Tracing Knowledge for Tracing Dropouts: Multi-Task Training for Study Session Dropout Prediction

Seewoo Lee*
Riiid! AI Research
seewoo.lee@riiid.co

Kyu Seok Kim*
Riiid! AI Research
kyuseok.kim@riiid.co

Jamin Shin
Riiid! AI Research
jamin.shin@riiid.co

Juneyoung Park
Riiid! AI Research
juneyoung.park@riiid.co

ABSTRACT

Study session dropout prediction allows for educational systems to identify when a student would stop a study session which gives vital information to prolong learning activity. Student session dropout can depend on many factors that are involved with the engagement when using the system. The student's knowledge level and their track records within the system are closely related to the student's willingness to continue with their study. Knowledge tracing as a task models the user's knowledge level given study history. The information from knowledge tracing can have significant impact on predicting the student's willingness to continue, which is why it is natural to train two tasks jointly for better generalization in dropout prediction task. While extensive research has been conducted individually on dropout prediction and knowledge tracing, the effect of jointly modeling two tasks has not been thoroughly investigated. Hence, we show that multi-task training of the study session dropout prediction model along with knowledge tracing boosts the performance of study session dropout prediction, especially on more challenging tasks and datasets. Specifically, with Transformer-based models, multi-task training significantly improves Area Under Receiving Operator Curve (AUROC) by 3.62% in further N-step dropout prediction task, which is a study session dropout prediction task under a more practical setting. Moreover, under label-scarce and class-imbalance settings, our method shows improvements of AUROC up to 12.41% and 11.22%, respectively. Our results imply that knowledge tracing is closely related to study session dropout prediction and can transfer positive knowledge in multi-task training, which provides a new way to better predict dropouts especially in difficult settings.

Keywords

Dropout prediction, Multi-task Training, Knowledge Tracing

1. INTRODUCTION

The advantages of e-learning has gathered the attention of both educators and researchers. One of the lasting problems in e-learning is the ability to maintain the user's attention during the system use. For instance, students in mobile learning environments are more prone to distractions and exhibit difficulties in concentration [16, 20, 5]. Thus, being able to properly identify when these issues occur will allow an Intelligent Tutoring System (ITS) [1] to appropriately and preemptively intervene. This task is called Study Session Dropout Prediction and has been recently proposed by [21]. Predicting such session dropout is a crucial task in Educational Data Mining (EDM) to understand student's behaviors and learning environments, which can lead to increased learning effect.

However, study session dropout prediction has not yet been extensively studied. Many recent research works have instead focused on predicting student dropout in environments like universities or Massive Open Online Courses (MOOC) [3, 13, 27, 33, 35]. Interestingly, [22, 9, 26] has also shown that one of the main reasons students drop out from schools or classes is their academic performance which is highly relevant to their knowledge states. Given such knowledge, we hypothesize that study session dropout can also be attributed to the knowledge states of students.

Hence, in this paper, we jointly model study session dropout prediction with knowledge tracing, which is a heavily studied task that [12, 23, 8] that aims to predict the student's future performance on *knowledge components* (e.g. questions or concepts) given the student's historical data. In this study, we address this issue through a machine learning methodology known as multi-task learning [4]. Multi-task learning jointly trains multiple tasks together to formulate a comprehensive understanding of the nature of the data. Specifically, we implement a multi-task training model that is trained with both session dropout prediction and knowledge tracing.

The contributions of this paper are as such:

- We provide a multi-task training framework to jointly model study session dropout prediction and knowledge tracing.
- We show that our multi-task training framework boosts the performance of the trained model on study session dropout prediction. Also, we show that our method elevates the performance in further N-step dropout prediction task, where the model has to predict dropouts not only in immediate time step, but also in future time steps.
- We perform extensive ablation studies to show that multi-task training shows even higher performance on more difficult experimental settings such as label scarcity and class-imbalance. We also show with ablation studies that the thresh-

old of lag time that we use to define session dropout also affects the performance of multi-task training.

