Deep Knowledge Tracing using Temporal Convolutional Networks

Nisrine Ait Khayi, Vasile Rus, Lasang Tamang
University of Memphis
{ntkhynyn,vrus, ljtamang}@memphis.edu

Abstract

The knowledge tracing (KT) task consists of predicting students' future performance on instructional activities given their past performance. Recently, deep learning models used to solve this task yielded relative excellent prediction results relative to prior approaches. Despite this success, the majority of these models ignore relevant information that can be used to enhance the knowledge tracing performance. To overcome these limitations, we propose a generic framework that also accounts for the engagement level of students, the difficulty level of the instructional activities, and the natural language processing embeddings of the text of each concept. Furthermore, to capture the fact that students' knowledge states evolve over time we employ a LSTM-based model. Then, we pass such sequences of knowledge states to a Temporal Convolutional Network to predict future performance. Several empirical experiments have been conducted to evaluate the effectiveness of our proposed framework for KT using Cognitive Tutor datasets. Experimental results showed the superior performance of our proposed model over many existing deep KT models. And AUC of 96.57% has been achieved on the Algebra 2006-2007 dataset.

1 Introduction

Modeling students' knowledge states reflecting their level of mastery with respect to a domain or set of targeted skills and concepts as well as predicting their future performance is an important task in learning science and engineering and is known as the Knowledge Tracing task (KT). It is usually leveraged to optimize students' learning trajectories, experiences, and outcomes. KT is a challenging task due to the complexity of the human learning processes (e.g., guessing, forgetting etc.) and the inherent difficulties of modeling knowledge (e.g., prior background; [Piech *et al*.,2015]). Further improvements in KT, which is the focus of our work, will have a wide range of benefits including better adaptation to individual learner's needs and, consequently, improved effectiveness at inducing learning gains and better learning experiences. There are other benefits of better KT solutions, as exemplified next. Given the tight link between domain models, i.e., the set of concepts to be mastered in a target domain, and knowledge tracing, better KT models, e.g., more accurate, will inform refinement of domain models. In addition, better KT will lead to designing new, more effective learning materials and instructional strategies.

Given the success of deep neural networks in other domains, deep KT models have gained significant attention recently. They use deep learning techniques to represent learners' latent knowledge states using large vectors of "artificial neurons". The parameters of these vectorial representations are inferred from data. Deep KT models use one hot encoding of the identification numbers (IDs) of the instructional activities, e.g., questions or problems to be solved, ignoring those items' characteristics. Several research works demonstrated that incorporating the items' semantics in the form of text embeddings into deep KT models can boost their prediction performance [Sonkar *et al*., 2020]. Despite the state-of-the-art results obtainedby these models, they suffer from major limitations. For instance, many of the existing deep KT models do not account for other important information such as the number of the correct attempts to solve a task and the duration of each step, which can be viewed as indicators of levels of engagement. To address these limitations and enhance the capabilities of KT models, many researchers incorporated into deep KT models various components such as learning ability [Minn *et al.,*2018], prior knowledge [Shen *et al*.,2020, and slipping and guessing effects [Cheng *et al*.,2020]. Furthermore, existing deep KT models consider all questions/items under a specific concept as equivalent observations. In attempt to improve instruction adaptation [Shailaja *et al.,* 2014] which will maximize (effectiveness) and speedup (efficiency) students' learning, we propose a generic framework that explores the underlying information among questions to enhance the performance of KT. The key components of our framework are:

- an NLP embedding component using the Sentence Universal Encoder [Cer, D.M. *et al*.,2018]. Given a question, we extract the semantics associated with its knowledge components (KCs) by averaging the embeddings corresponding to the textual descriptions of these knowledge components.

- a component capturing the engagement level of the student by dynamically assigning students into various groups based on their frequency interactions data using K-means clustering. To the best our knowledge this the first time an engagement level component is included in a deep KT model.
- a question/item difficulty component.
- knowledge embeddings resulted from the fusion of all these inputs (engagement level, text embeddings and item difficulty), using concatenation. The embeddings should lead to similar representations for items targeting similar concepts and having similar difficulty levels.
- an LSTM-based model to infer the students' latent knowledge states and a Temporal Convolutional Network to predict future responses.

