A New Interpretation of Knowledge Tracing Models' Predictive Performance in Terms of the Cold Start Problem

Rohini Das, Jiayi Zhang, Ryan S. Baker, Richard Scruggs
University of Pennsylvania
rohinidas604@gmail.com, {joycez, rybaker, rscr}@upenn.edu

ABSTRACT

Previous studies on the accuracy of knowledge tracing models have typically considered the performance of all student actions. However, this practice ignores the difference between students' initial and later attempts on the same skill. To be effective for uses such as mastery learning, a knowledge tracing model should be able to infer student knowledge and performance on a skill after the student has practiced that skill a few times. However, a model's initial performance prediction – on the first attempt at a new skill – has a different meaning. It indicates how successful a model is at inferring student performance on a skill from both their performance on other skills and from the difficulty and other properties of the first item the student encounters. As such, it may be relevant to differentiate prediction in these two contexts when evaluating a knowledge tracing model. In this paper, we describe model performance at a more granular level and examine the consistency of model performance across the number of student instances on a given skill. Results from our research show that much of the difference in performance between classic algorithms such as BKT (Bayesian Knowledge Tracing) and PFA (Performance Factors Analysis), as compared to a modern algorithm such as DKVMN (Dynamic Key-Value Memory Networks), comes down to the first attempts of a skill. Model performance is much more comparable by the time the student reaches their third attempt at a skill. Thus, while there are many benefits to using contemporary knowledge tracing algorithms, they may not be as different as previously thought in terms of mastery learning.

Keywords

Knowledge Tracing, Cold Start, Deep Knowledge Tracing, Bayesian Knowledge Tracing, Performance Factors Analysis, Dynamic Key-Value Memory Networks

1. INTRODUCTION

Knowledge Tracing (KT), attempting to measure student knowledge through performance during learning, is a critical component in modern intelligent tutoring systems and adaptive learning systems [18]. These models use students' previous performance to predict their proficiency on latent knowledge and infer their likelihood of success in future attempts within the learning system.

For well over a decade, Bayesian Knowledge Tracing (BKT; [5]) was the dominant algorithm in research on knowledge tracing – it remains the dominant algorithm in use in systems used at scale by students today. Later on, two waves of competing algorithms emerged – a first wave around 2010, including many psychometrically-influenced algorithms such as Performance Factor Analysis (PFA; [17]) and a second wave in the mid-to-late 2010s based on neural networks, including Deep Knowledge Tracing (DKT; [19]) and Dynamic Key-Value Memory Networks (DKVMN; [26]). Work over the last decade has shown that variants of BKT and PFA that take individual differences and timing into account perform better [9, 15, 25]. The current wave of algorithms based on neural networks, such as DKT and DKVMN, have reported further improvements to model fit [12, 26].

The comparisons between these algorithms have generally focused on metrics comparing overall success at predicting on later items, within the learning system applied to held-out students. In these comparisons, multiple large data sets are typically used, but performance is considered evenly across the data set. However, there are some reasons to think this may be a concerning practice. For one thing, even though the data sets used are typically large, these papers generally do not report if samples are large for all skills. Coetzee [4] notes that BKT parameter estimation is more precise for larger data sets than smaller data sets. Furthermore, Gervet [10] concluded that algorithms based on logistic regression, such as PFA, tend to underfit large datasets, while deep learning based algorithms, like DKT, tend to overfit larger datasets.

More concerningly, many data sets used in student modeling have skills which have only been encountered once or twice by many students, either due to stop-out [3] or rarely-tagged secondary skills. Slater and Baker [22] suggest that BKT models cannot be reliably fit unless there is sufficiently large pool of students who have at least three opportunities to practice each skill. As such, large proportions of existing data sets may reflect a seeming special case. Indeed, accurate prediction on these items likely reflects something different than accurate prediction after a student has had more practice. When a student has not yet worked on a skill, predicting their performance at this point represents what is referred to as a "cold start problem" – needing to perform well before having sufficient data for the current student [24]. It is possible that some more recent algorithms may perform better in these situations than earlier algorithms, either by using information from the student's performance on other skills or information on the difficulty or other properties of specific items. However, this better performance may reflect something different than the student's knowledge of the current skill being studied. As such, it may be meaningful to separate out cold start situations (for a given student and skill) from situations where the model has sufficient data to estimate the current skill by itself, when comparing KT algorithms.

In this paper, we study how the performance of three KT algorithms changes, depending on how much data the algorithms have on the current student's performance on the current skill. We compare the classic algorithms BKT and PFA to a more recent neural network-based algorithm, DKVMN, using the ASSISTments 2009-2010 Skill Builder data [7]. Within each model, the predictive performance, determined by AUC ROC (Area Under the Receiver-Operating Characteristic Curve) and RMSE (Root Mean Square Error) was analyzed at students' first through eighth encounter on a skill, reflecting the changes in model performance as students practice a skill more. We conclude with a discussion of the implications of our finding, for both the evaluation and use of knowledge tracing models.