2. RELATED WORKS

2.1 Dropout prediction

There have been many attempts to predict dropouts in various environments. Traditionally, dropout prediction has been incorporated to predict student dropouts [3, 27]. Following the proliferation of internet, dropout prediction became applicable in online services. [2, 15] incorporated deep learning methods in *Spotify Sequential Skip Prediction Challenge*, where the task is to infer the songs that will be skipped in the second half session after the first half. Models such as denoising autoencoder and variants of Long Short Term Memory (LSTM) network were utilized. [13, 33, 35] used deep learning methods such as Convolutional Neural Network (CNN) to predict students' dropouts in MOOC. Study session dropout prediction has been studied to discover student's involvements in mobile learning environments. [21] utilized the Transformer network [31], which replaced the recurrent architecture of Recurrent Neural Networks (RNN) by self-attention blocks, to predict the study session dropout probability in mobile learning environment. We used **D**eep **A**ttentive Study **S**ession Dropout prediction (DAS) model in [21] with multi-task training approach in our experiments.

2.2 Multi-Task training

Multi-task training or multi-objective training is a method to train a machine learning model with multiple objectives [37], which tries to enhance the performance on original task by sharing features with auxiliary tasks. It has been used across various domains in machine learning, such as Computer Vision and Natural Language Processing. For example, in [24], the authors used multi-task training for face labeling by training a CNN to handle both likelihoods and pairwise label dependencies. Multi-Task Deep Neural Network (MT-DNN, [25]), is a BERT [11] based model with several task-specific layers for multi-task learning, which outperforms vanilla BERT's performance on the GLUE benchmark [32].

There are also several applications of multi-task training in an educational field. Huang et al. [18] presents a transformer-based model that identifies whether a given voice of a teacher corresponds to a question or not, which solve the problem as a multi-class classification problem to recognize question types. Geden et al. [14] proposed LSTM-based model to predict correctness rate of all questions instead of the average correctness rate for the related questions. In [19], Huang et al. suggests Deep Reinforcement Learning based exercise recommendation system whose reward function is designed to satisfy multiple objectives.

2.3 Knowledge Tracing

Knowledge tracing is a task of modeling students' knowledge level given their learning activities. Knowledge tracing and dropout prediction shares the aspect that they both model students' responses given their learning histories. Bayesian Knowledge Tracing (BKT) is a traditional method which treats student's learning activities as binary variables representing whether the student understands a certain concept or not [36]. Some works proposed to incorporate deep learning methods in knowledge tracing. [29] feeds the users' one-hot encoded learning activities into RNN-based model architectures to output the correctness prediction probability. [7, 28] are the works that use Transformer-based architectures for knowledge tracing. SAINT [7] has a similar architecture to DAS, which uses Transformer's both encoder and decoder structure.

3. METHODS

3.1 Study Session Dropout Prediction

Formally, a student's learning history is given as a sequence of interactions

$$I = (I^{(1)}, I^{(2)}, I^{(3)}, \ldots, I^{(T)})$$

where each $I^{(j)} = (e^{(j)}, l^{(j)})$ includes meta-data of the question $e^{(j)}$ that a student solves at j-th step (e.g. question id, category of the question, question text, ...) and the meta-data of the student's response $l^{(j)}$ (e.g. response correctness, elapsed time, timeliness, ...) at j-th step. Then the *study session dropout prediction* is to estimate the probability

$$\mathbb{P}[y_{DP}^{(j)} = 1 | I^{(1)}, I^{(2)}, \ldots, I^{(j-1)}, e^{(j)}]$$

that the session dropout occurs after solving j-th question. Note that a sequence can contain multiple sessions. As in [21], we define one-hour inactivity as a session dropout, so that the dropout label at j-th step is given by

$$y_{DP}^{(j)} = \begin{cases} 1 & lt^{(j)} := st^{(j+1)} - st^{(j)} \geq 1 \text{ hour} \\ 0 & \text{otherwise} \end{cases}$$

where $st^{(j)}$ is the *start time* at j-th step, i.e. the time that user start to solve the question, and $lt^{(j)}$ is the lag time for the j-th interaction.

3.2 Input Representation

The representation of each interaction $I^{(j)} = (e^{(j)}, l^{(j)})$ is formulated similarly to the settings in [21]. Here are some minor differences of feature settings between our model and the original DAS model in [21]:

1. Instead of *start time*, we use the *lag time* feature. It is more directly related to the dropout and leads to the substantial gain in the model's performance. Since the distribution of a lag time is long-tailed, we use the logarithm of the lag time instead of the lag time itself. (See Figure 2 for the distribution of the lag time). It is used as a decoder's input, not for an encoder.
2. We use *continuous embedding* for *elapsed time*, instead of discrete embedding. More precisely, we first clip the actual elapsed time with maximum 300 seconds, then normalize it by dividing it with 300. After that, we get a latent embedding vector for the elapsed time et by $\mathbf{v} = \mathbf{v}(\mathrm{et}) = \mathrm{et} \cdot \mathbf{w}_{et}$, where $\mathbf{w}_{et}$ is a single trainable vector which has same dimension as the model.

