Rodrigo, M. M. T. et al. (Eds.) (2021). Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

BEKT: Deep Knowledge Tracing with Bidirectional Encoder Representations from Transformers

Zejie Tian[a*], Guangcong Zheng[c], Brendan Flanagan[b], Jiazhi Mi[c] & Hiroaki Ogata[b]
[a]*Graduate School of Informatics, Kyoto University, Japan*
[b]*Academic Center for Computing and Media studies, Kyoto University, Japan*
[c]*School of Computer and Communication Engineering, Northeaster University at Qinhuangdao, China*
*tianzejie123@gmail.com

Abstract: Knowledge tracing is the task of modelling each student's mastery of knowledge components by analysing a student's learning activities trajectories. Each student's knowledge state is modelled based on his or her past learning performance and is an important research area in improving personalized education. In recent years, many researches have focused on deep learning models that aim to solve the knowledge tracing problem. These methods have shown improved performance when compared to traditional knowledge tracing methods such as Bayesian Knowledge Tracing. However, as the input information into the model is a simple representation of the distinction of each student learning logs, the performance of past models are limited and it is hard to measure the relationship between each interaction. To address these problems, we propose the use of a state-of-the-art Bidirectional Encoder Representations from Transformers based model to predict student knowledge state by combining side information such as student historical learning performance. The bidirectional representation can analyse student learning logs in detail and help to understand student learning behaviours. An ablation study is performed to understand the important components of the proposed model and the impact of different input information on model performance. The results of the proposed model evaluation show that it outperforms existing KT methods on a range of datasets.

Keywords: Knowledge tracing, self-attention, BEKT, DKT

1. Introduction

In recent years, with the development of artificial intelligence and technologies, intelligent tutoring systems (ITSs) and massive open online courses (MOOCs) are attracting more and more students to use them to improve their learning performance. These platforms record a massive amount of data sequences of student's learning activities about knowledge components (KCs), which are composed by skill, concept, exercise, etc. The collection of data has attracted researchers to develop a variety educational tools and models to predict the student's knowledge state and produce personalized advice by analysing historical data. Not only can it give teachers insight into performance of a student more comprehensively so as to arrange personalized learning material, but can also let students know their mastery of knowledge.

Knowledge Tracing (KT) can be formalized as follows: depending on student's past performance logs $X = (x_1, x_2, \dots, x_t)$ predicts the next time interaction x_{t+1}. According to question and answer logs, x_t can be represented as a pair of (q_t, a_t), where q_t is the question which the student attempts and a_t is the result of student's answer, namely, whether it is correct or incorrect. KT aims to predict the probability of correctness for a student's next answer, for example: $\mathrm{p}(a_{t+1} = 1 | q_{t+1}, X)$.

Many models have been proposed to solve the KT problem, such as Item Response Theory (IRT) (van der Linden & Hambleton, 2013), Additive Factor Model (Cen, Koedinger, & Junker, 2006), and Performance Factor Analysis (PFA) (Pavlik Jr., Cen, & Koedinger, 2009) which is based on a logistic regression model. Another main kind of model that are popular are Bayesian Knowledge Tracing (BKT) (Corbett & Anderson, 1994) based on a Hidden Markov Model (HMM). More recently,

deep learning based models haven been gaining attention, such as: Deep Knowledge Tracing (DKT) (Piech, et al., 2015). DKT generally outperforms than traditional models, due to the powerful computation ability and the capability of capturing time sequence. Recently, some transformer based models have been proposed, such as: Self-Attentive Knowledge Tracing (SAKT) (Pandey & Karypis, 2019), Attentive Knowledge Tracing (AKT) (Ghosh, Heffernan, & Lan, 2020). These models apply an attention mechanism to deep learning models to improve the interpretability of model and increase performance. However, it should be noted that those models are limited in the utilization of student learning logs and strictly follow the time sequences.

