Majumdar, R. et al. (Eds.) (2025). Proceedings of the 1st International Conference on Learning Evidence and Analytics. Asia-Pacific Society for Computers in Education

Enhancing Attention-Based Knowledge Tracing with Digital Textbook Interaction

Kotaro KAWABATA[a]*, Fumiya OKUBO[b], Yuta TANIGUCHI[c], Cheng TANG[b] & Atsushi SHIMADA[b]
[a]*Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan*
[b]*Faculty of Information Science and Electrical Engineering, Kyushu University, Japan*
[c]*Research Institute for Information Technology, Kyushu University, Japan*
*kawabata.kotaro.707@s.kyushu-u.ac.jp

Abstract: Knowledge Tracing (KT) models a learner's knowledge state by analyzing past responses to predict future performance. While traditional KT models focus on response correctness, few studies incorporate learning activity data during study sessions. This study proposes a KT model that integrates features from digital textbook viewing logs to enhance knowledge estimation. Experiments on a university course dataset demonstrate that incorporating study-related contextual information improves prediction performance, highlighting the impact of digital learning behavior on KT.

Keywords: Knowledge Tracing, Learning Analytics, Digital Textbook, Deep Learning

1. Introduction

With the advancement of ICT, EdTech services such as Learning Management Systems (LMS) and digital textbooks have become widely adopted, enabling data-driven learning support. Knowledge Tracing (KT) utilizes learner's past learning activity data, such as correctness of responses, exercise content, and response time, to estimate their knowledge state and predict future performance. However, conventional KT models primarily rely on exercise records, which fail to capture broader learning behaviors beyond exercises.

Learning activities have been analyzed and utilized for grade prediction by leveraging log from LMS, e-portfolio systems, digital textbooks (Okubo et al,. 2017). However, many existing Knowledge Tracing (KT) models primarily focus on exercise response histories and may not sufficiently account for such diverse learning activities.

To address this limitation, this study enhances KT models by integrating digital textbook reading behaviors, specifically leveraging reading duration adjusted by content similarity between textbook materials and exercises. We validate our model by comparing it with a conventional exercise-based KT model, using a dataset collected from a university lecture. The models are evaluated using five key metrics (e.g., AUC, RMSE) to assess how the integration of learning process data influences predictive accuracy.

Our approach demonstrates improvements in knowledge state estimation by incorporating learning process information into KT models, highlighting the potential of integrating diverse learning activities beyond exercises.

2. Related Work

2.1 Knowledge Tracing (KT)

Knowledge Tracing (KT) is a task that estimates a learner's knowledge state based on the correctness of previously solved problems and predicts the correctness of the next problem. The advantages of KT include providing feedback on estimated proficiency levels and applications for personalized support tasks. The first KT model was Bayesian Knowledge

Tracing (BKT), proposed by Corbett and Anderson (1995), which utilized a Bayesian Network. However, since Piech et al. (2015) introduced Deep Knowledge Tracing (DKT) using a Recurrent Neural Network (RNN), research on KT models based on deep learning has gained significant attention. Other KT models include Self-Attentive Knowledge Tracing (SAKT) (Pandey and Karypis, 2019), which employs a Multi-head Attention mechanism.

Recent studies have incorporated various types of learner information. For example, Lu et al. (2024) utilize action logs to model students' knowledge states. However, such approaches cannot explicitly capture the content relevance between the learning and the predicted problems. We address this by integrating view time of relevant textbook pages into the attention mechanism.

3. Proposed Method

3.1 Knowledge Tracing Definition and Problem Formulation

In this study, KT is defined as follows: At each time step t, a learner's solved exercise is represented as e_t, and its correctness is denoted as r_t, where $r_t = 1$ if correct and $r_t = 0$ otherwise. The interaction at time step t is then defined as $x_t = (e_t, r_t)$. Thus, KT is a task that predicts the correctness probability $\hat{r}_{t+1}$ for the exercise e_{t+1} at time step $t + 1$, based on the sequence of past interactions $(x_1, x_2, \dots, x_t)$, aiming to make $\hat{r}_{t+1}$ as close as possible to the actual correctness r_{t+1}. In KT, et can be represented either at the exercise level or at the Knowledge Component (KC) level. However, in this study, we adopt the exercise-level representation.