We conducted series of experiments on the Cognitive Tutor datasets [Stamper *et al*.,2010] and we compared our proposed model to existing deep KT models.

2 Related Work

Several researchers incorporated important information about questions/items that can help better capture the wider learning context and therefore lead to improved ways to solve the KT problem. For example, Liu and colleagues [Liu *et al.*, 2020] proposed a Pre-training Embeddings via Bipartite Graph (PEBG) to learn a low dimensional embedding for each question based on additional information including question difficulty, question similarity, and skill similarity. They also introduced a product layer to fuse all the input features and obtain the final question embeddings which are incorporated into existing deep KT models. Experiments indicated an AUC (Area Under the Curve) improvement over state-of-the-art results by 8.6 % on average. Ghosh and colleagues [Ghosh *et al*.,2020] proposed an attentive knowledge tracing (AKT) model which combines an attentive neural model with various novel and interpretable model components inspired by cognitive and psychometric models. In addition to the question similarity, AKT uses Rasch model parameters: question difficulty and learning ability. The goal was to learn question embeddings that capture individual differences among questions targeting the same concept. Experimental results demonstrated improved performance of the AKT over prior KT models with a reported AUC improvement of up to 6%. To better model the individualization of prior knowledge and learning rates of various students, Shen, and colleagues [Shen *et al.,*2020] proposed a novel Convolutional Knowledge Tracing (CKT) model. More specifically, Hierarchical Convolutional layers were designed to extract learning rate features by processing many continuous learning interactions within a sliding window. Individualized prior knowledge was assessed according to students' historical learning interactions. Yang and colleagues [Yang *et al.,*2020] argued that knowledge tracing is affected by the most recent questions answered by students according to the forgetting curve theory. Based on this assumption, they proposed a Convolutional Knowledge Tracing (CKT) model that captures the long-term effect of the entire question-answer sequence and short-term effect of the recent questions using 3D convolutions. The experimental results showed an AUC improvement relative to existing KT models.

Cheng and colleagues [Cheng *et al*.,2020] proposed an adaptable knowledge tracing (AKT) framework that integrates slipping and guessing factors into the model to obtain more reasonable knowledge state results and leverage the semantics of question texts for more precise knowledge tracing. They obtained improved performance over several KT models.

Shin and colleagues [Shin *et al.,*2020] proposed SAINT+, a successor of SAINT which is a Transformer based knowledge tracing model. The main addition in this new version is incorporating two temporal features: elapsed time which is the time taken by the student to answer a question and lag time, the time interval between adjacent learning activities. The empirical results showed an improvement in the AUC over the SAINT model on EdNet, the largest public dataset in the education domain according to some metrics, e.g., data from almost 800k students. It should be noted that EdNet contains learning data from the domain of English learning, i.e., it does not target learning of complex STEM topics.

In a continuous effort to overcome the major limitation of deep KT models in capturing differences among questions targeting the same concept and to enhance the knowledge tracing ability, we consider additional information such as students' engagement level. We also consider the semantics of the key concepts targeted by each question using a novel natural language processing (NLP) algorithm, i.e., the Universal Sentence Encoder. We also employ for the first time a novel times series model based on a Temporal Convolutional Network to predict student performance.

3 Model Architecture

Our proposed framework comprises of several important components. First, we account for the engagement level of students in the form of a one hot encoded vector of the engagement cluster that the student belongs to at each time t. Second, our model uses an averaged embedding vector of the knowledge components associated with each question. The third component consists of a one hot encoded vector reflecting the difficulty level of each question. Fourth, we use a one hot encoded vector of the ID of each question. Then, we fuse all these inputs to generate a knowledge embedding for each question answered by the students. The sequence of these knowledge embeddings of each student are passed to an LSTM based model to learn students' knowledge states of the student for specific concepts. The sequences of these vector are then passed to a Temporal Convolutional Network (TCN) connected with a Sigmoid layer to predict performance.

