Context-aware Knowledge Tracing Integrated with The Exercise Representation and Association in Mathematics

Tao Huang[1], Mengyi Liang[2], Huali Yang[1], Zhi Li[2], Tao Yu[2] and Shengze Hu[1]
[1] National Engineering Laboratory for Educational Big Data, Central China Normal University, Wuhan, China
{tmht, yanghuali}@mail.ccnu.edu.cn, shengzehu@mails.ccnu.edu.cn
[2] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China
{mengyiliang, zhili, yt2542106000}@mails.ccnu.edu.cn

ABSTRACT
Influenced by Covid-19, online learning has become one of the most important forms of education in the world. In the era of intelligent education, knowledge tracing(KT) can provide excellent technical support for individualized teaching. For online learning, we come up with a new knowledge tracing method that integrates mathematical exercise representation and association of exercise(ERAKT). In the aspect of exercise representation, we represent the multi-dimensional features of the exercises, such as formula, text and associated concept, by using ontology replacement method, language model and embedding technology, so we can obtain the unified internal representation of exercise. Besides, we utilize the bidirectional long short memory neural network to acquire the association between exercises, so as to predict his performance in future exercise. Extensive experiments on a real dataset clearly proved the effectiveness of ERAKT method, they also verified that adding multi-dimensional features and exercise association can indeed improve the accuracy of prediction.

Keywords
Knowledge tracing. Context-aware. Exercise representation.

1. INTRODUCTION
As one of the key technologies of adaptive learning, knowledge tracking has become a research hotspot in adaptive education. The main task of knowledge tracking is to automatically track students' acquisition knowledge level with time according to their historical learning trajectory, so as to accurately predict their performance in future learning. In actual teaching, teachers can adjust teaching plan dynamically by predicting the result, improve teaching quality and teaching efficiency, and help teachers to achieve accurate teaching goal.

Knowledge Tracing method (KT) was first proposed by Atkinson. Bayesian knowledge tracing method (BKT) [1] is one of the most popular knowledge tracing methods in the early stage. BKT assumes that students will never forget a knowledge concept once they have mastered it, which is not in line with the actual teaching situation. Later, with the continuous development of deep learning, more and more scholars combined knowledge tracing tasks with deep learning methods, among which Deep Knowledge Tracing (DKT) [2] is the most popular and commonly used one. DKT partly solves the assumption error problem in BKT which does not conform to the actual teaching situation, and can more accurately represent the concept proficiency of learners. However, the assumption of concept state represented by a hidden layer in DKT is inaccurate, making a student's mastery level difficult to track. Furthermore, Jiani Zhang et al. proposed Dynamic Key-Value Memory Networks for Knowledge Tracing (DKVMN) [3] based on memory neural networks, and DKVMN is significantly better than BKT and DKT in terms of performance effect. In recent years, the University of Science and Technology of China team proposed some methods which integrated exercise records and exercise materials into KT based on the existing KT methods, such as EKT[11], qDKT[12], etc., which has stronger explanatory power and gradually improved performance effect.

However, most of the traditional knowledge tracing methods only consider the exercises records of students, using the covered concepts to index the exercises, ignoring the influence of exercise formula, text or concepts on a student's knowledge state. In fact, besides exercise interaction records, the multi-dimensional information of exercises has an important impact on a student's performance. Therefore, in order to solve the above problems, we propose a mathematical knowledge tracing method that integrates the representation and association of a student's exercises, so as to solve the problem of information loss caused by ignoring multi-dimensional representation and association of exercises in traditional knowledge tracing and improve accuracy of the method. Contributions of ERAKT are as follows:

(1) We propose a new context-aware knowledge tracing method that can automatically learn and predict a student's performance in the next exercise.
(2) We propose an exercise representation method that integrates multi-dimensional information, including text, formulate and associated concept.
(3) We propose a sequential question association mining method based on a bidirectional neural network to acquire association content between exercises.

2. RELATED WORK
2.1 Semantic representation
In the domain of text processing, the most important task is to transform text into a vector form which could be understood and processed by computers, that is, semantic representation. There are

Tao Huang, Mengyi Liang, Huali Yang, Zhi Li, Tao Yu and Shengze Hu "Context-aware Knowledge Tracing Integrated with The Exercise Representation and Association in Mathematics". 2021. In: Proceedings of The 14th International Conference on Educational Data Mining (EDM21). International Educational Data Mining Society, 360-366. https://educationaldatamining.org/edm2021/
EDM '21 June 29 - July 02 2021, Paris, France

many ways to get semantic representation, such as Word2vec, TextCNN, FastText, Bert etc.