3.2 Self-Attentive Knowledge Tracing (SAKT) and the Proposed Model

Figure 1 illustrates the overall architecture of the proposed model, which is derived from Self-Attentive Knowledge Tracing (SAKT) with modifications to incorporate additional learning-related information.

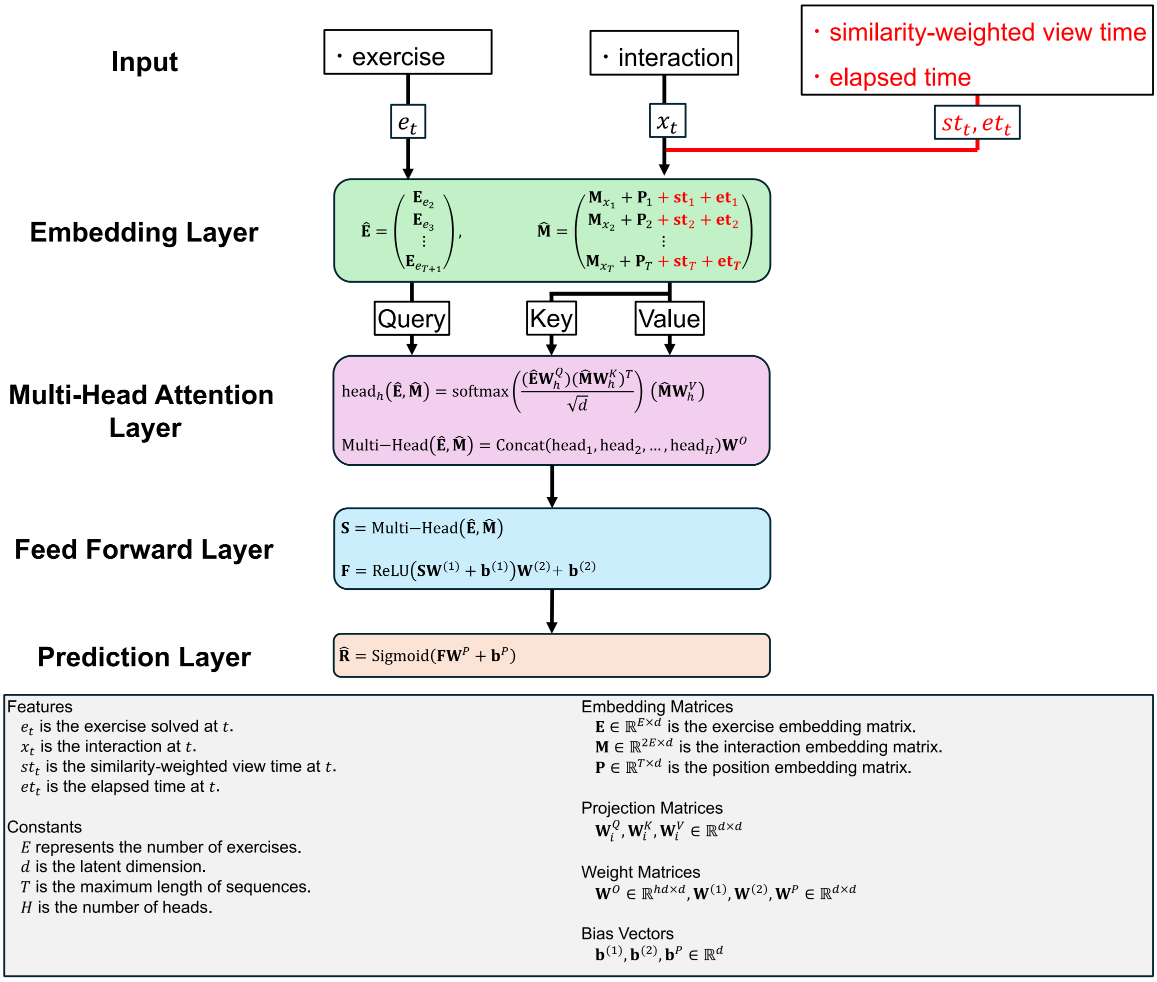
SAKT uses the exercise vector, whose correctness is to be predicted, as the query, while interaction vector with positional encoding serves as the key and value. These inputs are fed into the Multi-Head Attention mechanism, which generates an intermediate representation. Next, this representation is passed through a feedforward network to produce the output $\mathbf{F}$. Finally, the Prediction Layer converts $\mathbf{F}$ into the correctness probability. Binary cross-entropy is used as the loss function.

