1, \\ [\mathbf{0} \oplus x_t] & \text{if} \quad r_t = 0, \end{cases} \tag{2}$$

where $\oplus$ is the operation that concatenates two vectors.

With the combined exercising sequence of a student $s = \{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_T\}$, the hidden student state $h_t \in \mathbb{R}^{d_h}$ at her exercising step t is updated based on the current combined input $\widetilde{x}_t$ and the previous student state h_{t-1} as well as the formula in Eq. (1):

$$\begin{aligned} i_t &= \sigma(\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}\mathbf{i}} \widetilde{x}_t + \mathbf{Z}^{\mathbf{S}}_{\mathbf{hi}} h_{t-1} + \mathbf{b}^{\mathbf{S}}_{\mathbf{i}}), \\ f_t &= \sigma(\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}\mathbf{f}} \widetilde{x}_t + \mathbf{Z}^{\mathbf{S}}_{\mathbf{hf}} h_{t-1} + \mathbf{b}^{\mathbf{S}}_{\mathbf{f}}), \\ o_t &= \sigma(\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}\mathbf{o}} \widetilde{x}_t + \mathbf{Z}^{\mathbf{S}}_{\mathbf{ho}} h_{t-1} + \mathbf{b}^{\mathbf{S}}_{\mathbf{o}}), \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh(\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}\mathbf{c}} \widetilde{x}_t + \mathbf{Z}^{\mathbf{S}}_{\mathbf{hc}} h_{t-1} + \mathbf{b}^{\mathbf{S}}_{\mathbf{c}}), \\ h_t &= o_t \cdot \tanh(c_t), \end{aligned} \tag{3}$$

where $\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}*} \in \mathbb{R}^{d_h \times 4d_v}$, $\mathbf{Z}^{\mathbf{S}}_{\mathbf{h}*} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{b}^{\mathbf{S}}_{*} \in \mathbb{R}^{d_h}$ are the parameters in Student Embedding.

Particularly, the input weight matrix $\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}*} \in \mathbb{R}^{d_h \times 4d_v}$ in Eq. (3) can be divided into two parts, i.e., the positive one $\mathbf{Z}^{\mathbf{S}+}_{\widetilde{\mathbf{x}}*} \in \mathbb{R}^{d_h \times 2d_v}$ and the negative one $\mathbf{Z}^{\mathbf{S}-}_{\widetilde{\mathbf{x}}*} \in \mathbb{R}^{d_h \times 2d_v}$, which can separately capture the influences of exercise e_i with both right and wrong responses for a specific student during her exercising process. Based on these two types of parameters, Student Embedding can naturally model the exercising process to obtain student states by integrating both the exercise texts and the corresponding scores.

Prediction Output of EERNN

After modeling the exercising process of each student from steps 1 to T, in this subsection, we will introduce the detailed techniques of predicting her performance on exercise e_{T+1} at step $T+1$. Psychological results claim that student-exercise performances depend on the student states and exercise characteristics (DiBello, Roussos, and Stout 2006). Following this finding, we propose two implementations of prediction strategies under EERNN framework, i.e., a straightforward yet effective *EERNNM with Markov property* and a more sophisticated *EERNNA with Attention mechanism*, based on both the learned student states $\{h_1, h_2, \ldots, h_T\}$ and exercise embeddings $\{x_1, x_2, \ldots, x_T\}$.

EERNNM with Markov Property. For a typical sequential prediction task, Markov property is a well understood and widely used theory that assumes that the next state depends only on the current state and not on the sequences that precede it (Rabiner and Juang 1986). Given this theory, as shown in Figure 3(a), when an exercise e_{T+1} at $T+1$ step is posted to a student, EERNNM (1) assumes that the student applies current state h_T to solve the exercise; (2) leverages Exercise Embedding to extract semantic representation x_{T+1} from the exercise text e_{T+1}; (3) predicts the performance $\widetilde{r}_{T+1}$ on exercise e_{T+1} of her as:

$$\begin{aligned} y_{T+1} &= ReLU(\mathbf{W_1} \cdot [h_T \oplus x_{T+1}] + \mathbf{b_1}), \\ \widetilde{r}_{T+1} &= \sigma(\mathbf{W_2} \cdot y_{T+1} + \mathbf{b_2}), \end{aligned} \tag{4}$$

where $y_{T+1} \in \mathbb{R}^{d_y}$ denotes the overall presentation for prediction at $T+1$ exercise step. $\{\mathbf{W_1}, \mathbf{W_2}, \mathbf{b_1}, \mathbf{b_2}\}$ are the parameters. $\sigma(x)$ is the *Sigmoid* activation function and $\oplus$ is the concatenation operation.