Word2Vec[4] was proposed by Google's Mikolov team. It is the preliminary application of neural networks in semantic representation. Word2Vec conducts fixed-dimensional vectors to represent words. For sentences, the Doc2Vec method [5] is derived, which establishes a model by means of a neural network structure. Paragraph vectors are obtained during the training of the model. Both of them are typical unsupervised text representation methods. Compared with the traditional bag-of-words method, they can better integrate the internal information of exercise, such as context, semantics and word order.

Yoon Kim modified the input layer of traditional CNN. In 2014, he proposed a text classification method named TextCNN[6], which has a simpler network structure, smaller amount of calculations and faster speed of training. So compared with traditional CNN method, TextCNN performs better in field of semantic representation. Another method to get semantic representation is FastText [7]. FastText can train word vectors by itself without requiring pre-trained word vectors, which speeding up training and testing while maintaining high accuracy.

The Bert [8] method was also proposed by the Google team to solve the semantic representation problem. To deal with the effects of polysemous words in sentences, Bert exploits the transfomer model with Self-attention and Multi-Headed Attention mechanisms [9], which combines the context of the sentence to determine the specific semantics. Bert has a two-way function, allowing for more accurate results and adaptive learning in a multi-tasking environment.

2.2 Knowledge tracing considering multi-dimensional characteristics of exercises

With the rapid development of deep learning, more and more scholars exploit different deep learning methods to represent exercises in order to complete the knowledge tracking task and attempt to comprehensively consider the impact of different features of the exercises on knowledge tracing tasks. The multi-dimensionality mainly include textual materials and concepts involved in the exercises.

Therefore, the majority of existing KT methods utilize concepts to index exercises to avoid over-parameterization. For example, both DKT and DKVMN treat all the exercises covering the same concept as a single one. Compared with the former, the key-value matrix in DKVMN extends the hidden feature representation of the exercises, but it still does not take advantage of the characteristics of other dimensions. The Prerequisite-driven deep knowledge tracing(PDKT) [23] method integrates the structural information between concepts with the help of the Q matrix in the cognitive diagnosis theory, and specifically considers the contextual relationship between concepts. The self-attentive knowledge tracing (SAKT) [24] method exploits concepts to index exercises and introduces a self-attention mechanism to consider the degree of relevance between concepts. The Context-Aware Attentive Knowledge Tracing method(AKT) [13] utilizes a novel monotonic attention mechanism that relates a student's future responses to assessment exercises to their past responses; attention weights are computed using exponential decay and a context-aware relative distance measure, in addition to the similarity between exercises. Moreover, AKT utilizes the Rasch model to capture individual differences between exercises. The Graph-based Knowledge Tracing(GKT) [25] also introduces concepts to index exercises, at the same time constructs a graph method to represent the association between concepts, updates the student's knowledge status through the GRU mechanism.

The exercise materials which KT methods consider are mainly text materials and the concepts covered. EERNN (Exercise-Enhanced Recurrent Neural Network) [10] and qDKT(Question centric Deep Knowledge Tracing)[12] predict a student's performance only by making full use of his practice records and text of exercises. EKT (Exercise-aware Knowledge Tracing for Student Performance Prediction) [11] is an improved method based on EERNN. It is the first method to comprehensively consider the influence of a student's practice records and exercise materials (concepts and text contained) on his performance. But it is worth noticing that in EKT exercise text is represented by LSTM, due to its internal structure problem, LSTM can't parallel computing, resulting in dealing with text slower, so the effect is not very satisfactory. Exploring Hierarchical Structures for Recommender Systems (EHFKT)[21] make full use of concepts, difficulty, and semantic features to represent exercises. The first two are embedded using TextCNN, and semantic features are extracted using Bert. The Introducing Problem Schema with Hierarchical Exercise Graph for Knowledge Tracing(HGKT) [22] method exploits a hierarchical graph neural network to learn the graphical structure of the exercises. It also introduces two attention mechanisms to better mine knowledge state of learners, and utilizes the K&S diagnosis matrix to obtain the diagnosis result.

3. PROPOSED METHOD

The student's knowledge mastery is tracked by observing his interactive information in different exercises. It is a supervised learning sequence prediction problem in the field of machine learning. In this section, we will first define the problem and then describe the proposed method in detail.