3.1 Dynamic Assessment of the Engagement Level of Students using Clustering

Several research studies have revealed the positive correlation between student engagement and academic performance with higher engagement level associated with better grades [Casuso-Holgado *et al.,* 2013; Lee, 2014]. Therefore, we take into consideration this information in predicting the future student performance. Inspired by the frequency-based metrics proposed by Reid et al [Reid *et al.,*2012] to assess engagement level, we propose to use the following metrics available in the Cognitive Tutor dataset:

- **Step Duration**: the elapsed time of the question in seconds, calculated by adding all of the durations for transactions that were attributed to the question.
- **Incorrects:** total number of incorrect attempts by the student on the question.
- **Corrects:** total correct attempts by the student for the question.
- **Hints**: total number of hints requested by the student for the question.

Based on these metrics, we dynamically assess students' engagement level by clustering, where each cluster represents a level of engagement. Assigning students into a group with similar engagement level at each time step is performed by k-means clustering. Following the research results of Moubayed and colleagues [Moubayed *et al.,*2020], we consider a three-level model to classify the students' engagement levels. Hence, the parameter k of the k-means clustering algorithm is set to 3.

After clustering students into three groups with distinct levels of engagement, the clustering results are added to the training data. That is given a student with a specific concept (KC), we add its corresponding cluster value (e.g.,1,2 etc.).Then, a student's engagement level is encoded as a vector e_t^i with C+1 dimensions where C is the number of concepts or knowledge components in the dataset. We encode in this vector, the associated concepts, and the engagement level cluster that the student belongs to at each time step t (an instance in the data records) as following:

$$e_t^i\left[kc_t^i\right] = 1 \text{ and } e_t^i[C+1] = c_m \qquad (1)$$

where kc_t^i is the associated knowledge component or concept for the question q_t^i at a specific time t and c_m is the engagement level cluster where $c_m \in \{0,1,2\}$.

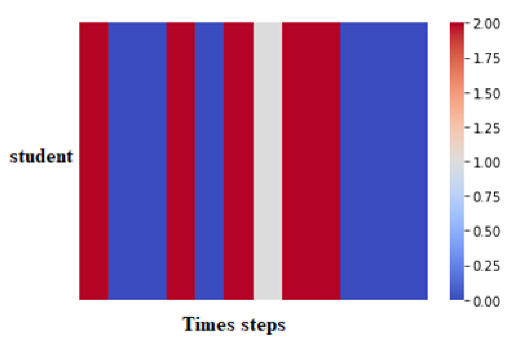

Figure 1. Engagement level evolution of a student at different time steps while interacting with Cognitive Tutor.

As an example, Figure1 illustrates the evolution of the engagement level of one student at different time steps on three knowledge components: "simple fractions", "multiplication", and "combine like terms". Each bar in the figure represents a different KC. The red color reflects that the student belongs to the low engagement group, blue color reflects belonging to the medium engagement group, and gray color reflects their belonging to the high engagement group. It is important to emphasize that the engagement level of the student differs from a concept to another.

3.2 NLP Embedding

To enhance the knowledge tracing ability and improve the prediction of our proposed deep KT model, we add an NLP embedding component that captures the semantics of the knowledge components. To this end, we use the recently proposed NLP embedding approach called Universal Sentence Encoder (USE) [Cer, D.M. *et al*.,2018]. USE employs the encoder component of the transformer.

