SQKT: A Student Attention-Based and Question-Aware Model for Knowledge Tracing

Qize Xie[1], Liping Wang[1(✉)], Peidong Song[1], and Xuemin Lin[1,2]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
{51194501074,51205902112}@stu.ecnu.edu.cn, lipingwang@sei.ecnu.edu.cn, lxue@cse.unsw.edu.au
[2] The University of New South Wales, Sydney, Australia

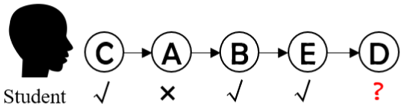
Abstract. The goal of Knowledge Tracing (KT) is to trace student's knowledge states in relation to different knowledge concepts and make prediction of student's performance on new exercises. With the growing number of online learning platforms, personalized learning is more and more urgently required. As a result, KT has been widely explored for recent decades. Traditional machine learning based methods and Deep Neural Network based methods have been constantly introduced for improving prediction accuracy of KT models and have achieved some positive results. However, there are still some challenges for KT research, such as information representation of high-dimentional question data, consideration of personalized learning ability, and so on. In this paper we propose a novel Student attention-based and Question-aware model for KT (SQKT), which can address the challenges by estimating student attention on different type of questions through history exercise trajectory. Firstly, we devise a weighted graph and propose a weighted deepwalk method to get the question embedding which is combined with the correlated skills as question representation. Secondly, we propose a novel student attention mechanism, which is dedicated for the updating of student's knowledge state. Finally, comprehensive experiments are conducted on 4 real world datasets, the results demonstrate that our SQKT model outperforms the state-of-the-art KT models on all datasets.

Keywords: Knowledge Tracing · Deep learning · Graph embedding · Attention-based model

1 Introduction

Knowledge Tracing (KT) [5] aims to estimate student's mastery of knowledge and predict student's future performance, which is a combination of artificial intelligence (AI) and education. As KT is one of the basic techniques for student behavior analysis, it can be widely used for knowledge recommendation, personalized learning path generation and learning evaluation, etc. Recently, with the

L. H. U et al. (Eds.): APWeb-WAIM 2021, LNCS 12859, pp. 221–236, 2021.
https://doi.org/10.1007/978-3-030-85899-5_17



Question	Skill
A	Probability, Function
B	Function, Inequality
C	Function, Inequality
D	Function, Monotonicity
E	Function, Inequality, Monotonicity

Fig. 1. A simple example of knowledge tracing process. Left shows a the exercising records of a student, where he has done question A, B, C and E, the right box shows the corresponding skills of each question, knowledge tracing is used to predict his performance on the new coming question D.

popularity of various online learning platforms, personalized learning is more and more urgently required. As a result, KT has attracted wide attention from related researchers for recent decades.

Generally, the data for KT mostly comes from student's behaviors on the online learning platforms, which contain the questions, responses, timestamps, etc. The questions are usually tagged to skills which is introduced to better represent the knowledge concepts, as is shown in Fig. 1. The algorithm of KT would utilize student's history behaviors and the info or structures about skills for study to predict student's future performance. During early-stage, the traditional machine learning methods is devised for KT. Representative work is Bayesian Knowledge Tracing (BKT) [5] which models knowledge states as a set of binaries, each representing the student's mastery of a single knowledge concept. In recent years, the Deep Neural Network (DNN) [21]-based methods is widely explored. Long short-term memory (LSTM) [8], as its sensitivity for time sequence, has been successfully introduced to update knowledge state at each timestamp. Moreover, skills and their relationships can be modeled as graph and Graph Neural Network (GNN) [24] based-methods is devised to aggregate the student's knowledge state of related skills.

Although the combination with LSTM and GNN has make KT more effective and accurate, challenges for KT research still remain: 1) Due to the high dimension and sparsity of questions data, most of existing methods only use related skills to represent a question. To a certain extent, skills can roughly replace questions for its closer relevance to Knowledge Concepts (KC), and the skills-based methods have achieved a fine empirical performance. However, the abandon of characteristics of questions may cause much information loss and performance degrade. For instance, in Fig. 1, question B and C have the same skills, but they are 2 totally different questions. Therefore, the feature extraction and utilization of questions is very important. 2) The existing KT models lack the ability to trace the latent variation of student's knowledge state. Either a set of binaries or a memory matrix can not fully represent the knowledge states of a student. We noticed that student havs attention when doing exercises, keep practicing on same-type questions can make student more concentrated on the type of questions. 3) The existing GNN based methods have a high dependence on dataset, thus lack of scalability.