3.3 Model

In this section, we describe our methodology to jointly perform training in dropout prediction and knowledge tracing. We use the shared model f to generate the shared feature representations for both dropout prediction and knowledge tracing. An arbitrary model f takes the sequences of question embeddings $e = [e^{(1)}, \ldots, e^{(j)}]$ and response embeddings $l = [l^{(1)}, \ldots, l^{(j-1)}]$ to produce the shared feature representation for dropout prediction and knowledge tracing. Then, the feature representation is fed into the final separate prediction layers to output predicted dropout probabilities and response correctness:

$$\hat{y}_{DP} = \sigma\big(\mathbf{W}_{\mathrm{DP}}(f(e,l)) + \mathbf{b}_{DP}\big)$$
$$\hat{y}_{KT} = \sigma\big(\mathbf{W}_{\mathrm{KT}}(f(e,l)) + \mathbf{b}_{KT}\big)$$

Further Steps	$N = 1$ (Positive label 4.06%)				$N = 5$ (Positive label 18.34%)				$N = 10$ (Positive label 32.52%)			
model	AUROC		AUPRC		AUROC		AUPRC		AUROC		AUPRC	
	vanilla	multi-objective	vanilla	multi-objective	vanilla	multi-objective	vanilla	multi-objective	vanilla	multi-objective	vanilla	multi-objective
LSTM	0.8704	**0.8717**	0.3208	**0.3229**	0.7586	**0.7599**	0.4577	**0.4598**	0.736	**0.7367**	0.5880	**0.5886**
GRU	0.8652	**0.8670**	0.3083	**0.3133**	0.7567	**0.7570**	0.4547	**0.4556**	0.7338	**0.7349**	0.5842	**0.5855**
DAS	0.8807	**0.8836**	0.3469	**0.3534**	0.7579	**0.7734**	0.4598	**0.4815**	0.7229	**0.7491**	0.5717	**0.6061**

Table 1: Test AUROCs and AUPRCs of DAS and RNN-based dropout prediction models. Further steps N of each task and its positive label proportions of the dataset are indicated in the top row ($N = 1$ corresponds to the original session dropout prediction task). Best result for each model is indicated in bold.

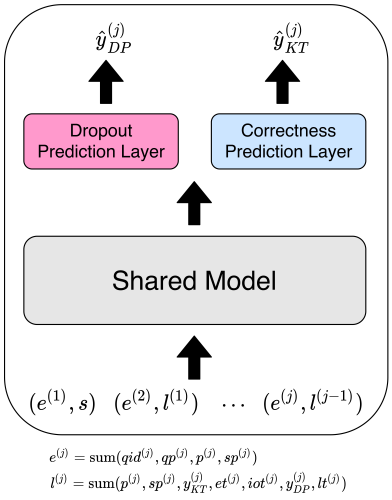

Figure 1: Overall architecture of our multi-task training scheme. Note that s is the starting token.

The major difference between our method and the previous dropout prediction models is that we are jointly training the model to predict both student dropout *and* response correctness, by using different prediction layer for each task at the end of the shared model. Using separate prediction layers, we produce both $\hat{y}_{DP} = [\hat{y}_{DP}^{(1)}, \ldots, \hat{y}_{DP}^{(j)}]$ and $\hat{y}_{KT} = [\hat{y}_{KT}^{(1)}, \ldots, \hat{y}_{KT}^{(j)}]$, which are predicted probabilities for study session dropout ($\hat{y}_{DP}$) and response correctness ($\hat{y}_{KT}$) for each time step. Training scheme of our approach is described in Figure 1. The major baseline that we use for our methodology is DAS [21], which is a Transformer-based model to predict study session dropout. The details of the architecture of DAS is described in Appendix A. We also do experiments with RNN-based model architectures - including LSTM and GRU [6, 17] - which are provided as baselines in [21] for comparison. For RNN-based models, we use encoder-only structure instead of encoder-decoder structure.