EERNNM presents a straightforward yet effective way for student performance prediction. However, in most cases, since the current student state h_T is the last hidden state of the LSTM-based architecture in Student Embedding, it may discard some important information when the sequence is long, which is called *Vanish* problem (Hochreiter and Schmidhuber 1997). Thus, EERNNM is unsatisfactory with student state representation for future prediction. To address this question, we propose another more sophisticated prediction strategy, i.e., *EERNNA with Attention mechanism*, to enhance the effects of important student states in the sequential exercising process for prediction.

EERNNA with Attention Mechanism. In Figure 1, students may get similar scores on similar exercises, e.g., student s_1 answers the exercises e_1 and e_3 right due to the possible reason that the both exercises are similar because of the same knowledge concept "Function" behind.

According to this observation, as the red lines illustrated in Figure 3(b), EERNNA assumes that the student state at $T+1$ exercising step is a weighted sum aggregation of all historical student states during the process based on correlations between exercise e_{T+1} and historical ones $\{e_1, e_2, \ldots, e_T\}$. Formally, at next step $T+1$, we define the attentive student state vector h_{att} as:

$$h_{att} = \sum_{j=1}^{T} \alpha_j h_j, \; \alpha_j = cos(x_{T+1}, x_j), \tag{5}$$

where x_j is the exercise embedding at j-th exercise step and h_j is the corresponding student state. *Cosine Similarities* α_j are attention scores for measuring the importance of exercise e_j in the history for new exercise e_{T+1}.

After obtaining attentive student state at step $T+1$, EERNNA predicts her performance on exercise e_{T+1} with the similar operation in Eq. (4) by replacing h_T with h_{att}.

Particularly, through Exercise Embedding, our attention scores α_j not only measure the similarity between exercises from syntactic perspective but also capture the correlations from semantic view (e.g., difficulty correlation), benefiting student state representation for prediction and model explanation. We will conduct experimental analysis for them.

Model learning

Objective function. The whole parameters to be updated in both proposed models mainly come from three parts, i.e., parameters in Exercise Embedding $\{\mathbf{Z}^{\mathbf{E}}_{\mathbf{w}*}, \mathbf{Z}^{\mathbf{E}}_{\mathbf{v}*}, \mathbf{b}^{\mathbf{E}}_{*}\}$, parameters in Student Embedding $\{\mathbf{Z}^{\mathbf{S}}_{\widetilde{\mathbf{x}}*}, \mathbf{Z}^{\mathbf{S}}_{\mathbf{h}*}, \mathbf{b}^{\mathbf{S}}_{*}\}$ and parameters in Prediction Output $\{\mathbf{W}_*, \mathbf{b}_*\}$. The objective function of EERNN is the negative log likelihood of the observed sequence of student exercise process. Formally, at t-th step, let $\widetilde{r}_t$ be the predicted score on exercise e_t through EERNN framework, r_t is the actual score, thus the overall loss for a specific student is defined as:

$$\mathcal{L} = -\sum_{t=1}^{T} (r_t \log \widetilde{r}_t + (1 - r_t) \log(1 - \widetilde{r}_t)). \tag{6}$$

The objective function is minimized using the Adam optimization (Kingma and Ba 2014). More details of settings will be specified in the experiments.

Testing Stage

So far, we have discussed the whole training stage of EERNN. After obtaining the trained EERNN based model $\mathcal{M}$, in the testing stage, given an individual student exercising record $s^p = \{(e_1^p, r_1^p), (e_2^p, r_2^p), \ldots, (e_T^p, r_T^p)\}$, we could predict her performance on the next exercise e_{T+1}^p followed by the steps: (1) apply model $\mathcal{M}$ to fit her exercising process s^p to get the student state at T step for prediction (i.e., h_T^p in EERNNM or h_{att}^p in EERNNA); (2) leverage Exercise Embedding in $\mathcal{M}$ to extract exercise embedding x_{T+1}^p; (3) predict her performance $\widetilde{r}_{T+1}^p$ with Eq. (4) (replacing h_T^p with h_{att}^p in EERNNA).