In this paper, we devise a novel knowledge tracing model to address the above challenges. Specifically, our model provides a graph-based embedding method for feature extraction and question representation, which can consider comprehensive info of student behaviors on various questions. Additionally, we propose a novel attention mechanism to estimate the student's learning ability on different knowledge concepts and this attention mechanism is dedicated for the updating of student's knowledge state. Our main contributions are summarized as follows:

1) To comprehensively represent the questions, we devise a weighted graph, propose a weighted deepwalk method to get the question embedding and combine it with the correlated skills as question representation. Our question representation can catch the latent relevance while solve the high dimension problem.
2) To enhance the ability of tracing the latent variation of student's knowledge state, we propose a student attention mechanism to add an attention weight when updating the knowledge state. Our student attention mechanism can cooperate with the traditional attention methods well.
3) Extensive and comprehensive experiments are conducted on 4 real world datasets, the experimental results demonstrate the effectiveness of proposed SQKT model. And the comparison to the state-of-the-art KT methods shows that our model achieves higher prediction accuracy.

2 Related Work

In this section, we introduce the progress of the development of Knowledge Tracing methods.

Traditional Knowledge Tracing Methods. Traditional machine learning methods always use logistic regression to classify the questions and skills by regarding each question or skill as a binary variable thus can signify whether the student has mastered the skill or not. Bayesian Knowledge Tracing (BKT) [5] is probably the most popular model in traditional knowledge tracing methods, which update the knowledge state for each student through a Hidden Markov Model (HMM). Based on the BKT model, Pardos et al. [18] introduced the item difficulty to the knowledge tracing model, and Baker et al. [2] utilized contextual estimation of slip and guess probabilities to improve the accuracy. Student individualization is also modeled as an implementation in IBKT [28] and MIBKT [17,28]. Factor Analysis models aim to learn common relations between different features such as (user, skills) pair, and use these common factors as predictors in logistic regression. E.g. Item Response Theory (IRT) [7] model simply use the difference between the mastery degree of student and the difficulty of skill. Multi-dimensional Item Response Theory (MIRT) [6] model has extended the IRT model to multidimensional abilities. Additive factor model (AFM) [3] has taken the student's number of attempts into account, on the basis of AFM, Pavlik et al. propose Performance Factors Analysis (PFA) [19] model which utilizes different bias for the number of the successful and failed attempts.

The methods of logistic regression have strong interpretability and expansibility and the traditional KT models based on BKT [5] have performed reasonably well. However, the explosion of educational data in recent times naturally benefited the deep neural network (DNN) models.

Deep Neural Network. Deep Knowledge Tracing (DKT) [21] first applies deep neural network in knowledge tracing, which utilizes a Recurrent Neural Network (RNN) [27] for KT that can extract the variation of the knowledge state from student's past learning history. Dynamic Student Classification Memory Networks (DSCMN) [12] model, as an extension of DKT, takes the side information of question difficulty into consideration. Dynamic Key-Value Memory Networks (DKVMN) [29] model proposes a Memory-Augmented Neural Network (MANN) [23] instead of traditional RNN. On the basis of DKVMN, Sequential Key-Value Memory Networks (SKVMN) [1] model uses a Hop-LSTM layer that can jump ahead in a sequence of related history records when training. Self-Attentive Knowledge Tracing (SAKT) [15] model utilizes the relevance of past interactions as attention for high-performance in sparse data. Relation-aware self-attention for Knowledge Tracing (RKT) [16] model takes the time interval between two interactions into account to improve the accuracy. Exercise-aware Knowledge Tracing (EKT) [9,25] framework proposes a EERNNA model which uses a bi-directional LSTM to learn the hidden word state of questions in order to distinguish different questions.

With the development of Graph Neural Network (GNN) [24], some GNN-based methods are proposed. Graph-based Knowledge Tracing (GKT) [14] method structures a graph to represent skills and uses GNN to aggregate the student's knowledge state of related skills. On the basis of EKT, Hierarchical exercise Graph for Knowledge Tracing (HGKT) [26] model utilizes a hierarchical graph to tackle with the question representation problem. Both of the two models use the text of the questions while no public dataset contains these text. Therefore, these two models can only test their effectiveness on specific datasets, which means the methods are not universally adaptable.