3.1 Problem Definition

Assuming that each student does exercises separately, we define the student sequence $s = \{(e_1, r_1), (e_2, r_2), \dots\dots, (e_T, r_T)\}$, where e_T belongs to exercises sequence E, which represents the exercises done by the student at time T. Usually, 0 or 1 is applied to mark whether his answer is correct, i.e. 1 means correct, 0 means wrong. In the ERAKT method, we add the text content of each exercise into the student's exercise sequence E, because we mainly focus on knowledge tracing task in mathematics, which generally contain not only words but also specific mathematical elements, so the text is represented as $e = \{w, f\}$, where $w = \{w_1, w_2, \dots\dots, w_M\}$ represents the words in the exercise, and $f = \{f_1, f_2, \dots\dots, f_M\}$ represents the mathematics components. Furthermore, the corresponding concept is a key information in KT task, $k = \{k_1, k_2, \dots\dots, k_M\}$, which is summarized into a concepts matrix for embedding, and the student sequence after embedding turns into sequence $s = \{(e_1, k_1, r_1), (e_2, k_2, r_2), \dots\dots, (e_T, k_T, r_T)\}, s \in S$.

The ultimate goal of our ERAKT method is to track a student's knowledge status through hidden layers to predict his performance in future exercise, that is, his response to exercise e_{T+1} at the next moment T+1. Besides, we take into account the student's record of exercises, the text content of the exercises and the concepts included. The ERAKT framework is shown in Figure 1. The method includes three major parts: exercise representation module, exercise association module and the performance prediction module.