Please note that, in the testing stage, student s^p can be either any one that exists in the training stage or a new student that never shows up. Equally, exercise e_i^p in s^p can also be either learned exercise or any new exercise. Specifically, when a new student without any historical record is coming, at step 1, EERNN can model him/her first state h_1 for the prediction on first exercise with the non-personalized prior h_0 (Figure 3) using Eq. (3). This is the comprehensive prediction generated from all trained student records. After that, EERNN can fit her own exercising process and make personalized predictions on following exercises. Similarly, when a new exercise is coming, Exercise Embedding in EERNN can learn its representation only based on its original text. Last

Table 1: The statistics of mathematics dataset.

Statistics	Original	Pruned
# of records	68,337,149	5,596,075
# of students	110,0726	84,909
# of exercises	1,825,767	15,045
# of knowledge concepts	550	447
Avg. exercises per student	\	65.9
Avg. words per exercise	\	27.3
Avg. knowledge concepts per exercise	\	54.2
Avg. exercises per knowledge concept	\	1.8

but not least, all the testing stage of EERNN do not require any model retraining. Therefore, EERNN can naturally handle with the cold start problem when making predictions for new students and new exercises.

4 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of EERNN from various aspects: (1) the prediction performance of EERNN against the baselines in both future and cold-start scenarios; (2) the attention effectiveness in EERNN; (3) meaningful visualization.

Experimental Dataset

The experimental dataset supplied by iFLYTEK Co., Ltd. is collected from Zhixue[1], a widely-used online learning system, which provides senior high school students with a large exercise resources for exercising. In this paper, we conduct experiments on the mathematical data records because the mathematical dataset is currently the largest in the system. To make sure the reliability of the experimental results, we filter the students that did less than 10 exercises and the exercises that no students have done. Table 1 shows the statistics of the dataset before and after preprocessing. Note that most exercises contain less than 2 knowledge concepts, and 54 exercises on average are related to one specific knowledge concept. These observations prove that the way to represent exercises as knowledge concepts cannot distinguish differences among exercises, causing some information loss.

Experimental Setup

Word Embedding. Please note that the word embeddings of mathematical exercises in Exercise Embedding are different from traditional ones, like news, because there are some mathematical formulas in the exercise texts. Therefore, we develop a *formula tool*[2] to transform each formula into a semantic feature word. After this initialization, each exercise is transformed into a word/feature sequence. Next, to extract the exclusive word embeddings for mathematics, we construct a corpus of all 1,825,767 exercises shown in Table 1 and train each word in exercises into an embedding vector with the 50 dimensions (i.e., d_0 = 50) by the public *word2vec* tool (Mikolov et al. 2013).

EERNN Setting. We now specify the network initializations in EERNN, we set the dimension d_v of hidden states in Exercise Embedding as 100, d_h of hidden states in Student Embedding as 100, and d_y of overall presentation vectors in prediction stage as 50, respectively.

Training Setting. To set up the training process, we follow (Orr and Müller 2003) and randomly initialize all parameters in EERNN framework with uniform distribution in the range $(-\sqrt{6/(ni+no)}, \sqrt{6/(ni+no)})$, where ni and no are the numbers of input and output feature sizes of the corresponding ones, respectively. Besides, we set mini batches as 32 for training and also use dropout (with probability 0.1) to prevent overfitting.

Comparison Methods. To demonstrate the effectiveness of EERNN framework, we compare our two implementations, i.e., EERNNM and EERNNA, with many models from various perspectives. First, to highlight the effectiveness of Exercise Embedding in EERNN, i.e., to validate whether or not it is effective to incorporate exercise texts for the prediction, we introduce two variants of EERNN, which are denoted as LSTMM and LSTMA. Then, we consider some traditional models: *Item Response Theory* (IRT), *Bayesian Knowledge Tracing* (BKT) from educational psychology and *Probabilistic Matrix Factorization* (PMF) and *Deep Knowledge Tracing* (DKT) from data mining area as the baselines. The details of them are as follows:

- *LSTMM*: LSTMM is a variant of EERNN framework. Here, for modeling process, we do not use the exercise texts and just utilize knowledge-specific representations for exercises as inputs and leverage traditional LSTM to model student exercising process. For prediction, LSTMM follows Markov property strategy as well as EERNNM.
- *LSTMA*: LSTMA is another variant of EERNN framework which contains the same modeling process as LSTMM. For prediction, LSTMA follows the strategy of Attention mechanism as well as EERNNA.
- *IRT*: IRT is a cognitive diagnostic model that models student exercising records by a logistic-like function (DiBello, Roussos, and Stout 2006).
- *BKT*: BKT is a typical knowledge tracing model for prediction that assumes the knowledge states of each student as a set of binary variables and traces them with a kind of hidden Markov model (Corbett and Anderson 1994).
- *PMF*: PMF is a factorization model that projects students and exercises into latent factors (Thai-Nghe et al. 2011).
- *DKT*: DKT is a recent deep learning method that leverages recurrent neural network to model student exercising process for prediction (Piech et al. 2015). The inputs are one-hot encodings of student-knowledge representations.

All models are implemented by PyTorch (Paszke and Chintala) using Python on a Linux server with four 2.0GHz Intel Xeon E5-2620 CPUs and a Tesla K20m GPU. All models are tuned to have the best performance.

Evaluation Metrics. We evaluate EERNN on student performance prediction from both regression and classification perspectives (Fogarty, Baker, and Hudson 2005; Wu et al. 2015; 2017). For regression, we select *Mean Absolute Error*

[1] http://www.zhixue.com

[2] The details of this tool are not the major focus of this work.

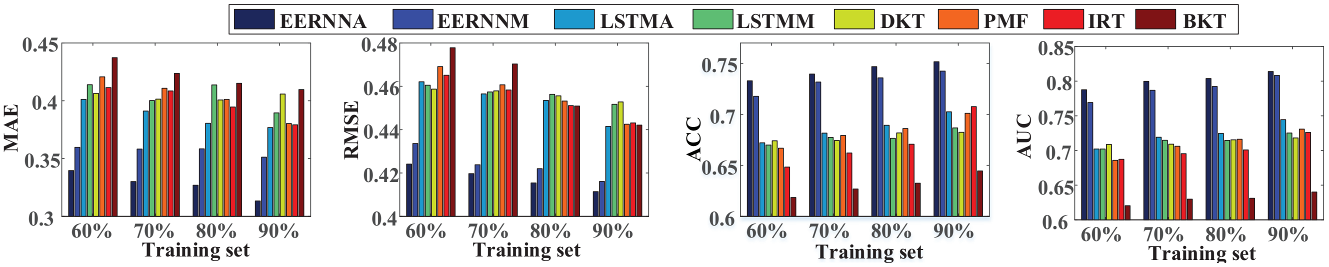


Figure 5: Overall results of student performance prediction on four metrics.

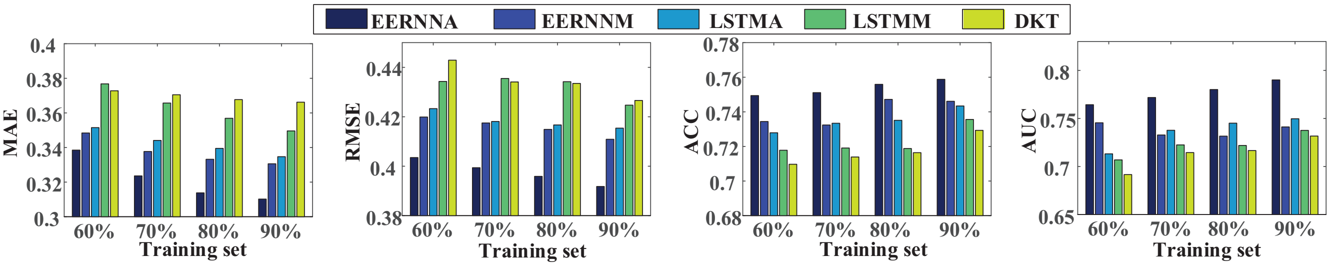


Figure 6: Results of student performance prediction on cold start new exercises on four metrics.

(MAE) and *Root Mean Square Error* (RMSE), to quantify the distance between predicted scores and the actual ones. The smaller the values are, the better the results have.

Besides, we also treat the prediction problem as a classification task, where a record with score 1 (0) indicates a positive (negative) instance. Thus, we use two metrics, i.e., *Area Under an ROC Curve* (AUC), *Prediction Accuracy* (ACC), for measuring. Generally, the value 0.5 of AUC or ACC represents the performance prediction result by randomly guessing, and the larger, the better.