The SQKT model proposed in this paper differs from all models above, which uses a Weighted Graph Neural Network to represent the high-dimension question data and adds a global attention mechanism to focus on both student attention and question attention. To the best of knowledge, our SQKT model is the first work to propose the idea about weighted graph embedding and student attention mechanism.

3 Problem Formulation

In an Interactive Educational System (IES) with $|S|$ students and $|Q|$ questions, each question contains one or more knowledge skills, every interaction of student will be recorded, our goal is to trace student's knowledge state based on his history records.

Here we denote the history records of one student as $R_s = \{(q_1, a_1, t_1), (q_2, a_2, t_2), ..., (q_N, a_N, t_N)\}, s \in S$, where $q_n \in Q$ represents the n-th question in the history record of student s , $a_n \in (0,1)$ represents the correctness, if the student answers correctly, a_n equals to 1, else a_n equals to 0, and t_n represents the timestamp when student answers the question. To trace student's mastery of each knowledge unit, knowledge skills are used to represent knowledge units. The knowledge skills that are included by the questions was counted by the online educational platform. Each question q_n can contain one or more corresponding knowledge skills $s_1, s_2, ...s_k$, while a knowledge skill can be included by many questions. Generally, the amount of knowledge skills is far less than the amount of questions.

Based on the above description, the problem about KT can be formally defined as follows: given the history record of a student $R_s = \{(q_1, a_1, t_1), (q_2, a_2, t_2), ..., (q_{n-1}, a_{n-1}, t_{n-1})\}$ and the knowledge skills related to each question $S_q = s_1, s_2, ..., s_k$, our goal is to trace student's mastery of knowledge and predict whether the student can answer the coming question q_n correctly.

4 The SQKT Method

In this section, we introduce the specific improvements of our SQKT model. The overall framework is shown in Fig. 2. We first construct a weighted graph by the relationship of questions, then use weighted-deepwalk to learn question representations. After get the question representations, we use Recurrent Neural Network (RNN) [27] with both student attention and question attention to update the knowledge state of the student and to predict his performance on the coming question. Here we just explain the main idea of the model, the detail about question representation and student attention mechanism is described in Sect. 4.1 and Sect. 4.2.

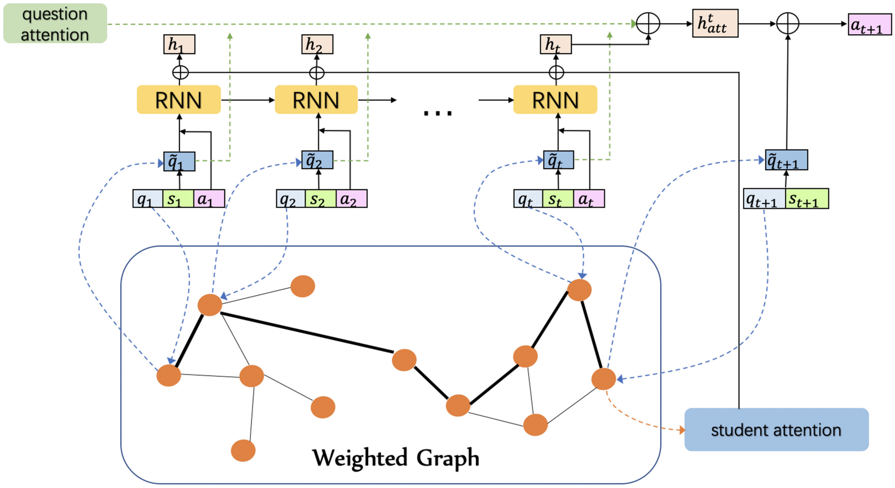
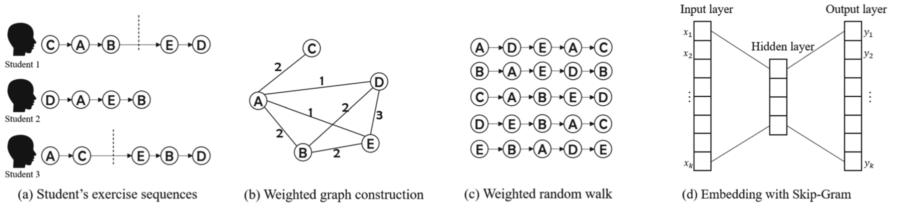


Fig. 2. An illustration of SQKT, which use a weighted graph and a Recurrent Neural Network (RNN) with question and student attention mechanism to get prediction.