3.4 Training objectives

Typically, Binary Cross-Entropy (BCE) loss is used in 2-class classification tasks, which include the cases of dropout prediction and knowledge tracing. We use the BCE function to compute $\mathcal{L}_{\text{DP}}$ and $\mathcal{L}_{\text{KT}}$, which are the losses for dropout prediction and knowledge tracing. We train the model with the loss

$$\mathcal{L} = \mathcal{L}_{\text{DP}} + \lambda_{\text{KT}} \mathcal{L}_{\text{KT}}$$

where λ_{KT} is a balancing hyper-parameter. Our experiments are performed with $\lambda_{\text{KT}} = 0.5$.

4. EXPERIMENTS

4.1 Experiment setup

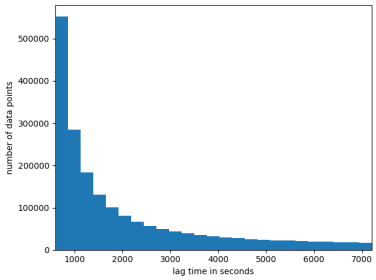

Figure 2: Distribution of data points with respect to the lag time. The lag time in the graph ranges from 600s (10 minutes) to 7200s (2 hours). The number of data points exponentially decays as their lag time increases.

mask rate	Positive Label Proportion	AUROC		AUPRC	
		vanilla	multi-objective	vanilla	multi-objective
50%	3.80%	0.8381	**0.8832**	0.2599	**0.3521**
90%	3.80%	0.7752	**0.8740**	0.1602	**0.3304**
95%	3.80%	0.7264	**0.7598**	0.1202	**0.1322**
99%	3.81%	0.6826	**0.6837**	**0.0984**	0.0901
50%	1.90%	0.8418	**0.8833**	0.2775	**0.3520**
90%	0.38%	0.7677	**0.8204**	0.1504	**0.2093**
95%	0.19%	0.7175	**0.7980**	0.1163	**0.1726**
99%	0.04%	0.6782	**0.7304**	0.0923	**0.1095**

Table 2: Test AUROCs and AUPRCs of the DAS model with various masking rates on *both labels (first 4 rows)* and *only positive dropout labels (last 4 rows)*. The first two columns indicate the rate of random masking and the proportion of labels in the training data for each mask rate. Best result for each masking rate indicated in bold.

We use the EdNet-KT1 dataset [8], the largest publicly available student interaction dataset collected by *Santa*[*], which is a mobile application for preparing Test of English for International Communication (TOEIC) exam. The proportion of the logs where session dropout occurred was 4.06% with the definition of dropout as one-hour lag time. The distribution of dropout labels w.r.t. the change of lag time in defining the dropout is described in Figure 2.

For RNN-based model architectures, we use the embedding and model dimension size of 256 and feedforward layer dimension size of 1024 with 2 number of layers. For DAS, we use the embedding and model dimension size of 512 and feedforward layer dimension size of 2048 with 4 number of layers. While training, we set the model's input sequence size as 100, and all the models are trained with Adam optimizer with Noam scheduling where the warmup step is 40000. We set the initial learning rate and the model's dropout rate as 0.001 and 0.1 respectively.

We evaluate our models with two metrics: Area Under Receiving Operator Curve (AUROC) and Area Under Precision Recall Curve (AUPRC). AUROC is the most widely used metric in the litera-

[*]https://aitutorsanta.com/

lag time	Positive Label Proportion	AUROC		AUPRC	
		vanilla	multi-objective	vanilla	multi-objective
600s	6.33%	0.8853	**0.8876**	0.4327	**0.4387**
1800s	4.61%	0.8786	**0.8821**	0.3582	**0.3657**
3600s	4.06%	0.8807	**0.8836**	0.3469	**0.3534**
5400s	3.79%	0.8768	**0.8857**	0.3309	**0.3519**
7200s	3.60%	0.8789	**0.8876**	0.3298	**0.3508**

Table 3: Test AUROCs and AUPRCs of the DAS model with various standards of lag time on defining dropouts (in seconds). The first two columns indicate the lag time used to define session dropout and the proportion of positive labels in the total dataset for each definition. Best result for each indicated in bold.

ture for evaluating the dropout prediction models because labels in dropout prediction settings are usually imbalanced. It is known that AUPRC is especially more informative than the AUROC when the dataset's labels are imbalanced [10, 30].