2. METHODS

2.1 Data

In this study, we utilized the ASSISTments Skill Builder 2009-2010 dataset [7], using the updated version which represents an item requiring multiple skills as a single data point [23]. This specific dataset was chosen because it has clearly defined skills and because this dataset had frequently been used to compare KT models in previous research [11, 13, 14, 23, 27].

In the data preprocessing stage, we removed items not linked to any skill. Each student attempt was annotated with how many opportunities to practice the relevant skill(s) the student had encountered so far – i.e., the first instance means the learner is encountering a skill for the first time, the eighth instance indicates that the learner is encountering the skill for the eighth time. The resultant data set consisted of 4,151 students who attempted 16,891 unique problems on 101 skills, resulting in 274,590 responses. While all the skills were included in model training, only the four most common skills are discussed below (see Table 1).

While using the ASSISTments platform, students have to correctly answer *n* problems in a row to achieve mastery of a skill (where *n* is set by the teacher but is usually three) and can only then move on to another skill. Given the design of the platform's three-in-a-row mastery learning approach, there is a drop in sample size as the number of instances increases (a common pattern in adaptive learning systems). There is also attrition due to stop-out, where students stop working on a problem set without mastering it [3]. Table 1 shows that across all four skills, the number of students encountering a specific skill *n* times decreased with instance. Of the four skills, an average 20% and 45% attrition rate is observed on the third and eighth instances, respectively.

Table 1: Number of students per instance in each skill

Skill Name	1	2	3	4	5	6	7	8
Addition and Subtraction Fractions	1353	1066	978	920	836	756	692	625
Addition and Subtraction Integers	1226	1021	790	693	640	579	510	460
Conversion of Fractions Decimals & Percents	1225	1145	1121	1034	982	928	852	781
Equation Solving Two or Fewer Steps	961	877	857	821	795	745	722	690

2.2 Model Construction

We constructed the following three knowledge tracing models with the preprocessed ASSISTments 2009 dataset: BKT, PFA and DKVMN. Each model was implemented with 5-fold student-level cross-validation. For the cross-validation, the dataset was split into five folds at the student level. Four folds were used to train the model and the trained model predicted student's performance in the 5th fold. Each part acted as the test set once. Predictions in the test sets were combined and used to compute AUC and RMSE for each opportunity to practice, within each skill. For comparability, the original skills were used to calculate opportunities to practice rather than the new skills derived by DKVMN. The folds were kept the same across models, reducing the likelihood of randomly favoring one algorithm over another. The metrics were averaged across the four skills in each instance for each model.

BKT and PFA predict students' success at each attempt based on their previous performance on the skill. When predicting a student's success on the first attempt of a new skill, without having any prior data, the initial prediction made by BKT and PFA reflect the overall student performance across the entire (training) data set on that skill, instead of the individual student's knowledge level on the skill. By contrast, the deep learning model DKVMN utilizes all of a student's historical data and exploits the underlying relationships between concepts. This transferability of prediction across skills can be expected to give the algorithm an advantage of making the initial predictions on a newly encountered skill. In fact, [14] studied the effect of interaction among skills in DKT, a closely-related deep learning model, and compared it to BKT. By comparing different approaches to leverage skill data, they concluded that DKT's better performance may be largely due to their use of a student's performance on one skill to predict performance on another skill, whereas skills are strictly separated in BKT. PFA occupies a middle ground, as skills do not directly influence each other, but their combinations in the training set may influence the model parameters found during fitting.

The two widely studied deep learning algorithms DKT and DKVMN utilize neural networks to discover underlying relationships among skills and items when predicting student performance. Because of this, both algorithms have shown significant improvements in model fit compared to traditional algorithms. However, DKT maps the relationships on item level while DKVMN fits a skill model from scratch by considering the relationship among skills and items. Given the purpose of the study is to understand whether transferring information between skills influences a model's accuracy during the first few opportunities, DKVMN is a closer comparison to BKT and PFA within the class of deep learning based KT algorithms.

2.2.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT; [5]) inputs performance into a simple Markov model that is also a Bayesian Network [20]. To fit BKT, we applied BKT-Brute Force [1] to the data set with a floor of 0.01 for all probabilities and a ceiling of 0.3 for guess and slip to avoid model degeneracy [2]. The algorithm produced estimations for guessing, slipping, initial knowledge, and learning transition probabilities for each of the skills, which were then used to predict the probability of success for each student on each opportunity to practice each skill.

2.2.2 Performance Factors Analysis

Performance Factors Analysis (PFA; [17]) is a model that predicts learner performance using a logistic function that models changes in performance through learners' success and failures within a skill. In this study, following the formulas in [17], the basin hopping algorithm was used to fit the model to obtain the optimal parameters.

A set of parameters for success, failure and skill difficulty was derived for each skill, which were then used to compute the probability P(m) that the student would perform correctly, for each student at each opportunity to practice each skill.