4.1 Question Representation

From the perspective of pedagogy, whether a student can answer a question correctly depends on both the question and the student's ability. For question representation, we not only use the related skills, but also focus on the latent relationship that can not be represented by skills. To catch the unique features of each question, we construct a weighted graph $G = (V, E)$ that shows the latent relevance between questions. In the weighted graph, each node represents a question, when question q_i and q_j follows $|t_i - t_j| < T$, we add 1 on the weight w_{ij} of edge e_{ij} between node v_i and node v_j. Figure 3 shows the overview of question representation process.



(a) Student's exercise sequences (b) Weighted graph construction (c) Weighted random walk (d) Embedding with Skip-Gram

Fig. 3. The overview of question representation process: (a) Students' exercise records, the dashed line means the time span of two exercise exceeded the threshold; these records are used to construct the weighted graph; (b) The weighted graph, where the number on the edges represents the weights; (c) The sequences generated from the weighted graph, the larger the weight, the more likely the edge will be chosen; (d) Use Skip-Gram algorithm to get question embedding

In the weighted graph, nodes represent questions and the weight of edges represent the correlation degree between the nodes at both ends. Improved on the basis of DeepWalk [20], we use a weighted deepwalk method to get the structural characterization of our weighted graph. We take each node as a starting point for random walk with the transition probability defined as:

$$p(v_i|v_j) = \frac{w_{ij}}{\sum_{k \in N_i} w_{ik}} \tag{1}$$

After generating the question sequences by random walk, we utilize the Skip-Gram [10,11] algorithm to learn the embeddings, which maximizes the co-occurence probability of two questions in an obtained sequence. The optimization goal is as follow:

$$\underset{\Phi}{minimize} - log \prod_{j=i-s, j \neq i}^{i+s} \Pr\left(v_j \mid \Phi\left(v_i\right)\right) \tag{2}$$

where s is the window size of the context questions in the sequences.

The embeddings of nodes in the weighted graph can reflect the latent relevance between questions, for each interaction at timestamp t, we concatenate the node embedding $\widetilde{\mathbf{q}}_t$ with the one-hot encoding of related skills $\mathbf{s}_t$ and project to d-dimension through a non-linear transformation as complete question representation:

$$\mathbf{q}_t = \mathrm{ReLU}\left(\mathbf{W}\left([\widetilde{\mathbf{q}}_t, \mathbf{s}_t]\right) + \mathbf{b}\right) \tag{3}$$

4.2 Student Attention Mechanism

Learning is a very complicated process. During the process of education, educators always divide the questions into lectures and teach systematically. Generalized by experience, keeping practice on questions of same lecture can be more effective than picking up questions randomly. Therefore, we assume that the learner's absorption of knowledge is based on his attention which generated from his history exercise record in a period of time. The devise of the student attention mechanism can guarantee that learners whose attention is on the same question type can absorb more knowledge than those who are not.

We first choose a hyper parameter T as the time threshold, at each timestamp t_{n+1}, the history question record $q_k \in R_s$ would be regarded as an influence to student$s$'s attention if $|t_{n+1} - t_k| < T$. The influence of history record on student's current attention is related to the time gap, the shorter time gap is, the more influence it will have. We use the following formulation to measure the extent of k-th history record's influence on student's current attention:

$$E_k = \mathrm{RelU}(\mathbf{W}\frac{1}{t_{n+1} - t_k} + \mathbf{b}) \tag{4}$$

where E_k presents the influence extent of k-th history record on student attention. Then we add the influence of all eligible history record with the coefficient of its influence extent to get student$s$'s current attention:

$$Att_s^t = \sum_{t_i > t-T} E_i * q_i \tag{5}$$

where t is the current timestamp and q_i can be calculated by Eq. (3).

Finally we use the cosine similarity between student's current attention Att_s^t and current question q_t as attention weight to measure his absorption of the question when updating knowledge state:

$$W_{att}^t = cos(Att_s^t, q_t) \tag{6}$$

As is shown in Fig. 4, orange nodes present questions from lecture A, green nodes present questions from lecture B, red nodes present student's attention. The blue thick line depicts student's exercise sequence while the red dotted line depicts student's attention sequence calculated by the equations above. When student transits from lecture A to lecture B when doing question 3, the attention weight W_{att}^3 declines correspondingly.