In the proposed model, additional information is incorporated into the key and value inputs of the Multi-Head Attention mechanism in SAKT. Specifically, we add information on digital textbook viewing related to the exercise and elapsed time, defined as the time a student takes to answer an exercise, following SAINT+(Shin et al., 2021). This modification enables the model to integrate the information that learners have studied to solve exercises.

To compute the similarity-weighted view time st_t, we first obtain the view time $vt_{t,i}$ for each textbook page i from log data. Then, we compute the cosine similarity between the exercise embedding $\mathbf{e}_t^{text}$ and each page embedding $\mathbf{e}_i^{page}$. The exercise embedding is based on the concatenation of the exercise and choices, while each page embedding uses OCR-extracted text from the textbook page. Both are computed using OpenAI's text-embedding-3-large model (OpenAI, n.d.). Finally, st_t is calculated as the sum of $vt_{t,i}$ multiplied by the corresponding similarity scores, as shown in Equation (1).

$$st_t = \sum_{i=1}^{I} \cos - \mathrm{sim}\left(\mathbf{e}_t^{text}, \mathbf{e}_i^{page}\right) vt_{t,i} \tag{1}$$

Following SAINT+, elapsed time, representing elapsed time is incorporated into the model. The total elapsed time for a test is extracted from the LMS log data, and then averaged by dividing it by the number of exercises in the test. Before being fed into the model, these features are normalized to a range of 0 to 1 and then vectorized.



$$\hat{\mathbf{E}} = \begin{pmatrix} \mathbf{E}_{e_2} \\ \mathbf{E}_{e_3} \\ \vdots \\ \mathbf{E}_{e_{T+1}} \end{pmatrix}, \quad \hat{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_{x_1} + \mathbf{P}_1 + \mathbf{st}_1 + \mathbf{et}_1 \\ \mathbf{M}_{x_2} + \mathbf{P}_2 + \mathbf{st}_2 + \mathbf{et}_2 \\ \vdots \\ \mathbf{M}_{x_T} + \mathbf{P}_T + \mathbf{st}_T + \mathbf{et}_T \end{pmatrix}$$

$$\mathrm{head}_h(\hat{\mathbf{E}}, \hat{\mathbf{M}}) = \mathrm{softmax}\left(\frac{(\hat{\mathbf{E}}\mathbf{W}_h^Q)(\hat{\mathbf{M}}\mathbf{W}_h^K)^T}{\sqrt{d}}\right)(\hat{\mathbf{M}}\mathbf{W}_h^V)$$

$$\mathrm{Multi{-}Head}(\hat{\mathbf{E}}, \hat{\mathbf{M}}) = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \dots, \mathrm{head}_H)\mathbf{W}^O$$

$$\mathbf{S} = \mathrm{Multi{-}Head}(\hat{\mathbf{E}}, \hat{\mathbf{M}})$$

$$\mathbf{F} = \mathrm{ReLU}(\mathbf{S}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

$$\hat{\mathbf{R}} = \mathrm{Sigmoid}(\mathbf{F}\mathbf{W}^P + \mathbf{b}^P)$$

Features
e_t is the exercise solved at t.
x_t is the interaction at t.
st_t is the similarity-weighted view time at t.
et_t is the elapsed time at t.

Constants
E represents the number of exercises.
d is the latent dimension.
T is the maximum length of sequences.
H is the number of heads.

Embedding Matrices
$\mathbf{E} \in \mathbb{R}^{E \times d}$ is the exercise embedding matrix.
$\mathbf{M} \in \mathbb{R}^{2E \times d}$ is the interaction embedding matrix.
$\mathbf{P} \in \mathbb{R}^{T \times d}$ is the position embedding matrix.