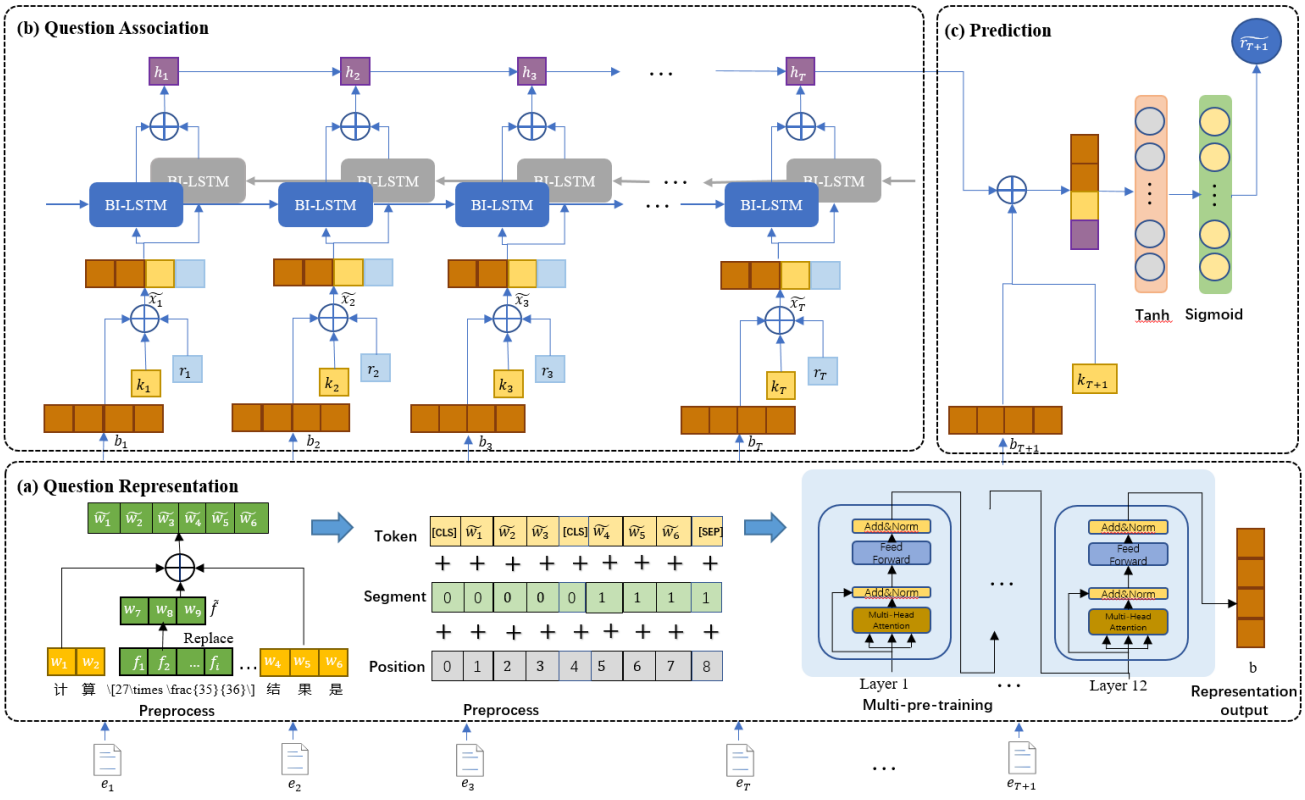

Figure 1. ERAKT framework. (a) The process of obtaining the exercise representation. After the replacement of the formula, we splice the formula with the original text, and the final exercise representation vector is obtained through 3 embedding layers and 12 layers of encoder. (b) Combine the exercise representation vector with the knowledge concepts and student response corresponding to the exercise, and input them into the Bi-LSTM network together to obtain the student's hidden state representation. (c) The student response prediction part, which predicts the student's response to the exercise at time T+1.

3.2 Exercise Representation Method

The information of the exercise includes the material itself and the concepts associated with it. To realize the unified representation of mathematics exercises, firstly, we need to construct the different dimension features in the exercises to represent the material itself and its associated concepts, and then integrate the feature representations of multidimensional exercises into a unified feature vector.

3.2.1 Preprocessing of exercise formula

Mathematical exercises involve text and formulas, which are represented by LaTex, and we need to convert formulas into unified text expressions in advance. Since the LaTeX formula in the exercises follows a set of unified coding rules, we first replace LaTeX formulas uniformly through ontology replacement method, then perform unified preprocessing together with other text.

In the unified preprocessing of the formula text, the entities and attributes in the exercises are identified and replaced from entities to attributes. During the replacement process, the replaced entities or attributes and the replaced forms are saved in a dictionary. , That means, replace these formula texts $f = \{f_1, f_2, \ldots\ldots, f_M\}$ to obtain $\tilde{f} = \{\widetilde{f_1}, \widetilde{f_2}, \ldots\ldots, \widetilde{f_M}\}$. The partial replacement relationship is shown in Table 1.

Table 1. Formula replacement table.

f	$\tilde{f}$
complement	补集
sqrt	根式
^	幂
+	加号
cm	厘米
pi	圆周率

After the replacement, splice $\tilde{f}$ with the text $w = \{w_1, w_2, \ldots\ldots, w_M\}$ according to the original position, and get the text representation of the exercise $\widetilde{w}$.

$$\widetilde{w} = w \oplus \tilde{f} \quad (1)$$

For exercise representation $\widetilde{w}$, we apply python's Chinese word segmentation package Jieba for word segmentation. Jieba has three segmentation modes: precise mode, full mode and search engine mode. Here we utilize precise mode to accurately segment a complete sentence into independent words according to the segmentation algorithm. Then use a self-built stop word list to delete some words that cannot express specific meanings. This specific process is shown in Figure 2.

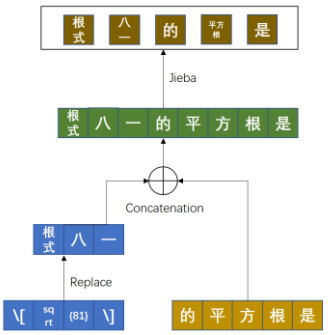

Figure 2. Example diagram of exercise formula preprocessing.

3.2.2 *Vector representation of multi-dimensional exercise*

In view of the superiority of Bert in the field of natural language processing, we exploit Bert to conduct self-supervised learning and training on the text features of exercises. We employ three embedding layers and a 12-layer encoder to pre-train the exercise representation. As shown in the question representation part in Figure 1, it includes three layers. The function of the token embedding layer is to convert each word segmentation into a 768-dimensional vector, then the segment embedding layer distinguishes differences between the vectors of two sentences, the position embedding layer can help understand the order of words. When all the embedding processes are done, we add the results of each layer element by element to get the input of the Bert encoder layer. After encoding by the 12-layer encoder, we can get the final text vector representation b_T.

$$b_T = e_{Token} + e_{Segment} + e_{Position} \quad (2)$$

In this way, semantic information of exercises can be obtained automatically without any extra expert manual coding.