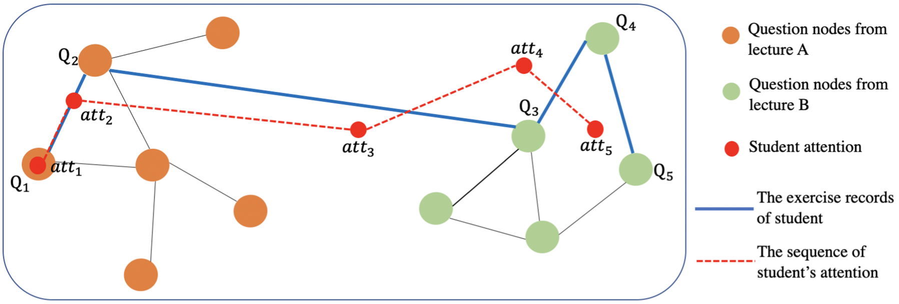

Q$_2$
Q$_4$
att$_4$
att$_2$
Q$_3$
att$_3$
att$_5$
att$_1$
Q$_1$
Q$_5$
Question nodes from lecture A
Question nodes from lecture B
Student attention
The exercise records of student
The sequence of student's attention

Fig. 4. Student attention sequence generated from his exercise records.

4.3 Modeling Process of SQKT

In this section, we will systematically elaborate SQKT modeling process. SQKT use weighted graph to better represent the questions, trace and update student knowledge state by RNN with both student attention and question attention mechanism.

Question-Answer Embedding. In SQKT model, we maintain a weighted graph which represent the latent relationship of questions, and use a weighted deepwalk method with Skip-Gram [11] algorithm to get the question embedding. When student has done a new question at timestamp t, the triplet (q_t, s_t, a_t) would be generated, we get the question embedding Q_t with dimension d_v from q_t and s_t through the weighted graph, extent the embedding vector to dimension $2d_v$ through a_t:

$$\widetilde{Q}_t = \begin{cases} [Q_t \oplus \mathbf{0}] & \text{if} \quad a_t = 1 \\ [\mathbf{0} \oplus Q_t] & \text{if} \quad a_t = 0 \end{cases} \tag{7}$$

where $\mathbf{0} = (0, 0, ..., 0)$ is a vector of all zeros with dimension d_v and $\oplus$ means concatenate, the embedding vector $\widetilde{Q}_t$ is the question-answer embedding which represent the complete triplet (q_t, s_t, a_t).

Knowledge State Evolution. After we get the question-answer embedding $\widetilde{Q}_t$, we use LSTM [8] to trace the knowledge state of student:

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i\left[\widetilde{Q}_t \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right] + \mathbf{b}_i\right) \tag{8}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f\left[\widetilde{Q}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right] + \mathbf{b}_f\right) \tag{9}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o\left[\widetilde{Q}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right] + \mathbf{b}_o\right) \tag{10}$$

$$\mathbf{c}_t = \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \tanh\left(\mathbf{W}_c\left[\widetilde{Q}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_c\right) \tag{11}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh\left(\mathbf{c}_t\right) \tag{12}$$

where i_t, o_t, f_t, c_t, h_t represents input gate, output gate, forget gate, cell state, hidden state respectively.

We introduce the concept of student attention, which can measure student's absorption of knowledge state. Using student attention when updating knowledge state, the Eq. (12) can be updated to:

$$\mathbf{h}_t = W_{att}^t \mathbf{o}_t \tanh(\mathbf{c}_t) \tag{13}$$

where W_{att}^t is the attention weight calculated by Eq. (6)

Prediction Output. Through markov property, we use student's current knowledge state h_t to predict whether he can answer question q_{t+1} correct or not, the prediction probability can be calculated as follow:

$$y_{T+1} = \mathrm{ReLU}\left(\mathbf{W_1} \cdot [h_T \oplus x_{T+1}] + \mathbf{b_1}\right) \tag{14}$$

where W_1, b_1 are parameters and $\oplus$ is concatenation operation.

Note that questions have attentions too and students may get similar score on similar questions. We consider the knowledge state h_t as a weighted sum aggregation of history questions based on its similarity with current question:

$$h_{att}^T = \sum_{i=1}^{T} \alpha_i h_i \tag{15}$$

where $\alpha_i = cos(x_{T+1}, x_i)$. After obtaining the attention mechanism, Eq. (14) can replace the h_t with h_{att}^t:

$$y_{T+1} = \mathrm{ReLU}\left(\mathbf{W_1} \cdot \left[h_{att}^T \oplus x_{T+1}\right] + \mathbf{b_1}\right) \tag{16}$$

We use the Sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ to normalize the result as prediction probability:

$$\tilde{y}_{T+1} = \sigma\left(\mathbf{W}_2 \cdot y_{T+1} + \mathbf{b}_2\right) \tag{17}$$

The student's answer to this question will be predicted to be correct if $\tilde{y}_{T+1} > 0.5$, else will be predicted to be wrong.