Projection Matrices
$\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times d}$

Weight Matrices
$\mathbf{W}^O \in \mathbb{R}^{hd \times d}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^P \in \mathbb{R}^{d \times d}$

Bias Vectors
$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \mathbf{b}^P \in \mathbb{R}^d$

Figure 1. Overall architecture of the proposed model. The parts highlighted in red indicate the elements newly added to the original SAKT framework.

4. Experiments

4.1 Datasets

To evaluate our model, we used data from "Information Science" courses at Kyushu University offered during the 2021-2023 academic years. The dataset consists of six courses, and its statistics are shown in Table 1.

Table 1. *Dataset Statistics*

User	Exercise	Interaction	Average Accuracy
236	132	15,891	0.8173

4.2 Experiments Setting

In this study, the evaluation metrics were Area Under the ROC Curve (AUC), Accuracy (ACC), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and F1 Score (F1). For comparison, we used SAKT, SSAKT (Choi et al. 2020), and SAINT (Choi et al. 2020), which are KT models incorporating an attention mechanism.

For all models, we used the Adam optimizer with a learning rate of 0.001 and set the dropout rate to 0.1. The embedding dimension d was selected from {50, 100, 150, 200}, and the number of heads h in the Multi-Head Attention was selected from {1, 2, 5}. The number of training epochs was determined by early stopping. The dataset was split into training, validation, and test sets in a ratio of 8:1:1, and we constructed 10 different non-overlapping test splits. Among these, the split that achieved the highest average AUC on the test set was selected, and the results on this split were reported as the model's performance.

4.3 Experimental Results

Table 2 presents the evaluation results for each model. This table shows that our model achieved the best performance. Since the proposed model outperforms SAKT, it is likely that the additional components we introduced contributed to the performance improvement.

Table 2. *KT performance prediction comparison*

Model	AUC	ACC	MAE	RMSE	F1
SAKT	0.6885	0.6276	0.4205	0.4745	0.6758
SSAKT	0.6216	0.5865	0.4586	0.4888	0.6148
SAINT	0.5039	0.5145	0.4946	0.5069	0.6190
Ours	**0.7016**	**0.6380**	**0.4159**	**0.4715**	**0.6877**

4.4 Attention Weights

Figure 2 compares attention weights of SAKT and our model, alongside similarity-weighted view time and elapsed time for a learner. Each heatmap shows queries (vertical) attending to past keys (horizontal). Our model focuses attention on keys 1–9 and 28–30, aligning with high view time, and suppresses attention on 31–34. This suggests it emphasizes content relevance, unlike SAKT's broader distribution.

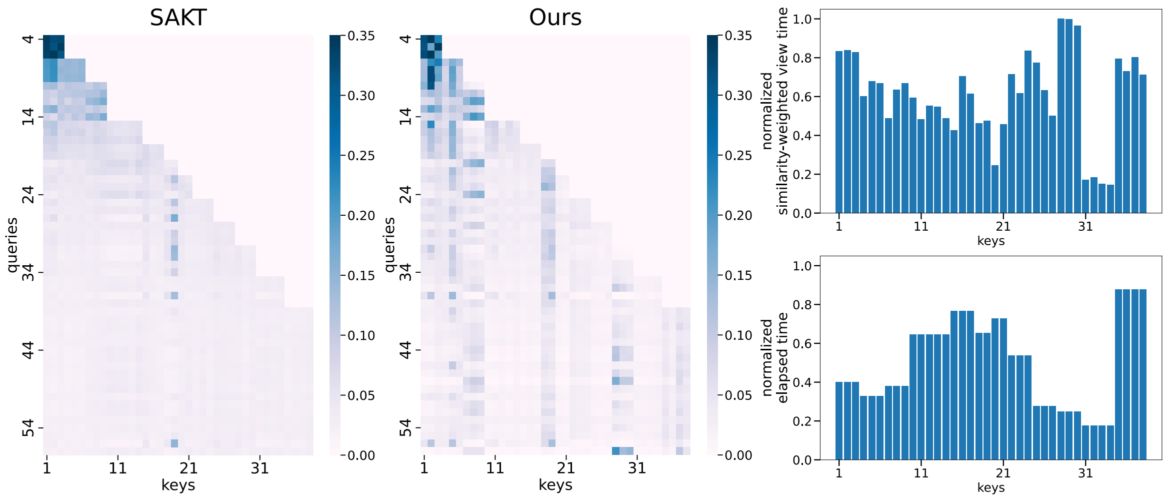


Figure 2. Comparison of SAKT attention weights (left), Ours attention weights (center), and normalized time features (right: normalized similarity-weighted view time on top, normalized elapsed time on bottom).