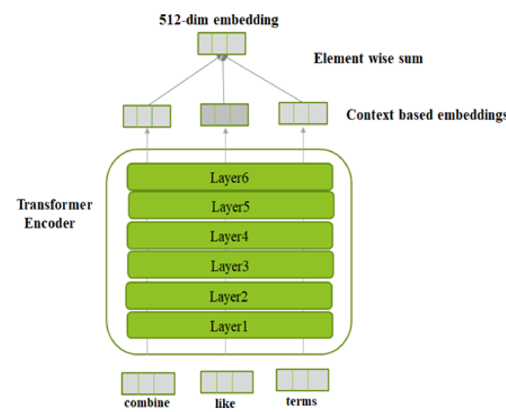

Figure 2. The architecture of the Universal Sentence Encoder

As shown in Figure 2, first, each textual description of a concept is converted to lowercase and tokenized into tokens using the Penn Treebank (PTB) tokenizer. USE relies on a self-attention mechanism that takes into consideration the token order and its surrounding context for generating each token's representation. The context-based token representations are then converted to a fixed length text vector by computing the element-wise sum of the representations at each token position. The encoder outputs a 512-dimensional vector for the knowledge component text embedding. Then, we average the embeddings of the knowledge components of each question at each time step t as a way to capture the semantics of each question with respect to the targeted concepts or KCs.

3.3 Difficulty Constraint

To capture difficulty of a question, the main assumption we make is that the more the number of mathematical concepts required to answer a specific question the more difficult and sophisticated the question is. Therefore, we calculate the difficulty level of each question by dividing the total number of KCs in a question by the total number of KCs in the Cognitive Tutor dataset.

After calculating questions' difficulty, we encode the question difficulty as a vector with *C+1* dimension where *C* is the number of concepts. Following the prior works, the first C entries of the vector represent the one hot encoded concepts. The last entry of the vector represents the difficulty level of the question.

3.4 Knowledge Tracing using LSTM.

After obtaining the knowledge embedding sequences of each student at time *t,* we pass them to an LSTM base model to compute the knowledge hidden states that represent the mastery level of different KCs.

The resulted hidden states are passed to the Temporal Convolutional Network (TCN) to compute the prediction of future answers.

3.5 Response Prediction using Temporal Convolutional Network (TCN)

Temporal Convolutional Networks (TCN) have been proposed first by Lea et al [Lea *et al.,*2016] for the video-based action segmentation task. The distinguishing characteristics of TCNs are:

- the convolutions in the architecture are causal, meaning that there is no leakage from future to past.
- the model can take a sequence of any length and map it to an output sequence with the same length.

Added to this, TCN-based models outperform the LSTM-based models for time series predictions.

Motivated by these advantages, we propose to use TCN to improve the prediction of future responses, as described next. Let H^i_t be the knowledge state with respect to the target domain of student *i* at time step t. The TCN's architecture is composed of 1D pooling/upsampling and channel wise normalization layers in the encoder. For each of the convolutional layers in the encoder, we apply a set of 1D filters that capture how the input signals evolve over the course of an action. Pooling enables us to efficiently compute activations over a long period of time. The channel wise normalization has been effective in recent CNN methods.

The i-th component of the activation vector output of the encoder is calculated as follows:

$$\hat{E}^l_{i,t} = f\left(b^{(l)}_i + \sum_{t'=1}^{d} \langle W^{(l)}_{i,t'}, E^{(l-1)}_{t+d-t'} \rangle\right) \qquad (2)$$

where f(.) is a Leaky Rectified Linear Unit, b^l represents the biases, $E^{(l-1)}$ is the activation matrix from the previous layer, and d is the filter duration, and it is set as the mean segment duration for the shortest class from the training set.

The pooled activation vector $\hat{E}^l_t$ is then normalized by its highest response at that time step m = $max_i \hat{E}^l_{i,t}$ with small $\boldsymbol{\varepsilon}$ such that :

$$E^{(l)}_t = \frac{1}{m+\varepsilon} \hat{E}^l_t \qquad (3)$$

The decoder part of the TCN is similar to the encoder and the order of the operations is the following: upsample, convolve, then normalize. The activation output is $D^{(l)}_t$.

Finally, we compute the prediction probabilities of answering future questions by passing the activation matrix output of the decoder to a sigmoid function as follows :

$$Y_t = sigmoid\left(D^{(l)}_t\right) \qquad (4)$$

4 Experiments

We conducted several experiments to demonstrate the effectiveness of our proposed model for the knowledge tracing (KT) task using the Cognitive Tutor datasets. Details about the data are provided next.