To the best of our knowledge, for fully understanding the student learning performance and updating of knowledge state, combing both past and current data into consideration is needed. This is mainly different to traditional single time sequence model like BKT, DKT etc. In other words, by fully analysing students past learning logs $(x_1, x_2, \dots, x_{t-1})$ can help us better understanding the student knowledge state at $x_{t-n}(n < t)$ rather than only utilizing $(x_1, x_2, \dots x_{t-n-1})$ data for analysis. Besides, properly calculating the relationship between interactions are important. In this paper, we propose a novel approach to solve the knowledge tracing problem, called Bidirectional Encoder Representation of Knowledge Tracing (BEKT) that is inspired by a Bidirectional Encoder Representations from Transformer model (Devlin, Chang, Lee, & Toutanova, 2019) with the aim of understanding a student's knowledge state and increased prediction accuracy when compared to previous models. In our model, we use student's learning activities sequence combining with their historical learning performance to predict the student current knowledge state. We utilize two phases in training the model. The pre-training can help to find proper relationship between student interaction logs and initializing the input representation. Fine-tuning is based on the pre-training for the predication of student current knowledge state. In the experiment, we examine the influence of different input information and show that the proposed method outperforms other KT models.

2. Related Work

As online learning is attracting attention, there is an ongoing open problem of how to effectively guide a student to achieve their learning goals and recommend appropriate learning materials. In the context of student modelling, knowledge tracing tries to measure the student's mastery level of the knowledge. A lot of methods based on different math and psychology models have been proposed to solve this problem.

One of the most popular models is BKT (Corbett & Anderson, 1994) based on hidden Markov models (HMM). The correctness of student answering is calculated by Bayesian formula according to the skills of each exercise. The BKT model updates dynamically based on student's response sequences. The model requires each exercise to be associated with a single skill and it is based on assumptions that the student won't forget and there are no relationships between knowledge components (KCs) which has been highlighted as a problem by previous works, but this shortcoming has been overcome by extension to the BKT model to a certain degree (Hawkins, Heffernan, & Baker, 2014; Käser, Klingler, Schwing, & Gross, 2017; González-Brenes, Huang, & Brusilovsky, 2014).

Learning factor analysis (LFA) (Cen, Koedinger, & Junker, 2006), based on the logistic regression model, is used to measure common factors of a students' ability while trying to deal with multi-KCs problems. According to the logistic regression model, item response theory (IRT) is the simplest model which can measure the distance of each student's ability and difficulty of exercises. Performance factor analysis (PFA) is an adaption of previous models, it believes the correctness of student response should be considered by the model. This model can use the performance of each student on different items to predict the probability that a student has mastered the KCs required by the items. Parameters of these models have highly interpretability but they cannot provide insights on the relationship between different KCs and computational power is limited.

Recently, through the development of deep learning, the Deep Knowledge Tracing (DKT) model was proposed based on a Recurrent Neural Network (RNN) to model a student's knowledge state and the relationship between skills through hidden vectors. Experimental results show that DKT outperforms traditional models like BKT, LFA and do not require complex feature engineering. The input information is simply the sequence of student response for each item, i.e. $x = \{q_t, a_t\}$, where q_t represents the skill index of items, and a_t is the correctness of the response. Due to the structure of the

model, the parameters of the model are hard to interpret and recent research has tried to fix these problems such as: DKVMN (Zhang, Shi, King, & Yeung, 2017). Also, some research has focused on optimizing the quality of the prediction results of DKT (Yeung & Yeung, 2018).

To further extent the interpretability and performance, SAKT (Pandey & Karypis, 2019) first uses the self-attention mechanism to solve KT problem. Attention mechanism has strong and flexible ability to capture the relationship of input information. Later in the work of AKT (Ghosh, Heffernan, & Lan, 2020), this method uses the entire learner's practice history with context-aware representations to fix the limitation of small windows in SAKT. Besides, they propose an adapted attention mechanism which let models gain more interpretability. To the best of our acknowledgement, both of them are using monotonic attention mechanism which may limit model to comprehensively understand student historical learning logs and future performance.

In this paper, we propose a BEKT model based on the structure BERT (Devlin, Chang, Lee, & Toutanova, 2019). Through leveraging the deep bidirectional representation, our model can gain more information from student interaction sequences. Besides, through our experiment, we precisely analyse the influence of different input information and the importance of pre-training. In the following section, we will introduce the proposal and discuss the details.