3.2.3 *Representation of concepts*

In the dataset, each exercise will be marked with concepts involved, $k = \{k_1, k_2, \dots\dots, k_M\}$. With reference to EKT, we select the first concept k_1,which is also the most relevant one to represent the concept involved in exercise, then all the concepts of exercises are encoded into a vector of length |E| through one-hot vector encoding, E is the set of exercises, |E| represents the number of exercises, the encoded vector is adjusted by a layer of sigmoid activation function into the concepts representation c_T.

After getting the text representation and concepts representation, we concatenate them together as the final exercise embedded matrix.

$$x_T = b_T \oplus c_T \quad (3)$$

$\oplus$ means concatenate two vectors in a certain dimension, and the length of vector obtained after concatenating is |E|+768.

3.3 Exercises association modeling

After obtaining the final merged exercise embedding matrix, we need to model the entire exercising process of each student and obtain his hidden state at each step, it will be affected by both the history exercises sequence and his responses.

3.3.1 *Student response embedding*

First of all, we combine the student's response with each exercise representation. Specifically, at each step t,we combine exercise embedding x_T with the corresponding score r_T as the input to the recurrent neural network.

We first extend the score r_T to a feature vector $0 = (0, 0, \dots, 0)$ with the same dimensions of exercise embedding x_T and then learn the combined input vector $\widetilde{x_T}$ as:

$$\widetilde{x_T} = \begin{cases} x_T \oplus 0 & r_T = 1 \\ 0 \oplus x_T & r_T = 0 \end{cases} \quad (4)$$

After the concatenating, the student sequence becomes $s = \{\widetilde{x_1}, \widetilde{x_2}, \dots\dots, \widetilde{x_T}\}$.

3.3.2 *One-way time series knowledge tracing*

Like the original DKT method, a hidden layer is applied to track changes in student's knowledge status, the formula is as follows:

$$i_T = \sigma(W_i \cdot [h_{T-1}, \widetilde{x_T}] + b_i) \quad (5),$$

$$f_T = \sigma(W_f \cdot [h_{T-1}, \widetilde{x_T}] + b_f) \quad (6)$$

$$o_T = \sigma(W_o \cdot [h_{T-1}, \widetilde{x_T}] + b_o) \quad (7)$$

$$\widetilde{c_T} = tanh(W_c \cdot [h_{T-1}, \widetilde{x_T}] + b_c) \quad (8)$$

$$c_T = f_T \cdot c_{T-1} + i_T \cdot \widetilde{c_T} \quad (9)$$

$$h_T = o_T \cdot tanh(c_T) \quad (10)$$

Where c_T is the long-term state at time T, i_T, f_T, and o_T are the input gate, forget gate, and output gate in LSTM respectively. tanh represents the tanh activation function, $tanh(z_i) = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i})$, σ represents sigmoid activation function, $\sigma(z_i) = 1/(1 + e^{-z_i})$.

3.3.3 *Bidirectional time series knowledge tracing*

In order to better obtain the association between the exercises, we introduce a bidirectional long and short-term memory neural network [14] to obtain the hidden state representation of the students, because Bi-LSTM can make full use of the exercises representation in both forward and backward directions [15], it can obtain the association between the exercises. Specifically, after getting $s = \{\widetilde{x_1}, \widetilde{x_2}, \dots\dots, \widetilde{x_T}\}$, we set the input of the first layer of LSTM to $\overrightarrow{\boldsymbol{h}}^{(0)} = \overleftarrow{\boldsymbol{h}}^{(0)} = \{\widetilde{x_1}, \widetilde{x_2}, \dots\dots, \widetilde{x_T}\}$, at each time T, the forward hidden state of each layer $(\overrightarrow{h_T^{(l)}}, \overrightarrow{c_T^{(l)}})$ and backward hidden state $(\overleftarrow{h_T^{(l)}}, \overleftarrow{c_T^{(l)}})$ updates with the input from previous layer from each direction. The specific formula is as follows:

$$\overrightarrow{h_T^{(l)}}, \overrightarrow{c_T^{(l)}} = \text{LSTM}\left(\overrightarrow{h_T^{(l-1)}}, \overrightarrow{h_{T-1}^{(l)}}, \overrightarrow{c_{T-1}^{(l)}}; \overrightarrow{\theta}_{\text{LSTM}}\right) \quad (11)$$

$$\overleftarrow{h_T^{(l)}}, \overleftarrow{c_T^{(l)}} = \text{LSTM}\left(\overleftarrow{h_T^{(l-1)}}, \overleftarrow{h_{T+1}^{(l)}}, \overleftarrow{c_{T+1}^{(l)}}; \overleftarrow{\theta}_{LSTM}\right) \quad (12)$$