4.1 The Data

Our datasets come from the Cognitive Tutor [Anderson *et al.,*1995] that teaches students algebra (middle school and high school). Cognitive Tutor presents a problem to a student in form of questions (also called steps) of many skills/concepts. That is every question targets multiple KCs. The Cognitive Tutor uses Knowledge Tracing to determine when a student has mastered a skill. In this work, we consider the following attributes of each record in the dataset: Student ID, Step Name that represents the question, Step Duration (sec), KC(Default/SubSkills) that represents the associated skill or concept, Incorrects, Corrects, Hints, and Correct First Attempt that is a binary attribute and we consider it as the target to be predicted in our model.

In this paper, we use three of the development datasets [Stamper *et al*.,2010]: the "2005-2006 Algebra" dataset, the "2006-2007 Algebra" dataset, and the "Bridge to Algebra 2006-2007" dataset. The Algebra I dataset consists of 813,661 total responses over 387 skills covering practice attempts for 3,310 students. The Bridge to Algebra dataset contains data from 1,146 students and includes 3,679,199 total logged responses for 494 skills. Table 1 details important statistics of the data after preprocessing.

	#Records	#Students	#Skills
Algebra I 2005-2006	809694	574	113
Algebra I 2006-2007	2270384	1338	492
Algebra to Bridge	3679199	1146	494

Table 1. Cognitive Tutor Datasets Statistics

4.2 Experimental Setup

Data Preprocessing. Consistent with data preprocessing in prior works, we conducted the following data preprocessing steps: removed duplicate records, removed records with NaN KCs, removed records with dummies KCs, discarded learners that have fewer than 10 interactions with the system and discards skills answered by less than 10 students.

Training and Testing. In all experiments, we perform 5-cross fold validation using a Tesla K80 GPU and a total of 12 GB of RAM. We split all datasets at the student level: at each iteration, 80% of students were used for training and 20% were used for testing. For LSTM, we considered 100 units as the dimensionality of the output space. For the TCN parameters settings, we set the number of filters to use in the convolutional layers to 64. The kernel size was set to 6 and we considered a dilation list = [1,2,4,8,16,32,64]. An Adam optimizer [13] with a learning rate of 0.001 was used and gradients were clipped to 1.0 to prevent exploding gradients. Due to the large size of our datasets, we considered a small batch size of 5. To evaluate the performance of our model, we customized the Binary Cross Entropy loss since our model gives predictions for all skills/concepts. Given a new student in the testing data, the model predicts her performance in all the concepts in the Cognitive dataset. Thus, to generate the prediction for a specific skill, we take the column-wise dot product between the predictions and a one-hot encoding of the skill.

Our model was evaluated on the test data and the performance of the model is reported using the area under the ROC curve (AUC). AUC measure ranges from 0.5 reflecting a low ability to distinguish from correct and incorrect answers to 1.0 reflecting a perfect discrimination. We compute the AUC by obtaining the prediction of each student in the testing data across all concepts.

Compared Methods. To demonstrate the effectiveness of our proposed model, we compare its AUC results with existing methods the reported results on the same Cognitive Tutor data sets. The methods are:

- **DAS3H** [Chofin et al.,2019]: incorporates item-skills relationships and forgetting effects.
- **qDKT** [Sonkar et al., 2020]: models every learner's success probability on individual questions over time. qDKT incorporates graph Laplacian regularization to smooth predictions and uses an initialization scheme inspired by the fastText algorithm.
- **DynEmb** [Xu et al.,2020]: enables the tracking of student knowledge without the concept/skill tag information that other KT models require.
- **Transformer-based DKT** [Pu,et al.,2020]: a Transformer based model that addresses the forgetting issue by accounting for elapsed time. It also uses the questions-skills associations to learn representations of both frequent and rare questions.