3. Method

In this paper, we propose the use of BEKT to solve the knowledge tracing problem. The model architecture is a multi-layer bidirectional transformer encoder, with a self-attention mechanism and bidirectional analysis, making the model more powerful to understand the student past learning logs.

There are two steps of BEKT framework: pre-training and fine-tuning. As our intuition: (1) For understanding the student certain learning behaviours, the nearest exercises which are closely to the student current practice are more valuable and relevant than something far away. (2) when analyse the student historical learning logs, both future and past information are helpful. In other words, for analysing the certain student past behaviours, we should utilize the entire student learning logs to gain an overall point of view. Therefore, during the pre-training, we utilize mask mechanism to cover certain student past practice questions and push the model to conclude surrounding learning information and utilize it to guess the masks. It helps the self-attention to notice the time distance between student learning logs and to refine the vectorization of each input information. During fine-tuning phase, the BEKT model parameters are initialized by the pre-training and fine-tuned using the label of correctness for each question answering by students. An overview of the proposed model is shown in Figure 1.

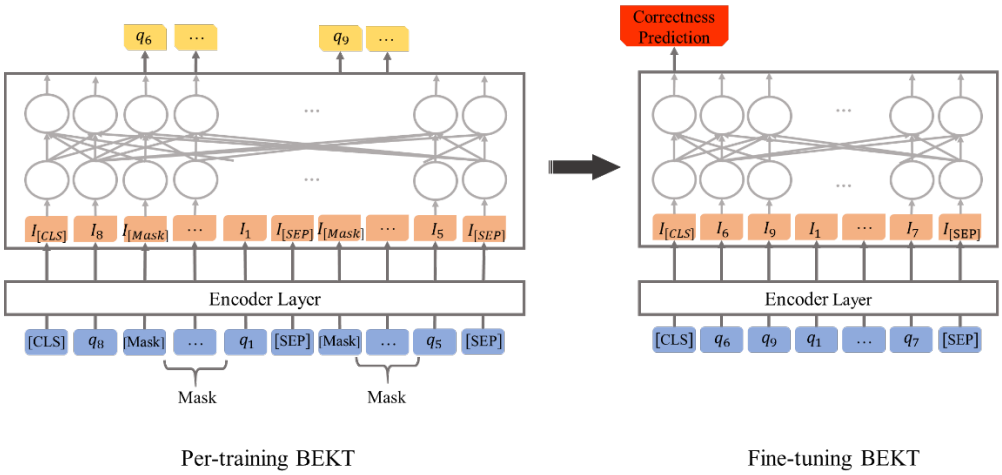


Figure 1. The overlook of BEKT.

We will discuss details of our model in the following parts.

3.1 Input Representations

Our proposed model uses the past learning trajectory of students $X = (I_1, I_2, I_3, \ldots, I_n)$ to predict whether the response to I_n is correct or not. Moreover, the input sequence set I is composed by five embedding parts: position encoding, concepts encoding, question encoding, response encoding and difficulty encoding. Following previous work, we transform each part into real-valued embedding vectors as $E \in \mathbb{R}^d$ and each input pair of time t can be represented as $(q_t, c_t, r_t, d_t, p_t)$. c_t and q_t characterize information about concepts and questions. In most of real-world educational systems, the question sets are generally much larger than concepts and only part of them are assigned to students. Therefore, due to the sparsity of problems, many of existing knowledge tracing models only utilize the concept to represent the student exercise, in this case, $q_t = c_t$. But in our methods, with the help of pre-training, each question can gain more flourish and meaningful information to overcome the sparsity problem. In our experiment, including both questions and concepts can gain more better performance. r_t characterize information about student answering result, which can be represented as two sperate embedding for correct and incorrect answering, respectively. d_t characterize information about difficultly of question, which can be defined as the ratio of correctness for certain question in training dataset. It's an important feature to help distinct the difference of question and overcome the sparsity issue (Zhang, Shi, King, & Yeung, 2017). In our setting, we divided it into five levels according to computed correctness ratio and represent it into different embeddings. p_t characterize information about the order of student activities, because self-attention mechanism is not inherently aware of the time sequence. As for Student's Knowledge states are continually changing follow their interaction with the learning system, the position encoding keeps updating to follow the changing model. Therefore, we utilize the learnable embeddings (Vaswani, et al., 2017) to model the temporal dynamics of attentions.