The association between exercises can be captured by Bi-LSTM. Since the hidden state of each direction only contains the association of one direction, it is beneficial to combine the hidden state of both directions together to obtain the final student hidden state representation:

$$H_T = \text{concatenate}\left(\overrightarrow{h_t^{(L)}}, \overleftarrow{h_t^{(L)}}\right) \quad (13)$$

3.4 Student performance prediction

After the above steps, we get the student's hidden learning state sequence $\{H_1, H_2, \dots\dots H_T\}$ and the exercise sequence $\{x_1, x_2, \dots\dots x_T\}$, both of which will affect the student's final answer. We utilize two layers of bidirectional neural networks to obtain the predicted student performance, as shown in the formula:

$$y_{T+1} = Tanh(W_1 \cdot [H_T \oplus \widetilde{x_{T+1}}] + b_1) \quad (14)$$

$$\widetilde{r_{T+1}} = \sigma(W_2 \cdot y_{T+1} + b_2) \quad (15)$$

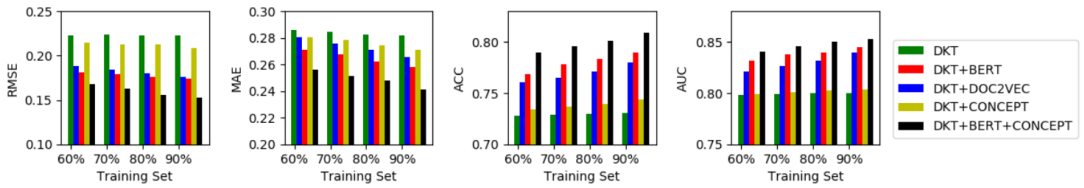

Figure 4. Four indicators performance after adding multi-dimensional feature.

The first layer uses the Tanh activation function, and the second layer uses the sigmoid activation function. After two layers, the final prediction result $\widetilde{r_T}$ is obtained. It is a scalar, which represents the probability of answering the question e_T correctly.

3.5 Training and optimization

The method is optimized by conducting the binary cross-entropy loss function, which calculates the loss between the true response r_T and the probability of correct answer $\widetilde{r_T}$, and adjusts the model parameters such as exercise embedding parameters and student response embedding by inverse transfer until the value converges. The loss function is defined as:

$$\mathcal{L} = -\sum (r_T \log \widetilde{r_T} + (1 - r_T) \log(1 - \widetilde{r_T})) \qquad (16)$$

4. EXPERIMENTS

In order to ensure the reliability of the experimental results, we carry out several baseline comparison experiments on a real dataset. This section will focus on the selection of data set and the comparison of benchmark models, as well as a discussion of the final experiments on a real dataset. This section will focus on the selection of data set and the comparison of benchmark models, as well as a discussion of the final experimental results.

4.1 Dataset

The method we proposed was validated on a dataset called Eanalyst-math. The data comes from a widely used evaluation system in China [16], from offline to online, that selects elementary school math exercises as experimental subjects. The data collected by EAnalyst-math mainly includes homework, unit tests, and term tests. Each assignment or evaluation is regarded as a collection of exercises, which is more in line with the actual education situation in China. EAnalyst-math recorded a total of 525,638 interactions from 1,763 students, with an average of 298.1 responses per student.

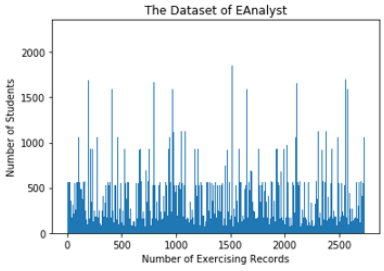

Figure 3. Student-interaction distribution diagram of the Eanalyst-math dataset.

4.2 Settings

In order to predict students' future response, we can evaluate it by classification and regression respectively [17]. From the perspective of classification, the area under the Receiver Operating Characteristic (ROC) curve AUC and the prediction accuracy ACC are used to measure the prediction performance. From a regression point of view, we choose Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to quantify the distance between the predicted result and the actual response.