Experimental Results

Student Performance Prediction. We partition the dataset to compare the results of all models on student performance prediction. For each student's sequential exercising record, we use the beginning 60%, 70%, 80%, 90% exercises as training sets, and the remains are as testing sets, respectively. We repeat all experiments 5 times and report the average results using all metrics, which are shown in Figure 5.

There are several observations. First, both EERNNA and EERNNM perform better than all other methods. The results indicate that EERNN framework can make full use of exercising records and exercise texts, benefiting the prediction. Second, models with Attention mechanism (EERNNA, LSTMA) outperform those with Markov property (EERNNM, LSTMM), which demonstrates that it is effective to track focused student embeddings based on similar exercises for the prediction. Third, both EERNNA and EERNNM generate better result than their variants (LSTMA, LSTMM) and DKT, showing the effectiveness of Exercise Embedding. This observation also suggests that EERNN could alleviate the information loss caused by knowledge-specific representations. Last but not least, we observe that traditional models (IRT, PMF and BKT) do not perform as well as all deep learning models in most cases. We guess a possible reason is that all these RNN based models can capture the change of student exercise process, where the deep neural network structures are suitable for student performance prediction.

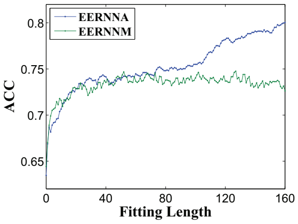
In summary, all above evidences demonstrate that EERNN framework has a good ability to predict student performance by taking full advantage of both the exercising records and the texts of exercises.

Cold Start Prediction. We conduct experiments to evaluate the performance of EERNN in the cold start situation from exercise perspective. Here, we only test the prediction results of the models, trained on 60%, 70%, 80%, 90% training sets, on new exercises (that never show up in training) in the corresponding testing sets, using all metrics, respectively. Please note that, we do not change any training process and just select cold start exercises for testing, thus all the testing do not need retraining. For better illustration, we report the results of all 5 deep learning based models.

As shown in Figure 6, there are also the similar experimental observations as Figure 5, which demonstrates the effectiveness of EERNN framework again. These results indicate the superiority of EERNN framework that it can well deal with the cold start problem when predicting student performance on new exercises.

Attention Effectiveness. As mentioned in Section 3, we hold that *EERNNA with Attention mechanism* can track the focused states of students during the student exercising process to improve prediction performance, which is superior to EERNNM. To highlight the attention effects, we compare EERNNA and EERNNM (trained on 90% training set) for prediction in the corresponding testing set with different fitting lengths of all students, using ACC and AUC metrics.

From Figure 7, both EERNNA and EERNNM generate



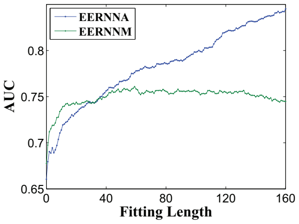


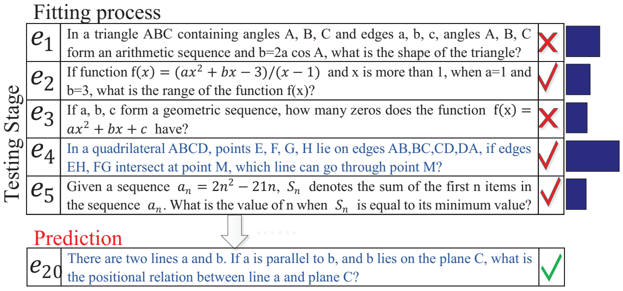
Figure 7: The attention effects in fitting process for testing.

similar results when the fitting sequence is short (less than 40). However, as the sequence length increases, EERNNA performs better gradually. Moreover, when it surpasses about 60, EERNNA outperforms EERNNM significantly on both metrics. This phenomenon indicates that EERNNM is effective for the prediction at the beginning of student exercising process but discards some important information when the sequence is long. Comparatively, EERNNA enhances the local student states with attention mechanism, benefiting the prediction. Besides, notice that both EERNNA and EERNNM obtain about 0.65 results on both metrics (better than randomly guessing 0.5) by the prior student state h_0 (Figure 3) in the case of predicting the first performance of students without any record (fitting length is 0). This finding also shows that EERNN can ensure a certain effect when meeting the cold start problem from student perspective.