4.4 Optimization

We use gradient decent to optimize the parameters in our model. The overall loss can be formulated as:

$$\mathcal{L} = -\sum_{t=1}^{T} \left(a_t \log \widetilde{y}_t + (1 - a_t) \log (1 - \widetilde{y}_t)\right) \tag{18}$$

where a_t is the actual binary score, while $\widetilde{y}_t$ is our predicted score.

5 Experiments

In this section, we conduct several experiments to evaluate the performance of our model on the following aspects: 1) The accuracy of prediction comparison between SQKT and the other baseline models. 2) The representation ability of proposed question embedding method based on weighted graph. 3) The effectiveness of our student attention mechanism.

5.1 Datasets

To evaluate the prediction accuracy, we test the proposed SQKT model and other baseline methods on 4 real world datasets. The datasets were carefully selected that comprehensively covers mathematics, programming and many other fields.

Mynereus[1] is a dataset collected from Mynereus programming Platform, with a total of 86772 records from 202 students on 184 questions. There are 48 skills about these questions.

ASSISTments2009[2] is a dataset collected from the ASSISTments online tutoring platform during the school year 2009–2010. Due to the duplicated record problem, we removed the duplicated records and the rest dataset has 4151 students with 110 questions on 123 type of skills.

ASSISTments2015[3] is collected from the same tutoring platform with ASSISTments2009 during year 2015–2016. In ASSISTments2015 dataset, each question only related to one skill. After dataprocess for duplicated records, there are 161,723 records from 4,210 students reserved in the dataset.

Ednet[4] is a dataset collected over 2 years by Santa, which is a multi-platform AI tutoring service. The dataset includes total 131,441,538 interactions from 784,309 students and 13,169 questions on 293 type of skills. Since the Ednet dataset is too large, we randomly choose 5,000 students with 1,079,483 records.

The dataset statistics are shown in Table 1.

Table 1. Dataset statistics

Dataset	#Questions	#Students	#Skills	#Records
Mynereus	184	202	48	86,772
ASSISTments2009	13016	4,151	110	325,637
ASSISTments2015	9073	4,210	100	161,723
Ednet	11187	5000	187	1,079,483

5.2 Baselines

The following KT models are chosen as baselines to measure the performance of the proposed SQKT model:

- **BKT** [5] models knowledge state as a set of binaries and use a Hidden Markov Model to update knowledge state.

[1] Mynereus: http://code.mynereus.com.

[2] ASSISTments2009: https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010.

[3] ASSISTments2015: https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data.

[4] Ednet: https://github.com/riiid/ednet.

- **KTM** [22] is the most comprehensive factor analysis model of KT, which has taken much side information into consideration.
- **DKT** [21] is the first deep learning KT method, which utilize a Recurrent Neural Network to extract the variation of the knowledge state.
- **DKVMN** [13] as an expansion of DKT, proposed a Mempry-Augmented Neural Network (MANN) [23] to represent the knowledge state of a student.
- **SKVMN** [1] as an expansion of DKVMN, use Hop-LSTM network in its sequence modeling.
- **GKT** [14] is a Graph Neural Network (GNN) based KT model, which casting the knowledge structure as a graph.

5.3 Metrics

We use AUC (the area under the Receiver Operating Characteristic (ROC) curve) to evaluate the KT models' prediction accuracy. The AUC score varies from 0 to 1, the higher the number is, the better the model performs. When the AUC score equals 0.5, the predictive model's accuracy is as same as random guess.

5.4 Model Evaluation

During experiments, each dataset was split into two parts: 70% for training and validation and 30% for testing. We used 5-fold cross validation to separate each training and validation subset, we divide the subset into 5 equal-sized parts, use 4 parts for training and 1 part for validation in turn.

Here the hyperparameters are chosen by grid search, we chose 0.01 as the learning rate, 0.1 as the epsilon value for Adam optimizer, 0.5 as the lambda for L2 loss, 5000 as the time threshhold, 5 as the window size of deep walk, and 100 as question embedding dimension.