We evaluate the effect of multi-task training on two tasks. The first is the standard study session dropout prediction described in 3.1, where the task is to estimate the probability that the session dropout occurs after the current time step. However, in the actual service, it is more important to predict whether the user will dropout within several future time steps in order to respond to the user's engagement in advance. Thus, we also evaluate our method on *further N-step dropout prediction* task, which predicts whether the user will dropout within further N time steps. In further N-step dropout prediction, the number of future time steps that the model has to consider increases as N increases. We perform further N-step dropout predictions with $N \in \{5, 10\}$. For all tasks, we measure AUROC and AUPRC on LSTM, GRU, and DAS with and without multi-task training to validate the effect of our method.

4.2 Main results

The results of multi-task training on the study session dropout prediction are given in Table 1. The multi-task training with knowledge tracing improves AUROC and AUPRC for the dropout prediction across all models. We also present the effect of our method in further N-step dropout predictions in Table 1. The results show that in further N-step dropout prediction tasks, multi-task training increases the performance of the model by larger margins than in immediate dropout prediction task. Note that multi-task training shows higher increase in AUROC when $N = 10$ since the future steps that the model has to consider increases with N, leaving more room for multi-task training to help the model. Table 1 also includes the proportion of positive labels of the dataset in each task to explain the difference of AUPRCs between the tasks. Although further 10-step prediction shows lower AUROCs compared to other tasks, since its dataset is less imbalanced, it shows higher AUPRCs.

4.3 Ablation study

We performed ablation studies on immediate study session dropout prediction task for fair comparisons. We perform ablations on label scarcity, imbalanced datasets, and various standards on dropout definition as follows.

4.3.1 Scarce Label for Dropout Prediction

It has been known that multi-task training shows higher performance when the label of target domain is scarce [34]. To verify this notion, we evaluated the multi-task training on datasets with different levels of dropout prediction label scarcity. Specifically, we randomly masked out both positive and negative dropout labels in different proportions ranging in {50%,90%,95%,99%}. The results in Table 2 shows that multi-task training indeed shows higher performance when dropout prediction labels are scarce. Since at least some amount of labels are needed for the models to converge, the result with 99% mask rate fails to show meaningful results.

4.3.2 Imbalanced Dataset

As we mentioned before, study session dropout prediction usually suffers from the imbalanced dataset. In our case, the rate of the positive label is only 4.06% of the total data. We conjecture that our multi-task training approach is also helpful when the label of the dataset is extremely imbalanced. To show this, while training, we randomly masked out certain proportion of **positive** dropout labels during training, and evaluated the model on the same validation and test set as before. Note that this is different from 4.3.1 since 4.3.1 performs random masking on both positive and negative dropout prediction labels. The proportion of random masking also ranges in {50%,90%,95%,99%}. The results are given in the Table 2. Results show that multi-task training outperforms vanilla model more heavily on imbalanced datasets.

4.3.3 Definition on Dropout

Although we define one-hour (3600s) inactive lag time as a session dropout, other definitions of a dropout may be utilized to better analyze student's learning activities. Thus, we see how the effect of multi-task training varies with the change in the definition of a session dropout. Figure 2 shows the distribution of the number of dropout labels w.r.t. the inactivity duration (lag time). We compare the results with various lag time standards of a session dropout in $\{600s, 1800s, 3600s, 5400s, 7200s\}$. The results are given in Table 3. Results show that multi-task training tends to perform better in tasks with higher inactivity duration standards of a session dropout. This is because the tasks with higher lag time standards have more imbalanced datasets. Since imbalanced datasets tend to have lower AUPRC, tasks with higher lag time standards have lower AUPRCs.

5. CONCLUSIONS

In this paper, we proposed a multi-task training approach with knowledge tracing to boost the performance of study session dropout prediction. We hypothesized that the commonality between the dropout prediction and knowledge tracing tasks would be beneficial to predict dropouts. We empirically validated with Transformer-based and RNN-based models that multi-task training enhances the dropout prediction performance especially in further N-step dropout prediction, which is a more practical task in real service. Moreover, we performed extensive ablation studies to demonstrate that multi-task training shows even better performance on more difficult experimental settings. We remain the multi-task training with other tasks in the field of Artificial Intelligence in Education (AIEd) as the future work.

6. REFERENCES

[1] J. R. Anderson, C. F. Boyle, and B. J. Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.