Visualization. Particularly, EERNNA has a great power of explaining and analyzing the prediction results for each student by attention mechanism, i.e., the attention score α in Eq. (5). Figure 8 illustrates the attention scores for a student in the experiment as an example. Here, EERNNA predicted the student can answer exercise e_{20} right, because she even got right answers on a more difficult similar exercise e_4 in the past. From their texts, we can conclude that both e_{20} and e_4 are all "Geometry" exercises and e_4 is more difficult than e_{20}. This visualization hints that EERNNA provides a good way for result analysis and model explanations, which is also meaningful in the educational applications.

5 Conclusions

In this paper, we presented a novel *E*xercise-*E*nhanced *R*ecurrent *N*eural *N*etwork (EERNN) framework to predict student future performance by taking full advantage of student exercising records and the texts of exercises. Specifically, for modeling student exercising process, we first designed a BiLSTM to extract exercise semantic representations from texts without any expertise and information loss. Then, we proposed another LSTM architecture to trace student states by embedding exercise encodings. For making prediction, we designed two strategies under EERNN, i.e., a straightforward *EERNNM with Markov property* and a sophisticated *EERNNA with Attention mechanism*. Comparatively, EERNNA can track the focused information for making prediction, which is superior to EERNNM. Finally, extensive experiments on a large-scale real-world dataset demonstrated the effectiveness of EERNN framework and also claimed that EERNN could well deal with the cold start problem.



e_1 In a triangle ABC containing angles A, B, C and edges a, b, c, angles A, B, C form an arithmetic sequence and b=2a cos A, what is the shape of the triangle? ✗

e_2 If function f(x) = $(ax^2 + bx - 3)/(x - 1)$ and x is more than 1, when a=1 and b=3, what is the range of the function f(x)? ✓

e_3 If a, b, c form a geometric sequence, how many zeros does the function f(x) = $ax^2 + bx + c$ have? ✗

e_4 In a quadrilateral ABCD, points E, F, G, H lie on edges AB,BC,CD,DA, if edges EH, FG intersect at point M, which line can go through point M? ✓

e_5 Given a sequence $a_n = 2n^2 - 21n$, S_n denotes the sum of the first n items in the sequence a_n. What is the value of n when S_n is equal to its minimum value? ✓

e_{20} There are two lines a and b. If a is parallel to b, and b lies on the plane C, what is the positional relation between line a and plane C? ✓

Figure 8: Attention visualization in EERNNA of a student. We predict her performance on e_{20} based on her past 19 exercise records (we only show the first 5 exercises for better illustration). Right bars show the attention scores of all exercises based on e_{20}.

In the future, we would like to consider the characteristics of different exercise types (e.g., the subjective exercises with continuous scores) for the prediction. Second, we will extend EERNN framework to incorporate the information of knowledge concepts. Third, we are also willing to integrate some educational theories (e.g., learning and forgetting curves) (Anzanello and Fogliatto 2011; Von Foerster 2007).

6 Acknowledgments

This research was partially supported by grants from the National Basic Research Program of China (973 Program Grant No. 2015CB351705) and the National Natural Science Foundation of China (Grants No. 61572030, 61672483, U1605251 and 61403358). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299).

References

Anderson, A.; Huttenlocher, D.; Kleinberg, J.; and Leskovec, J. 2014. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*, 687–698. ACM.

Anzanello, M. J., and Fogliatto, F. S. 2011. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics* 41(5):573–583.

Baker, R. S., and Yacef, K. 2009. The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining* 1(1):3–17.

Cen, H.; Koedinger, K.; and Junker, B. 2006. Learning factors analysis-a general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems*, volume 4053, 164–175. Springer.

Chen, Y.; Liu, Q.; Huang, Z.; Wu, L.; Chen, E.; Wu, R.; Su, Y.; and Hu, G. 2017. Tracking knowledge proficiency of students with educational priors. In *Proceedings of the 26th ACM International Conference on Conference on Information and Knowledge Management*. ACM.

Corbett, A. T., and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4(4):253–278.

De La Torre, J. 2009. Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics* 34(1):115–130.

Desmarais, M.; Beheshti, B.; and Naceur, R. 2012. Item to skills mapping: deriving a conjunctive q-matrix from data. In *Intelligent tutoring systems*, 454–463. Springer.

DiBello, L. V.; Roussos, L. A.; and Stout, W. 2006. 31a review of cognitively diagnostic assessment and a summary of psychometric models. *Handbook of statistics* 26:979–1030.