4.3 Results and Discussion

Table 2 lists the AUC performance results of our proposed model using the three datasets of Cognitive Tutor. Including all the model components: the engagement level, the question difficulty and the knowledge component text embeddings, the model performs better on the Algebra 2006-2007 dataset with an AUC of 96.57%, followed by the Algebra 2006-2007 dataset with an AUC of 92%, and the Bridge dataset with an AUC of 91%. As shown in Table 4, these results reflect a significant performance gain in comparison with existing deep KT methods.

Model	AUC Result (%)
EL+QD+KCE-Algebra2005-2006	92
EL+QD+KCE-Algebra2006-2007	96.57
EL+QD+KCE-Bridge2006-2007	91

Table 2. The performance results of the proposed model and the ablation experiments results.

Table 3 lists some examples of the results of our proposed framework based on the engagement level and the difficulty of the questions. As shown in the table, the engagement level of the student has more impact on the performance in comparison with the difficulty of the questions. Students with high levels of engagement perform well regardless of the difficulty of the questions. However, it is less probable that the low engaged students perform well especially when the question is difficult.

Engagement Level	Difficulty of Questions	Performance	% of testing data
High (1)	Easy (<0.5)	High	13
High (1)	Easy (<0.5)	Low	6
High (1)	Difficult (>0.5)	High	13
High (1)	Difficult (>0.5)	Low	6
Low (2)	Easy (<0.5)	High	0.4
Low (2)	Easy (<0.5)	Low	1
Low (2)	Difficult (>0.5)	High	0.47
Low (2)	Difficult (>0.5)	Low	2.68

Table 3. Example of the results in terms of engagement level and difficulty of the question

Ablation experiments. To test the impact of each key component of our model on the overall performance, we conducted a series of ablation experiments. The ablation experiments are:

- First, we removed the engagement level component from the model inputs and evaluated the resulting model on the Algebra2006-2007 dataset. The experiments showed that the model AUC has decreased by 1.07 %. Based on the existing literature, this is a significant statistical drop in the AUC results reflecting the important of this component.
- Second, we removed the question difficulty from the model inputs. The results showed that the model AUC has decreased by 0.01% which is not statistically significant. Thus, the question difficulty component is not important in the framework.
- Third, we removed the knowledge component text embeddings from the model's inputs. The AUC of the model has decreased, reflecting the importance of this component. Generally, the Universal Sentence Encoder boosts the prediction ability of deep learning models.

Model	AUC Result (%)
EL+QD+KCE-Algebra2006-2007	**96.57**
DAS3H	86
qDKT	89.5
DynEmb	86
Transformer- based DKT	78.4

Table 4. Comparison of performance results between our proposed model and the existing DKT methods using the Cognitive Tutor datasets

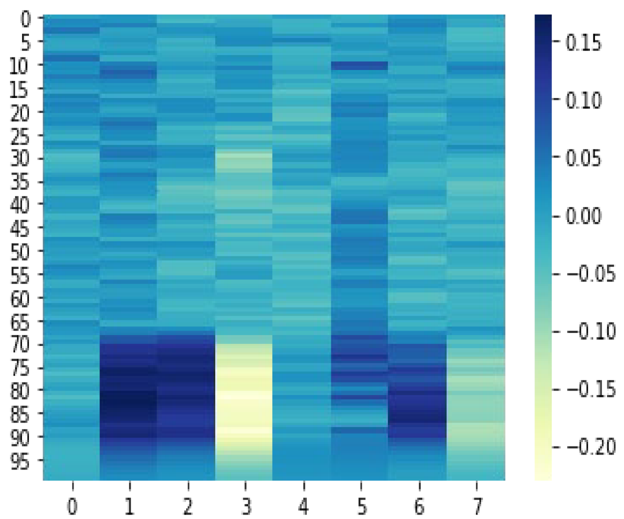

Figure 3. Evolution of knowledge states of a student over time.