[2] F. Beres, D. M. Kelen, and A. A. Benczur. Sequential skip prediction using deep learning and ensembles. In *International Conference on Web Search and Data Mining*, 2019.

[3] F. D. Bonifro, M. Gabbrielli, G. Lisanti, and Z. Stefano. Student dropout prediction. In *International Conference on*

Artificial Intelligence in Education, pages 129–140. Springer, 2020.

[4] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[5] Q. Chen and Z. Yan. Does multitasking with mobile phones affect learning? a review. *Computers in Human Behavior*, 54:34–42, 2016.

[6] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[7] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and H. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning*, pages 341–344, 2020.

[8] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, pages 69–73. Springer, 2020.

[9] F. J. da Costa, M. de Souza Bispo, and R. de Cássia de Faria Pereira. Dropout and retention of undergraduate students in management: a study at a brazilian federal university. *RAUSP Management Journal*, 53(1):74–85, 2018.

[10] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning.*, pages 233–240, 2006.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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)*, pages 4171–4186, 2019.

[12] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266, 2009.

[13] W. Feng, J. Tang, and T. X. Liu. Understanding dropouts in moocs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 517–524, 2019.

[14] M. Geden, A. Emerson, J. Rowe, R. Azevedo, and J. Lester. Predictive student modeling in educational games with multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 654–661, 2020.

[15] C. Hansen, C. Hansen, S. Alstrup, J. G. Simonsen, and C. Lioma. Modelling sequential music track skips using a multi-rnn approach. In *International Conference on Web Search and Data Mining*, 2019.

[16] B. A. Harman and T. Sato. Cell phone use and grade point average among undergraduate university students. *College Student Journal*, 45(3):544–550, 2011.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[18] G. Y. Huang, J. Chen, H. Liu, W. Fu, W. Ding, J. Tang, S. Yang, G. Li, and Z. Liu. Neural multi-task learning for teacher question detection in online classrooms. In *International Conference on Artificial Intelligence in Education*, pages 269–281. Springer, 2020.

[19] Z. Huang, Q. Liu, C. Zhai, Y. Yin, E. Chen, W. Gao, and G. Hu. Exploring multi-objective exercise recommendations in online education systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1261–1270, 2019.

[20] R. Junco. Too much face and not enough books: The relationship between multiple indices of facebook use and academic performance. *Computers in human behavior*, 28(1):187–198, 2012.

[21] Y. Lee, D. Shin, H. Loh, J. Lee, P. Chae, J. Cho, S. Park, J. Lee, J. Baek, B. Kim, et al. Deep attentive study session dropout prediction in mobile learning environment. *arXiv preprint arXiv:2002.11624*, 2020.

[22] S. A. Lim and R. Rumberger. Why students drop out of school: A review of 25 years of research. 2008.

[23] R. V. Lindsey, M. Khajah, and M. C. Mozer. Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in neural information processing systems*, pages 1386–1394. Citeseer, 2014.

[24] S. Liu, J. Yang, C. Huang, and M.-H. Yang. Multi-objective convolutional learning for face labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3451–3459, 2015.

[25] X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, 2019.

[26] M. Manacorda. Grade failure, drop out and subsequent school outcomes: quasi-experimental evidence from uruguayan administrative data. 2006.

[27] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura. Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, 33(1):107–124, 2016.

[28] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 384–389, 2019.

[29] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 01, pages 505–513, 2015.

[30] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and P. Illia. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[32] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[33] W. Wang, H. Yu, and C. Miao. Deep model for dropout prediction in moocs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, pages 26–32, 2017.

[34] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell. Characterizing and avoiding negative trasfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition, 2019.

[35] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Mooc dropout prediction: How to measure accuracy? In *Proceedings of the Fourth ACM Conference on Learning*, pages 161–164, 2017.

[36] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013.