Embretson, S. E., and Reise, S. P. 2013. *Item response theory*. Psychology Press.

Fogarty, J.; Baker, R. S.; and Hudson, S. E. 2005. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, 129–136. Canadian Human-Computer Communications Society.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Huang, Z.; Liu, Q.; Chen, E.; Zhao, H.; Gao, M.; Wei, S.; Su, Y.; and Hu, G. 2017. Question difficulty prediction for reading problems in standard tests. In *AAAI*, 1352–1359.

Khajah, M.; Wing, R. M.; Lindsey, R. V.; and Mozer, M. C. 2014a. Incorporating latent factors into knowledge tracing to predict individual differences in learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, 99–106.

Khajah, M. M.; Huang, Y.; González-Brenes, J. P.; Mozer, M. C.; and Brusilovsky, P. 2014b. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *Proceedings of Workshop on Personalization Approaches in Learning Environments (PALE 2014) at the 22th International Conference on User Modeling, Adaptation, and Personalization*, 7–12. University of Pittsburgh.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Kuh, G. D.; Kinzie, J.; Buckley, J. A.; Bridges, B. K.; and Hayek, J. C. 2011. *Piecing together the student success puzzle: research, propositions, and recommendations: ASHE Higher Education Report*, volume 116. John Wiley & Sons.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Orr, G. B., and Müller, K.-R. 2003. *Neural networks: tricks of the trade*. Springer.

Pardos, Z., and Heffernan, N. 2011. Kt-idem: introducing item difficulty to the knowledge tracing model. *User Modeling, Adaption and Personalization* 243–254.

Paszke, A., and Chintala, S. Pytorch.

Pavlik Jr, P. I.; Cen, H.; and Koedinger, K. R. 2009. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, 505–513.

Rabiner, L., and Juang, B. 1986. An introduction to hidden markov models. *ieee assp magazine* 3(1):4–16.

Shang, S.; Chen, L.; Jensen, C. S.; Wen, J.-R.; and Kalnis, P. 2017. Searching trajectories by regions of interest. *IEEE Transactions on Knowledge and Data Engineering* 29(7):1549–1562.

Tan, M.; Santos, C. d.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Thai-Nghe, N., and Schmidt-Thieme, L. 2015. Multi-relational factorization models for student modeling in intelligent tutoring systems. In *Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on*, 61–66. IEEE.

Thai-Nghe, N.; Drumond, L.; Krohn-Grimberghe, A.; and Schmidt-Thieme, L. 2010. Recommender system for predicting student performance. *Procedia Computer Science* 1(2):2811–2819.

Thai-Nghe, N.; Drumond, L.; Horváth, T.; Krohn-Grimberghe, A.; Nanopoulos, A.; and Schmidt-Thieme, L. 2011. Factorization techniques for predicting student performance. *Educational recommender systems and technologies: Practices and challenges* 129–153.

Toscher, A., and Jahrer, M. 2010. Collaborative filtering applied to educational data mining. *KDD cup*.

Von Foerster, H. 2007. *Understanding understanding: Essays on cybernetics and cognition*. Springer Science & Business Media.

Wilson, K. H.; Xiong, X.; Khajah, M.; Lindsey, R. V.; Zhao, S.; Karklin, Y.; Van Inwegen, E. G.; Han, B.; Ekanadham, C.; Beck, J. E.; et al. 2016. Estimating student proficiency: Deep learning is not the panacea. In *In Neural Information Processing Systems, Workshop on Machine Learning for Education*.

Wu, R.-z.; Liu, Q.; Liu, Y.; Chen, E.; Su, Y.; Chen, Z.; and Hu, G. 2015. Cognitive modelling for predicting examinee performance. In *IJCAI*, 1017–1024.

Wu, R.; Xu, G.; Chen, E.; Liu, Q.; and Ng, W. 2017. Knowledge or gaming?: Cognitive modelling based on multiple-attempt response. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 321–329. International World Wide Web Conferences Steering Committee.

Xu, Y., and Mostow, J. 2010. Using logistic regression to trace multiple sub-skills in a dynamic bayes net. In *Educational Data Mining 2011*.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, 171–180. Springer.

Zhang, J.; Shi, X.; King, I.; and Yeung, D.-Y. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, 765–774. International World Wide Web Conferences Steering Committee.