At the beginning of each sequence, we add the special token [CLS] which is always used to aggregate the information of whole input representation (Devlin, Chang, Lee, & Toutanova, 2019) and also add another special token [SEP] at the end of it to indicate a completed input. Both of them embedding are same with the others which can be denoted as $E \in \mathbb{R}^d$, d is the dimension of these embedding. The final input sequence I shows in Figure 2, The size of I is equal to n, when the length of student interaction logs is smaller than n, we repetitively add paddings to each encoding layers at the end of it.

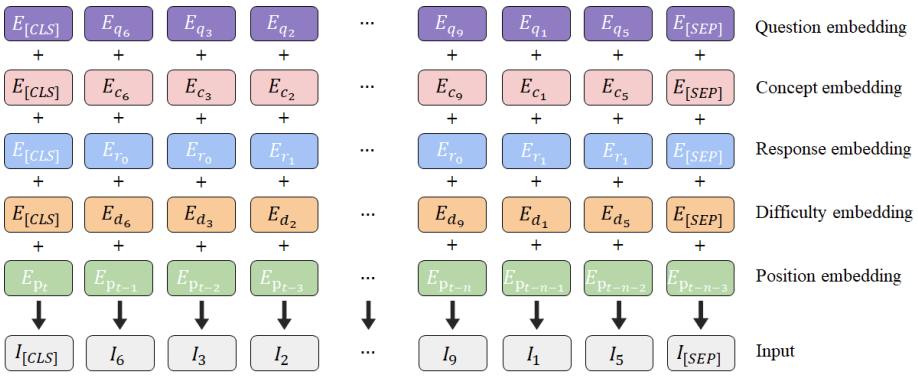


Figure 2. Input Representation.

3.2 Pre-training BEKT

The pre-training is the most important part to accomplish our hypothesis. We utilized the *mask mechanism* to accomplish the bidirectional model. Traditional knowledge tracing model interprets student behaviors from past to current. As our intuition, we could more comprehensively understand student learning activities combining with the experience of past and the next attempts of current, so that our model can more accurately infer student current knowledge state.

Therefore, we automatically mask 15% embedding vectors of each interaction sequence and directly let model to predict the masked questions combining with the whole input information and the correctness of attempts. Within the chosen mask embedding vectors, it has 80% to be replaced by [mask], which can let model use both directional information to analyze without "seeing itself". And 10% to be replaced by a random embedding selected from the "embedding bank", which can add some noise to the model. The left 10% remains current representations, which prevents the model only utilize the surrounding information and neglect itself.

Noticeably, we utilize the multitask learning during pre-training. We only mask the (q_t, c_t, d_t) at certain time t and minimize the average of cross-entropy loss when predicting the mask embeddings of question, concept and difficulty. Each hidden vectors of mask input information will be fed into output layer of SoftMax and to predict the signed label. As we obeserving of experiment datasets, student often continuously attempts some questions of the same concept for learning. Therefore, the mask mechanism can also help to build up the relationship of those question to construct a mix embedding representation of question, which can be an effective way to solve the sparsity problem and improve the model performance.

3.3 Fine-tuning BEKT

In the fine-tuning, the input is a single sequence showing in the right structure of Figure 1.