Each data set is divided into 7:3 based on students, 70% is utilized for training verification, and 30% for testing. In order to avoid the contingency of the evaluation results, we implement the standard five-fold cross-validation division for all models and all training validation subsets, that is, 80% training set and 20% validation set, and exploit the average value as the final comparison result.

There are many hyperparameters in the model, among which the number of hidden units (h), batch_size (b) and learning rate (l) will have a greater impact on the results. We conducted many experiments to explore the influence of the changes of these hyperparameters on the performance of the model, and finally found that the performance results were optimal when l=0.09, h=16, and b=16.

4.3 Results

4.3.1 Accuracy comparison

We compare our ERAKT with three other baselines on the dataset. The experimental results are shown in Table 2. In general, ERAKT has significantly improved AUC and ACC results, MAE and RMSE results are significantly lower, which proves that the performance of ERAKT is better than the others. Especially, our ERAKT performs better the EERNN, a state-of-the-art model which including the exercise content information. Next, the exercise-aware methods (i.e. EERNN and ERAKT) outperform other models that ignore the exercise content (i.e. DKT and DKVMN). This experimental result validates the conclusion of EKT [11].

Table 2. Accuracy comparison

Method	AUC	ACC	MAE	RMSE
DKT	0.79	0.7301	0.2827	0.2233
DKVMN	0.8783	0.8072	0.2678	0.1346
EERNN	0.8836	0.8213	0.2495	0.131
ERAKT	0.9025	0.8407	0.2203	0.1278

4.3.2 The influence of multi-dimensional features of exercises on the prediction results

In view of the fact that multi-dimensional features will affect the performance of knowledge tracing, we explore the impact of different features on the performance of the model by adding them into the model. The results are shown in Table 3. It can be seen that the effect of ERAKT, which integrates multi- dimensional features, is significantly better than other models which only add semantic features or concepts features. It is worth mentioning that all the models we mentioned above perform better than the original DKT model.

Table 3. The influence of multi-dimensional features of exercises on the prediction results

None	DKT	0.79
Semantics	DKT+Doc2Vec	0.8266
	DKT+Bert	0.8325
Concept	DKT+Concept	0.83
Multi-dimensional features	DKT+Bert+Concept	0.8463

From a semantic point of view, no matter which method (Doc2Vec or Bert) is adopted to obtain the exercise representation, the results after embedding the method have been greatly improved, which shows that the text content of the exercises does have a non-negligible impact on the prediction result. From the perspective of concepts, the addition of conceptual features, while not much improved, was still about 1% higher than the original knowledge tracking method.

In order to better eliminate the influence of the data set division ratio on the results, 60%, 70%, 80% and 90% of the data set are applied as the training set, and the rest as the test set. As shown in Figure 4, the result is consistent with the performance of 70% division, with both AUC and ACC increasing and RMSE and MAE decreasing as the data set increases, which proves that the increase in the training set can Enhance forecasting effect.

4.3.3 The impact of association on the prediction results

After the experiment proved that integrating exercises representation can improve the accuracy of prediction, we then designed a comparative experiment to explore the impact of exercises association in predicting.

Table 4. The influence of exercise association on prediction results

RNN	LSTM	GRU	BI-LSTM
0.8463	0.8543	0.8637	0.9065

After we integrate exercises representation, we chose different time series modeling methods to model and predict student's responses. A comparison of the more commonly applied RNN [18], LSTM [19] and GRU [20] with the Bi-LSTM used in our method is shown in table 4. It can be seen that RNN is the worst performer, with LSTM and GRU in the middle, and Bi-LSTM the best. It proves that exercises association has a great influence on the prediction performance.

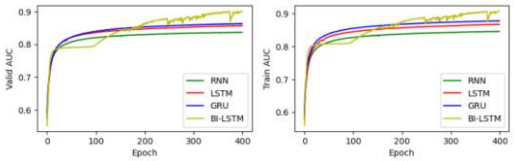

Figure 5. Auc and Loss convergence fluctuation diagram after adding exercise association

We have drawn the model convergence of the four time series modeling methods. It can be seen that the Bi-LSTM fluctuates slightly, and the model convergence curves of the other three methods are relatively smooth.

5. CONCLUSION AND DISCUSSION

In our proposed method, we propose a new context-aware KT method that integrates mathematical exercise representation and association of exercises, through which we can predict the performance of a student on the exercises, thus helping teachers to adjust their teaching plans dynamically.