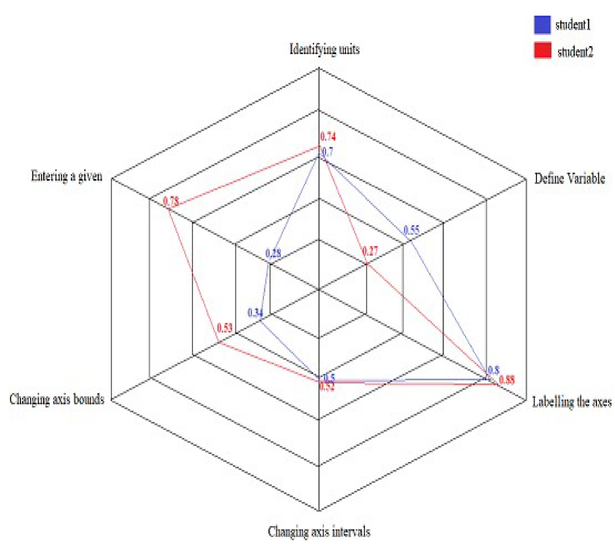

Figure 4. Mastery level of two students in six concepts.

Visualizations. Naturally, students' knowledge states evolve over time. To illustrate this evolution process, we generated several visualizations which are more intuitive and user-friendly representations of mastery levels at one moment in time and over time. Figure 3 depicts a student's knowledge states over a sequence of 91 questions, covering 17 concepts (e.g., identifying units, perform multiplication, simple fractions, etc.), from the Algebra 2005-2006 dataset. At each time step t, a knowledge state consists of an activation vector of 8 dimensions. The lighter the color the lower the mastery level of the student on a specific concept. Figure 4 shows the mastery level of two students on 6 concepts. Students can recognize their poor knowledge points and the ITS makes individual learning schemes by tracing the knowledge state.

5 Conclusions

In a continuous research effort to enhance the ability of knowledge tracing and improve the prediction of future responses, we proposed a generic framework that includes several important components. First, the model assesses dynamically the engagement level of students across concepts and incorporates this information into the learned knowledge embeddings. Second, it includes the semantics of the knowledge concepts through learning embeddings using a the recently proposed method called the Sentence User Encoder. Third, the model calculates the difficulty of each question and uses this information together with the other inputs to make predictions. Knowledge embeddings of students are then learned by concatenating all these components. Finally, students' knowledge states over time are modeled using an LSTM neural network. These learned sequences of the hidden states are then passed to a Temporal Convolutional Network to predict the future performances of students. The experimental results showed the effectiveness of our proposed model in comparison with existing deep KT methods, yielding high AUC results on Cognitive Tutor datasets. The conducted ablation experiments demonstrated the importance of the engagement level , the text embeddings of the knowledge components and the Temporal Convolutional Network algorithm. Their elimination or substitution led to a decrease in the AUC results. More specifically, the students showing high levels of engagement perform very well for both the difficult (difficulty rate >0.5) and easy questions (difficulty rate <0.5). However, it is less probable that students with low levels of engagement perform well, especially, when a question is difficult. This probability increases slightly when the question is easy.

In our future work, we will apply this proposed framework on other educational datasets. We will also investigate additional ways to enhance the model's interpretability and improve its prediction's performance.

Acknowledgements. "The Learner Data Institute is sponsored by the National Science Foundation (NSF; award #1934745). The opinions, findings, and results are solely the authors' and do not reflect those of NSF."

References

[Anderson *et al.,*1995] Anderson, John R, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. “Cognitive tutors: Lessons learned”. *The journal of the learning sciences*,1995, 4 (2), 167–207.

[Casuso-Holgado *et al.,*2013] Casuso-Holgado, M.J., Cuesta-Vargas, A., Moreno-Morales, N., Labajos Manzanares, M.T., Barón-López, F.J., & Vega-Cuesta, . The *association between academic engagement and achievement in health sciences students*. 2013,BMC Medical Education, 13, 33 - 33.