Table 2. The AUC score of all KT models on all Datasets

Model	Mynereus	ASSISTments09	ASSISTments15	Ednet
BKT	0.7132	0.6271	0.6304	0.7401
KTM	0.7854	0.7169	0.6830	0.7829
DKT	0.8082	0.7961	0.7131	0.8519
DKVMN	0.8187	0.8157	0.7268	0.8721
GKT	0.8023	0.7940	0.7172	0.8790
SKVMN	0.8174	0.8348	0.7469	0.8760
SQKT	**0.8312**	**0.8416**	**0.7527**	**0.8841**

The overall performances of all KT models are shown in Table 2 and Fig. 5. From the result, we can sum up the following conclusions.

First of all, deep learning models generally outperform the traditional knowledge tracing models with an average improvement of 9.36% on AUC score, due

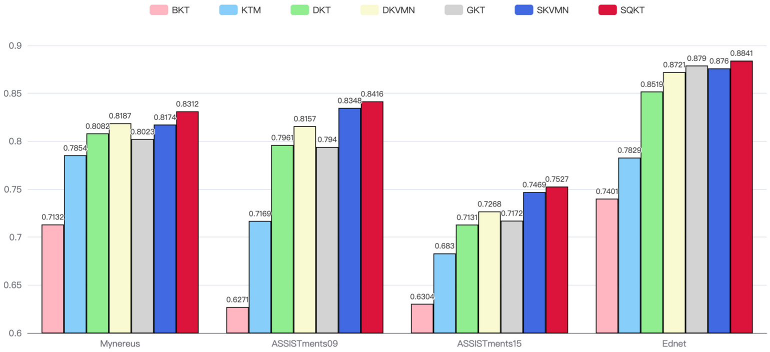


Fig. 5. The AUC score results of 7 KT models over 4 datasets

to the deep neural network's ability to learn complex student learning patterns. Second, the existing graph-based KT model such as GKT [14] and other DL models have advantages and disadvantages of each, GKT has a better score on Ednet dataset, while DKVMN and SKVMN performs better on ASSISTments datasets, which shows that the existing graph based methods are not perfect. Third, DL models with memory structure (such as DKVMN [29] and SKVMN [1]) performs better than no memory structure models (such as DKT), which shows the effectiveness of memory structure in storing student knowledge units. Last but not least, the proposed SQKT model outperforms all other existing models on all 4 datasets, the usage of question information and student attention have enhanced the prediction accuracy with an average of 0.8% in comparison to the state of art SKVMN model.

5.5 Ablation Studies

We also designed several ablation studies to further investigate the effect of our question representation and student attention module.

First, we compare our question representation module with 3 other methods, separately using random generalized embedding matrix, GCN (Graph convolutional network) and GAT (Graph attention network) to get the question embeddings. We denote these models as SQKT-Rand, SQKT-GCN and SQKT-GAT. The comparative experiment on 3 models is shown in Table 3.

Table 3. The AUC score of 3 comparative models and SQKT on all datasets

Dataset	SQKT-Rand	SQKT-GCN	SQKT-GAT	SQKT
Mynereus	0.8210	0.8307	0.8311	**0.8312**
ASSISTments09	0.8371	0.8386	0.8392	**0.8416**
ASSISTments15	0.7480	0.7516	0.7511	**0.7527**
Ednet	0.8769	0.8820	0.8824	**0.8841**

Next, we remove the student attention module, treat student's attention weight on all questions as the same, and denote this model as QKT. The comparative experiment result on QKT and SQKT model is shown in Table 4.

Table 4. The AUC score of QKT and SQKT on all datasets

Dataset	QKT	SQKT
Mynereus	0.8301	**0.8312**
ASSISTments09	0.8357	**0.8416**
ASSISTments15	0.7461	**0.7527**
Ednet	0.8760	**0.8841**

From the results, we can find that our question representation method achieved the best auc score among all 4 methods, while the attention module has proved to be effective through ablation experiment. It is worth mentioning that the student attention mechanism achieves a better improvement on larger dataset with longer time span. The comparative and ablation experiments have demonstrate the effectiveness of the modules we have proposed.