Our task is to predict the correctness of student attempts, so we change the structure of output representation in Fine-tuning BEKT. The input is composed by the current question embedding, skill embedding and difficulty embedding in the [CLS] position and all the other input remains unchanged. To further analysis the feature of hidden vectors, we add a small residue neural network upon BERT. The output of the [CLS] hidden vector is fed into residue neural network to predict the correctness of the student current attempt I_n.The output is $\hat{r}_t \in [0,1]$ represent the correctness of student answering. The parameters of model are learnt by minimizing the binary cross entropy loss between the student true performance r_t and predict performance $\hat{r}_t$:

$$\mathrm{L} = \sum_t -(r_t log\hat{r}_t + (1 - r_t)\log(1 - \hat{r}_t))$$

Instead of random initializing the BEKT, parameters of fine-tuning are straightforward inherited from the pre-training. In the pre-training, the BEKT gets the knowledge of student learning behaviours and updating of knowledge states. So based on the priori knowledge, the fine-tuning BEKT can more accurately trace the student knowledge states and converge more quickly.

4. Experiment

4.1 Datasets

In order to evaluate our model, we chose 4 widely used real-world datasets showing in table 1.

Table 1. *Datasets Information*

Dataset	Users	Skill tag	Interactions	Mean correctness	Mean student interaction
ASSISTment 2009	3,684	124	336,886	0.654	91
ASSISTment 2015	19,917	100	683,801	0.706	34
ASSISTment challenge	1,709	102	942,816	0.373	552
Statics2011	333	1,223	189,927	0.765	570

ASSISTment 2009[1]: This dataset is collected by the ASSISTment online tutoring systems. According the research (Xiong, Zhao, Van Inwegen, & Beck, 2016), the duplicated problems of old datasets have been found. The new dataset fixes the problems and released by the ASSISTment system. We deleted the none skill name data during pre-processing and length of interaction smaller than 5. The final dataset contains 3,684 students with 336,886 question-answers interactions based on 124 skills.

ASSISTment 2015[2]: This dataset contains 19,917 students' responses for 100 skills with 683,801 interactions. Noticed that we delete the data which correct values are not 0 or 1. Even though it has a lot of interactions of providing by students, but the average interactions per student are pretty low.

[1] https://sites.google.com/site/ASSISTmentdata/home/assistment- 2009-2010-data/skill-builder-data-2009-2010

[2] https://sites.google.com/site/ASSISTmentdata/home/2015- ASSISTment-skill-builder-data

ASSISTment challenge[3]: These datasets are provided by the 2017 ASSISTment data mining competition. There are 1,709 students with 942,816 interactions and 102 skills.

Statics2011 (Zhang, Shi, King, & Yeung, 2017): The data is collected from engineering statics courses containing 333 students, 1,223 skill tags and 189,927 interaction pairs. We combine the problem name and step name together as an exercise tags, therefore it has maximum number of exercise tags comparing to other datasets.

4.2 Evaluation Methodology

In the experiment, we chose to compare the proposed model with three state-of-the-art Knowledge tracing models on four different publicly available educational datasets. The details of these models are described in the related work.

To measure the accuracy and generalizability of the proposed model when compare to the three state-of-the-art models, we apply 5-fold cross-validation at the student level in our experiments, which means that the students are split randomly into 5 distinct groups and the training and validation sets selected from these groups. For the measurement of model prediction performance, we calculate the Area Under the receiver operator characteristic Curve (AUC) as described by Fawcett (2006).

Our model was implemented in Python using the Pytorch framework. The parameters of model are initialized from a Gaussian distribution with zero mean and standard deviation δ. The ADAM optimizer is used to optimize the model parameters with the learning rate of 5^{e-4} for the pretrain, 2^{e-4} for the fine-tuning and a batch size of 128 for each dataset. We choose different length of sequence according to the average length of student interactions, so we set n=80 for ASSISTment challenge and Statics2011 and n=50 for ASSISTment2009 and ASSISTment 2015. For the hyper- parameters of BEKT, we set up with 8 Transformer blocks, 8 self-attention heads and the hidden size of 256. We set of two layers small residue network for fine-tuning correctness predictions with size 512 and 256 separately and activation uses ReLU.

5. Results and Discussion

5.1 Student Performance Prediction

To show the effectiveness of the proposed model, we measure the prediction performance of the models using AUC. Higher AUC measurements represent greater model accuracy. Results of test AUC on different datasets are shown in Table 2, with the best prediction performance for each dataset shown in bold.