[37] Y. Zhang and Q. Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

APPENDIX

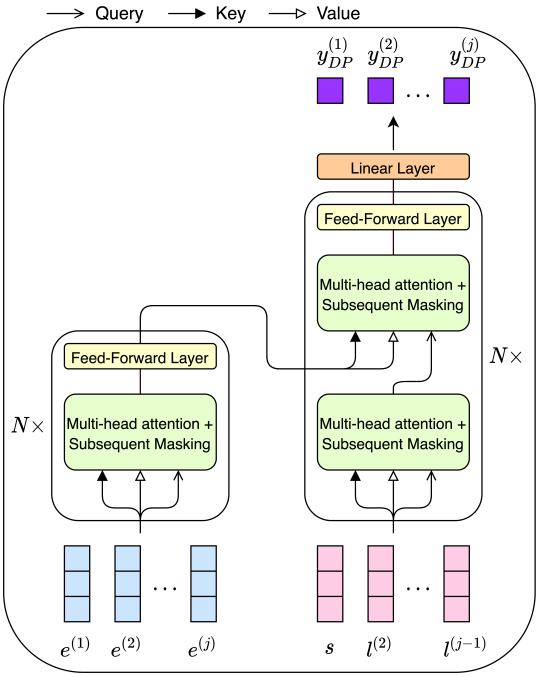

Figure 3: Overall architecture of DAS. Note that s is the starting token and N is the number of layers.

A. ARCHITECTURE OF DAS

In this section, we review the overall architecture of Deep Attentive Study Session Prediction (DAS) [21], which we use for the baseline of our model. DAS is a transformer-based model that consists of an encoder and a decoder. Encoder includes N encoder blocks where each block has a multi-head self-attention layer and a fully connected feed-forward layer. After each layer, residual connection and layer normalization are applied. Decoder also includes N decoder blocks with each block including a multi-head self-attention layer, a multi-head encoder-decoder attention layer, and a fully connected feed-forward layer. The encoder-decoder attention layer takes the output of the encoder as keys and values, and output of self-attention layer as queries to perform the attention mechanism. Each layer in the decoder block is also followed by residual connection and layer normalization. The encoder takes the sequence of question embeddings $e = [e^{(1)}, \ldots, e^{(j)}]$ and produces the outputs $h = [h^{(1)}, \ldots, h^{(j)}]$ that are fed into the decoder's encoder-decoder attention layers. The decoder takes the sequence of response embeddings $l = [s, l^{(1)}, \ldots, l^{(j-1)}]$ and encoder's outputs h, producing the hidden vectors which are fed through the final linear layer to output the predicted dropout probabilities $\hat{y}_{DP} = [\hat{y}_{DP}^{(1)}, \ldots, \hat{y}_{DP}^{(j)}]$. Note that s is the starting token for the first position of the sequence. The overall process of DAS can be described as:

$$h = \text{Encoder}(e)$$
$$\hat{y}_{DP} = \sigma\big(\mathbf{W}_{\text{DP}}\text{Decoder}(s, l, h) + \mathbf{b}_{DP}\big)$$

where s is the start token embedding. The overall architecture of DAS is described in Figure 3.

We will now describe the components of each block in encoder and decoder. Each block mainly consists of a multi-head attention layer and a fully connected feed-forward layer. Multi-head attention network in each block takes queries, keys, and values of the sequence as inputs. Queries, keys, and values of $head_i$ are computed by multiplying weight matrices W_i^Q, W_i^K, W_i^V to the inputs as follows:

$$Q_i = e_Q W_i^Q = [Q_i^{(1)}, \ldots, Q_i^{(j)}]$$
$$K_i = e_K W_i^K = [K_i^{(1)}, \ldots, K_i^{(j)}]$$
$$V_i = e_V W_i^V = [V_i^{(1)}, \ldots, V_i^{(j)}]$$

Then, multi-head attention with h attention heads is computed as:

$$\text{Multihead}(e_Q, e_K, e_V) = \text{Concat}(head_1, \ldots, head_h)W^O$$
$$\text{where } head_i = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i$$

d_k is the dimension of K_i, which is incorporated for scaling. W^O is the matrix to combine the outputs from multiple attention heads and to produce the final output of multi-head attention mechanism. Note that multi-head self attention uses same inputs to compute queries, keys and values while multi-head encoder-decoder attention uses outputs from the encoder as keys and values, which can be expressed as $Multihead(l, h, h)$. In order to prevent cheating from the future time steps, subsequent masks to the attention layers are incorporated. The fully connected feed-forward network applies linear transformation after adding non-linearity to the outputs of the multi-head attention layer as follows:

$$\text{FFN}(M) = \text{ReLU}(MW_1 + b_1)W_2 + b_2$$
$$\text{where } M = \text{Multihead}(e_Q, e_K, e_V)$$