4.5 Ablation Study

The results of the ablation study on the impact of similarity-weighted view time and elapsed time are shown in Table 3. SAKT+ST represents a model that adds only similarity-weighted view time to SAKT, while SAKT+ET represents a model that adds only elapsed time to SAKT. From this table, it can be observed that the model incorporating both features achieved the highest accuracy in four out of five evaluation metrics, except for ACC. Additionally, SAKT+ST outperformed SAKT in all evaluation metrics, whereas SAKT+ET only outperformed SAKT in AUC but not in the other metrics.

These results suggest that combining the two features is effective in enhancing the predictive performance of Knowledge Tracing. Furthermore, the superiority of SAKT+ST over SAKT+ET can be attributed to the greater influence of similarity-weighted view time on Attention Weights, as discussed in Section 4.4.

Table 3. *KT performance prediction comparison in ablation study*

Model	AUC	ACC	MAE	RMSE	F1
SAKT	0.6885	0.6276	0.4205	0.4745	0.6758
SAKT+ST	0.6963	**0.6398**	0.4191	0.4730	0.6853
SAINT+ET	0.6900	0.6237	0.4210	0.4776	0.6705
Ours	**0.7016**	0.6380	**0.4159**	**0.4715**	**0.6877**

4.6 Discussion and Limitations

Our model shows improved accuracy by using textbook viewing data, but there are limitations. The method depends on the availability of detailed log data and may not generalize to different courses or institutions. Also, the content similarity is based on simple text matching, which could be refined.

5. Conclusion

In this study, we proposed a Knowledge Tracing model that incorporates learning-related information during student learning. By integrating digital textbook viewing time weighted by similarity to the exercises, our model enables more accurate predictions than models relying solely on in-exercise interactions. Experimental results on real-world datasets showed that our model outperforms existing methods, highlighting the value of incorporating external learning resources. However, generalizability remains limited by the availability of log data, and future work should refine similarity techniques and validate the approach in broader contexts.

Acknowledgements

This research was supported by JST CREST JPMJCR22D1 and JSPS Grant-in-Aid for Scientific Research JP22H00551. We would like to express our gratitude for their support.

References

Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C., & Heo, J. (2020). Towards an appropriate query, key, and value computation for knowledge tracing. Proceedings of the Seventh ACM Conference on Learning @ Scale, 341–344. https://doi.org/10.1145/3386527.3405945

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modelling and User-Adapted Interaction, 4(4), 253–278. https://doi.org/10.1007/BF01099821

Lu, Y., Tong, L., & Cheng, Y. (2024). Advanced Knowledge Tracing: Incorporating Process Data and Curricula Information via an Attention-Based Framework for Accuracy and Interpretability. *Journal of Educational Data Mining*, *16*(2), 58–84.

Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. (2017). A neural network approach for students' performance prediction. Proceedings of the Seventh International Learning Analytics & Knowledge Conference, 598–599. https://doi.org/10.1145/3027385.3029479

OpenAI. (n.d.). *Vector Embeddings*. OpenAI Platform. Retrieved June 12, 2025, from https://platform.openai.com/docs/guides/embeddings

Pandey, S., Karypis, G. (2019). A Self-Attentive model for Knowledge Tracing. arXiv preprint arXiv:1907.06837.

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*. Curran Associates, Inc.

Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., & Choi, Y. (2021). SAINT+: Integrating Temporal Features for EdNet Correctness Prediction. LAK21: 11th International Learning Analytics and Knowledge Conference, 490–496. Presented at the Irvine, CA, USA. doi:10.1145/3448139.3448188