Table 2. *Student Performance Prediction Comparison*

Dataset	AUC							
	DKT	DKT+	DKVMN	SAKT	AKT-NR	AKT	BEKT-NS	BEKT
ASSISTment 2009	0.7547	0.7221	0.7456	0.7268	0.7494	0.7594	0.7549	**0.8227**
ASSISTment 2015	0.7125	0.7049	0.6970	0.6875	-	0.7015	0.7139	**0.7167**
ASSISTmentChall	0.7182	0.7163	0.6568	0.7071	0.7159	0.7536	0.7141	**0.7746**
Statics2011	0.8071	0.8027	0.8002	0.7773	-	0.8189	0.8254	**0.8307**

We compare the BEKT model with the state-of-art DKT, the optimal variations of DKT(DKT+), DKVMN model, SAKT model and AKT model. AKT-NR means the AKT model only utilizes only skill input information without considering the difference between question. In other word, in this case, $q_t = c_t$. BEKT-NS means that BEKT only utilizes the skill information as with other models and BEKT is our full model as discussed in the model section. We implemented the same data pre-processing as BEKT on their models. Overall, BEKT outperforms previous knowledge tracing models.

[3] https://sites.google.com/view/ASSISTmentdatamining

For the ASSISTment 2009 data, the test AUC of BEKT is 0.8227 which is better than 0.7547 of DKT, 0.7221 of DKT+ and 0.7456 of DKVMN, 0.7268 of SAKT, 75.94% of AKT. On the ASSISTment 2015 dataset which student average interaction length is shortest, therefore it can't show the integration power of our model sufficiently. BEKT gains small improvement over DKT, DKT+, DKVMN, SAKT, AKT with 0.7167 over 0.7125, 0.7049, 0.6970, 0.6875, 0.7015 respectively. With regard to Statics2011, which has the maximum number of student average interaction length, BEKT improvements of 0.118 than the best state-of-art model AKT. For ASSISTment challenge, BEKT achieves improvement over DKT, DKT+ and DKVMN, SAKT, AKT with 0.7182, 0.7163, 0.6568, 0.7071, 0.7536 respectively.

For equally comparing purpose, we also conduct the additional experiment which utilize only skills without any other side information as the others model implemented. It should notice that in this case, we should utilize the AKT-NR for comparing purpose. ASSISTment 2015 and Statics2011 dataset only contain the skill information, so there is no difference between AKT-NR and AKT on these two datasets. According to the result, by comparing BEKT-NS with other models on all datasets, our model still outperforms than others except in ASSTIment Challenge dataset.

In summary, BEKT outperforms than other methods across all the datasets. This result demonstrates that the pre-training can help model comprehensive understanding of student knowledge states and the side information as difference of question and their difficulty can further improve the model performance.

5.2 Ablation Study

In order to understand the importance of different parts of the proposed BEKT model contribute to the performance, we conducted an ablation study (Pandey & Karypis, 2019). There are several steps and parts of the input representation that could contribute more to the overall performance. Table 3 shows the performance of different parts of BEKT on all the datasets without question difficulty information. We utilize the Q to represent question, C for the concept, R for the result, P for the position information. For example, BEKT-QCRP means the input representation in certain time t is (q_t, c_t, r_t, p_t). Results for ASSISTment 2015 and Statics2011 were omitted because the datasets do not contain question information.

Table 3. *Student Performance Prediction Comparison*

Architecture	ASSISTment 2009		ASSISTment 2015		ASSISTment challenge		Statics2011	
	NP	P	NP	P	NP	P	NP	P
BEKT-QCRP	0.7602	**0.7666**	-	-	0.7652	**0.7720**	-	-
BEKT-QRP	0.7264	0.7647	-	-	0.7679	0.7710	-	-
BEKT-CRP	0.7538	0.7549	0.7107	**0.7139**	0.7135	0.7141	0.8121	**0.8254**

Pretraining: In the Table 3, we compare the BEKT model with pre-training (P) and without pre-training (NP) result in a different situation. The result shows that model with pre-training outperforms than directly train the model. Pretraining does help BEKT understanding student interactions and changing of knowledge states. Because the information provided by per student is different according to their learning logs, so the ASSISTment challenge and Statics2011 gain more promotion than the other two datasets.