[Cer, D.M. *et al*.,2018] Cer, D.M., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., & Kurzweil, R. Universal Sentence Encoder. In. *Proceedings of the 2018 conference of EMNLP.*

[Cheng et al.,2020] Cheng, S., Liu, Q., & Chen, E.. Domain Adaption for Knowledge Tracing. *ArXiv*,2020, abs/2001.04841.

[Choffin et al.,2019] Choffin, B., Popineau, F., Bourda, Y., & Vie, J. J. ,2019. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. In. Proceedings of the EDM conference..

[Ghosh et al.,2020] Ghosh, A., Heffernan, N., & Lan, A.S. Context-Aware Attentive Knowledge Tracing. In *Proceedings of the 26th ACM SIGKDD*. 2020

[Lea et al.,2016]] Lea, C., Vidal, R., Reiter, A., & Hager, G. D. Temporal convolutional networks: A unified approach to action segmentation. In *Proceedings of the European Conference on Computer Vision* , 2016(pp. 47-54). Springer, Cham.

[Lee,2014] Lee, J.-S.. The relationship between student engagement and academic performance: Is it a myth or reality? *The Journal of Educational Research*,2014 107, 177 - 185.

[Liu et al.,2020] Liu, Y., Yang, Y., Chen, X., Shen, J., Zhang, H., & Yu, Y.. Improving Knowledge Tracing via Pre-training Question Embeddings. In *Proceedings of the 29th AJICAI*,2020.

[Minn et al., 2018] Minn, S., Yu, Y., Desmarais, M. C., Zhu, F., & Vie, J. J.. Deep knowledge tracing and dynamic student classification for knowledge tracing. *In Proceedings of the 2018 IEEE International Conference on Data Mining*,2018 (ICDM) (pp. 1182-1187).

[Moubayed et al.,2020] Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H.. Student Engagement Level in an e-Learning Environment: Clustering Using K-means. *American Journal of Distance Education*, 2020, 34, 137 - 156.

[Piech et al.,2015] Piech, C., Bassen, J., Huang, J., Gan guli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J.. Deep knowledge tracing. *In Advances in neural information processing systems* (pp. 505-513),2015.

[Pu,et al.,2020] Pu, S., Yudelson, M., Ou, L., & Huang, Y. 2020. Deep Knowledge Tracing with Transformers. In Proceedings of the Artificial Intelligence in Education, 12164, 252 - 256.

[Reid et al.,2012] Reid, L. F.. Redesigning a large lecture course for student engagement: Process and outcomes. *The Canadian Journal for the Scholarship of Teaching and Learning*, 2012,3, 2–39

[Shailaja et al.,2014] Shailaja, J., & Sridaran, R.. Taxonomy of e-learning challenges and an insight to blended learning. In *Proceedings of the International conference on intelligent computing applications* ,2014 (icica’14) (pp. 310–314)

[Shen et al.,2020] Shen, S., Liu, Q., Chen, E., Wu, H., Huang, Z., Zhao, W., Su, Y., Ma, H., & Wang, S.. Convolutional Knowledge Tracing: Modeling Individualization in Student Learning Process. In *Proceedings of the 43rd ACM SIGIR*,2020.

[Shin et al.,2020] Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., & Choi, Y.. SAINT+: Integrating Temporal Features for EdNet Correctness Prediction. *ArXiv*,2020 abs/2010.12042.

[Sonkar et al., 2020] Sonkar, S., Waters, A., Lan, A.S., Grimaldi, P.J., & Baraniuk, R. 2020. qDKT: Question-centric Deep Knowledge Tracing. In *Proceedings of the International Conference EDM*,2020.

[Stamper et al.,2010] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., & Koedinger, K.R.. [Development] data set from *KDD Cup 2010 Educational Data Mining Challenge*,2010.

[Yang et al.,2020] Yang, S., Zhu, M., Hou, J., & Lu, X.. Deep Knowledge Tracing with Convolutions. *ArXiv,*2020, abs/2008.01169.

[Xu et al.,2020] Xu, L., & Davenport, M. 2020. Dynamic Knowledge Embedding and Tracing. In Proceedings of the International Conference of EDM