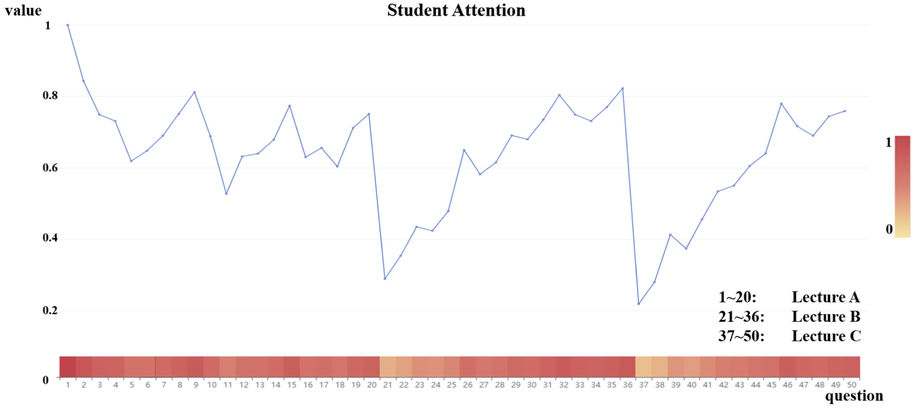


Fig. 6. The visualization of student attention through a student's exercise record

Figure 6 visualizes the variation of a student's attention during his learning process from Ednet [4] dataset. We intercepted the first 50 questions of the students' exercise record, and shows the attention on each question on the picture. The darker the red is, the more attention the student get, which means he can learn more on the question. The 50 questions are from 3 different lectures and the student finish these 3 lectures in turn. From the Fig. 6, we can see that the student attention have a clear reduction when he switch to a new lecture (around question 21 and 37). This phenomenon is very close to the actual human learning process, that keeping practice systematically on same-type questions can be more effective than practising randomly.

6 Conclusion

In this paper, we introduced a novel Student attention-based and Question-aware model for Knowledge Tracing (SQKT). In SQKT model, we first proposed a question representation method, which use Weighted Deep Walk method with Skip-Gram algorithm based on a weighted graph constructed from questions relationship. Then we introduced a student attention mechanism to measure attention weight when updating student knowledge state. Finally we use RNN with question attention to predict student's performance on the new coming question. Abundant experiments and ablation studies were conducted on SQKT model, the experiment result shows that SQKT model outperformed the state-of-the-art models over all datasets, and the ablation study proves the reasonableness and effectiveness of the proposed methods. For future work, more side information could be taken into consideration, and the structure of RNN network can be further optimized.

Acknowledgments. This work was supported by NSFC 61401155.

References

1. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 175–184 (2019)
2. Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_44
3. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis – a general method for cognitive model evaluation and improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 164–175. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_17
4. Choi, Y., et al.: EdNet: a large-scale hierarchical dataset in education. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 69–73. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_13
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adap. Inter. **4**(4), 253–278 (1994)
6. Desmarais, M.C., d Baker, R.S.: A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling User-Adapted Interact. **22**(1), 9–38 (2012)
7. Embretson, S.E., Reise, S.P.: Item Response Theory. Psychology Press, London (2013)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Liu, Q., et al.: EKT: exercise-aware knowledge tracing for student performance prediction. IEEE Trans. Knowl. Data Eng. **33**(1), 100–115 (2019)

10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546 (2013)
12. Minn, S., Desmarais, M.C., Zhu, F., Xiao, J., Wang, J.: Dynamic student classification on memory networks for knowledge tracing. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11440, pp. 163–174. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16145-3_13
13. Minn, S., Yu, Y., Desmarais, M.C., Zhu, F., Vie, J.J.: Deep knowledge tracing and dynamic student classification for knowledge tracing. In: 2018 IEEE International conference on data mining (ICDM), pp. 1182–1187. IEEE (2018)
14. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 156–163. IEEE (2019)
15. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)
16. Pandey, S., Srivastava, J.: RKT: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1205–1214 (2020)
17. Pardos, Z.A., Heffernan, N.T.: Modeling individualization in a Bayesian networks implementation of knowledge tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13470-8_24
18. Pardos, Z.A., Heffernan, N.T.: KT-IDEM: introducing item difficulty to the knowledge tracing model. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 243–254. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22362-4_21
19. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis-a new alternative to knowledge tracing. Online Submission (2009)
20. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
21. Piech, C., et al.: Deep knowledge tracing. arXiv preprint arXiv:1506.05908 (2015)
22. Rendle, S.: Factorization machines. In: 2010 IEEE International Conference on Data Mining, pp. 995–1000. IEEE (2010)
23. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning, pp. 1842–1850. PMLR (2016)
24. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
25. Su, Y., et al.: Exercise-enhanced sequential modeling for student performance prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
26. Tong, H., Zhou, Y., Wang, Z.: HGKT: introducing problem schema with hierarchical exercise graph for knowledge tracing. arXiv preprint arXiv:2006.16915 (2020)
27. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**(2), 270–280 (1989)

28. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18
29. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)