Question: By comparing the result of BEKT-QCRP and BEKT-CRP, model with question information can gain more prosperous information and performance. In our experiment, the ASSISTment2009 dataset contains 15925 number of questions and ASSISTment Challenge contains 1183 number of questions. Therefore, ASSISTment 2009 dataset questions are much sparser than ASSISTment Challenge. From the result BEKT-QCRP, we can see the pre-training can effectively solve the sparsity problem.

Table 4 shows the influence of question difficulty. The question difficulties indicate the student average performance on measuring the difference of question. Comparing with the BEKT contains the question difficulties (BEKT-D) and BEKT without question difficulties (BEKT-ND), it clearly shows that question difficulties can effectively improve model performance. Especially, in ASSISTment2009, it improves the AUC of the model by 0.561, which could indicate it can also help to solve the sparsity problems.

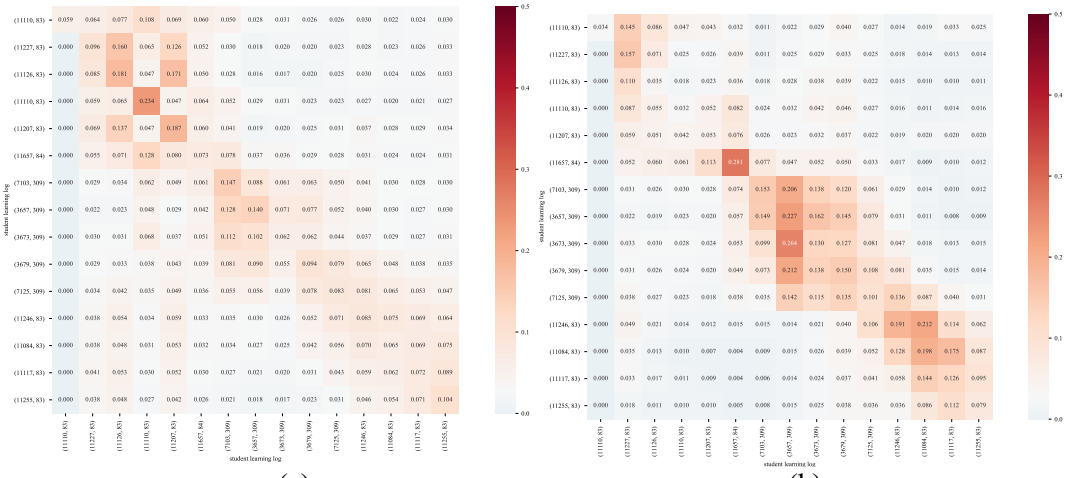
Table 4. *Ablation Study of BEKT with Difficulties*

Architecture	ASSISTment 2009		ASSISTment 2015		ASSISTment challenge		Statics2011	
	NP	P	NP	P	NP	P	NP	P
BEKT-D	0.8154	**0.8227**	0.7117	**0.7167**	0.7693	**0.7746**	0.8244	**0.8307**
BEKT-ND	0.7602	0.7666	0.7107	0.7139	0.7652	0.7720	0.8121	0.8254

In conclusion, with the pre-training BEKT can get more comprehensive view of student behaviours, updating of knowledge state and solve the sparsity problem of questions. The different side information can provide more detail information to help model improving the performance.

5.3 Attention Weights Visualizing

Figure 3 shows the visualizing of attention mechanism using ASSISTment2009 dataset of certain student learning logs.



(a) (b)

Figure 3. Visualization of attention weights
(a) input without concept information (b) input with concept information.