Experiments have verified the effectiveness and reliability of our method. It can be seen from the experimental results that the prediction results are significantly improved after integrating multi-dimensional features and exercise association.

As for the exercise semantic representation, Bert can obtain more exercise information, which is better than Doc2Vec after integrating. This is because Bert realizes the processing of data of time series through the attention mechanism and it supports parallel computing, which is validated in Bert [8]. In the case of sufficient resources, the computing speed of Bert will be much faster than LSTM, and the residual network which inside of Bert can prevent the network structure from being too complicated. It makes the model perform better.

In the aspect of exercise association, in section 4.3.3, we use four different time series modeling methods, in which the Bi-LSTM exploited in ERAKT method has the best effect, then GRU's performance is relatively well among the remaining three. This is because of the internal structure of them. LSTM and GRU can solve the problem of long-term memory and can avoid the problem of gradient disappearance in RNN. Compared with LSTM, GRU can reduce the risk of over-fitting. Therefore, GRU has the best prediction performance and RNN is the worst, this conclusion can also be obtained in LSTM [19] and GRU [20]. But all of them three can not get the association between exercises.

The bidirectional structure of Bi-LSTM not only preserves the past information, but also the future one. Therefore, all the content can be effectively used to obtain the association between the exercises, which greatly improves the accuracy of prediction.

6. FUTURE WORK

At present, our research has achieved phased results, which can be applied in the actual teaching environment to assist teachers in teaching activities. Our future work will focus on two aspects:

(1) Explore knowledge tracking model that integrates multiple knowledge concepts, and at the same time integrate the sequence association between them.

(2) Show the students' mastery of each knowledge concept systematically. So as to to improve the accuracy of prediction, systematically promote it to facilitate the teaching work of teachers.

7. REFERENCES

[1] Yudelson M V, Koedinger K R, Gordon G J. Individualized bayesian knowledge tracing models[C]//International conference on artificial intelligence in education. Springer, Berlin, Heidelberg, 2013: 171-180.

[2] Piech C, Bassen J, Huang J, et al. Deep knowledge tracing[J]. Advances in neural information processing systems, 2015, 28: 505-513.

[3] Zhang, Jiani & Shi, Xingjian & King, Irwin & Yeung, Dit-Yan. (2017). Dynamic Key-Value Memory Networks for Knowledge Tracing. 765-774. 10.1145/3038912.3052580.

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[5] Le Q, Mikolov T. Distributed representations of sentences and documents[C]//International conference on machine learning. 2014: 1188-1196.

[6] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[7] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.

[8] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[9] VASWANI, Ashish, et al. Attention is all you need. In: Advances in neural information processing systems. 2017. p. 5998-6008.

[10] Su Y, Liu Q, Liu Q, et al. Exercise-enhanced sequential modeling for student performance prediction[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[11] Huang Z, Yin Y, Chen E, et al. Ekt: Exercise-aware knowledge tracing for student performance prediction[J]. IEEE Transactions on Knowledge and Data Engineering, 2019.

[12] Sonkar S, Waters A E, Lan A S, et al. qDKT: Question-centric Deep Knowledge Tracing[J]. arXiv preprint arXiv:2005.12442, 2020.

[13] Ghosh A, Heffernan N, Lan A S. Context-aware attentive knowledge tracing[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 2330-2339.

[14] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015).

[15] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bidirectional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016).

[16] Huang, T., Li, Z., Zhang, H., Yang, H., & Xie, H. (2020). EAnalyst: Toward Understanding Large-scale Educational Data. *EDM*.

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

[18] Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." arXiv preprint arXiv:1409.2329 (2014).

[19] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[20] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[21] S. Wang, J. Tang, Y. Wang and H. Liu, "Exploring Hierarchical Structures for Recommender Systems," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 6, pp. 1022-1035, 1 June 2018, doi: 10.1109/TKDE.2018.2789443.

[22] Tong H , Zhou Y , Wang Z . HGKT : Introducing Problem Schema with Hierarchical Exercise Graph for Knowledge Tracing[J]. 2020

[23] Chen P, Lu Y, Zheng V W, et al. Prerequisite-driven deep knowledge tracing[C]//2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 39-48.

[24] Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. In Proc. International Conference on Educational Data Mining. 384–389.

[25] Nakagawa H , Iwasawa Y , Matsuo Y . Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network[C]// IEEE/WIC/ACM International Conference on Web Intelligence. ACM, 2019.