We use the relevance matrix to show the influence between student interactions. Each interaction is represented as (q_t, c_t). There are three concepts included in the student interactions: 83 (Divisibility Rules), 84 (Prime Number), 309 (Order of Operations +,-,/,* () positive reals). We conduct two different experiment to see the influence weight between student interaction sequence. Due to the leftist interaction is what the model will predict, it can't be seen by other interactions. Therefore, the first column is masked as 0 weight. In the figure 3(a), it shows the weight relationship without knowing the concrete concept of each question. We can see three box area inside graph which indicates that our model can capture the concept information of each question automatically. In other words, the question within each box may have some kind of relationship and influence to each other. It can also capture the latent relationship between concepts like model perceives that the question belongs to concept 83 and concept 84 has some correlation. Intuitively, the concept 83 and 84 do have some relationship for the division property. For comparison, we also show the relationship between student interaction when giving the concept information in figure 3(b). It can capture more accurate relationship then only giving question information and it can also capture the latent relationship between concept. From both figure 3(a) and 3(b), we can see clearly diagonal which shows our first hypothesis that *the nearest exercises which are closely to the student current practice are more valuable and relevant than something far away.* For the MOOC platforms, it may not available to assign the concept to each question like showing as 3(a). But our model can help to infer the concepts to each question and keep highly accuracy.

6. Conclusions and Future Work

In this paper, we propose a new model BEKT, which utilizes the student historical learning logs and initializes the model using two tasks: pre-training and fine-tuning. Each students past learning information can help the model not only trace the student knowledge state, but also understand the distinction between students. The pre-training with bi-directional deep representation helps the model more comprehensively understand student knowledge states and learning behaviours as shown by an ablation study. Experiments on different datasets show that our model outperforms than state-of-the-art models. However, there are still some limitations of the BEKT that should be addressed. For example, the pre-training phase has a high computation cost to train even though fine-tuning phase of the model can converge sooner. Different skill embedding dimensions of datasets may also influence the performance of model, and further investigation is needed to prove the current assumption.

For future work, we will try to develop more a concise pre-training task to speed up the whole process. Also, we will incorporate side information and time windows to further improve the input representation and comprehension of the model. Besides simply add all encoding parts together, we will also particularly investigate the influential of each part and find an appropriate way to compose them together.

Acknowledgements

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (B) 20H01722, JSPS Grant-in-Aid for Scientific Research (Exploratory) 21K19824, JSPS Grant-in-Aid for Scientific Research (S) 16H06304 and NEDO JPNP20006 and JPNP18013.

References

Cen, H., Koedinger, K., & Junker, B. (2006, June). Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems* (pp. 164-175). Springer, Berlin, Heidelberg.

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

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171-4186).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.

Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2330-2339).

González-Brenes, J., Huang, Y., & Brusilovsky, P. (2014). General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th international conference on educational data mining* (pp. 84-91). University of Pittsburgh.

Hawkins, W. J., Heffernan, N. T., & Baker, R. S. (2014, June). Learning Bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In *International Conference on Intelligent Tutoring Systems* (pp. 150-155). Springer, Cham.

Käser, T., Klingler, S., Schwing, A. G., & Gross, M. (2017). Dynamic Bayesian networks for student modeling. *IEEE Transactions on Learning Technologies, 10*(4), 450-462.

van der Linden, W. J., & Hambleton, R. K. (Eds.). (2013). *Handbook of modern item response theory*. Springer Science & Business Media.

Pandey, S., & Karypis, G. (2019, January). A self-attentive model for knowledge tracing. In *12th International Conference on Educational Data Mining, EDM 2019* (pp. 384-389).

Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *Artificial Intelligence in Education* (pp. 531-538).

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep Knowledge Tracing. *Advances in Neural Information Processing Systems*, *28*, 505-513.

Xiong, X., Zhao, S., Van Inwegen, E. G., & Beck, J. E. (2016). Going deeper with deep knowledge tracing. *International Educational Data Mining Society*.

Yeung, C. K., & Yeung, D. Y. (2018, June). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* (pp. 1-10).

Zhang, J., Shi, X., King, I., & Yeung, D. Y. (2017, April). Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web* (pp. 765-774).

Liu, Y., Yang, Y., Chen, X., Shen, J., Zhang, H., & Yu, Y. (2020). Improving knowledge tracing via pre-training question embeddings. *arXiv preprint arXiv:2012.05031*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in *neural information processing systems* (pp. 5998-6008).