2.2.3 Dynamic Key-Value Memory Networks

Developed based on neural networks, Dynamic Key-Value Memory Networks (DKVMN; [26]) employs two matrices that capture states and the relationships between skill and student mastery to predict performance on items and estimate mastery on a set of automatically-derived skills. We utilized code from Zhang et al. [26] to implement the DKVMN model and used the set of parameters that produced the optimal outcome for the ASSISTments 2009 dataset in the study. The model outputs a probability of success for each student at each problem.

3. RESULTS

3.1 AUC Results

Table 2 summarizes the average AUC results for each of the eight opportunities to practice each skill and the combined AUC for opportunities three through eight in the BKT, PFA, and DKVMN models. Additionally, the overall AUC across the first eight opportunities is also reported for the four skills. Note that the overall AUC only includes the targeted four skills in the first eight attempts and therefore, should not be considered to be the overall AUC of the algorithm across the entire data set.

For the first eight instances, a general upward trend is observed in AUC for all three models. Starting at the first instance, the AUC value for BKT is 0.49, PFA is 0.52, and DKVMN is 0.65. At this point, the AUC value for the DKVMN model is much greater than that of other two models, by approximately 0.15. Compared to BKT and PFA, DKVMN is better at making the initial prediction on the very first time a student sees a skill. In fact, at this point, both BKT and PFA are performing at or below chance.

In the following instances, the values of BKT and PFA became closer to the performance of DKVMN. In fact, by the fourth instance, the models' AUC values were fairly similar, having a range of 0.65-0.70. From the fourth opportunity to the eighth, the AUC values increased by 0.02 to 0.06 across skills. Performance stayed similar between algorithms at this point, but DKVMN still tended to achieve slightly higher performance. Across the 3rd-8th opportunities, DKVMN averaged AUC 0.02-0.05 higher than the other two algorithms (0.70 versus 0.68 for BKT and 0.65 for PFA). These trends can be seen in Figures 1-3.

Table 2: Average AUC values in each instance

Model Type	1	2	3	4	5	6	7	8	3-8	All (1-8)
BKT	0.49	0.63	0.68	0.68	0.68	0.68	0.66	0.70	0.68	0.66
PFA	0.52	0.59	0.63	0.65	0.65	0.65	0.66	0.71	0.65	0.63
DKVMN	0.65	0.68	0.70	0.70	0.70	0.69	0.69	0.72	0.70	0.69

Figure 1: AUC results for BKT model across instances

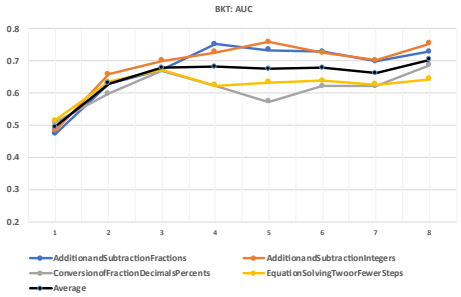


Figure 2: AUC results for PFA model across instances

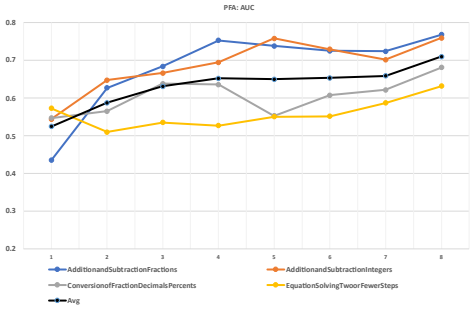
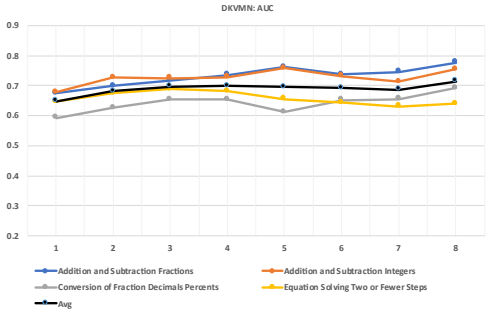


Figure 3: AUC results for DKVMN model across instances



3.2 RMSE Results

Table 3 summarizes the average RMSE results for each opportunity to practice the skills and the combined RMSE for the 3rd-8th opportunities and the 1st-8th opportunities in the BKT, PFA, and DKVMN models. Again, the RMSE reported in the table only considers the targeted four skills in the first eight opportunities.

The RMSE demonstrates a downward trend across the first eight opportunities in all three models. As RMSE measures the difference between actual and predicted values, lower RMSE values indicate more accurate predictions. In the first instance, the RMSE value for BKT is 0.49, PFA is 0.51, and DKVMN is 0.47. As the RMSE value for DKVMN is better than that of BKT and PFA, similar to the AUC value, DKVMN is better able to predict student knowledge at the first attempt (0.02 better than BKT and 0.04 better than PFA).

In the following instances, the values of BKT and PFA became closer to the performance of DKVMN. In fact, by the fourth instance, the models' RMSE values were fairly similar, having a range of 0.43-0.46. From the fourth opportunity to the eighth, the RMSE values in all three models roughly remained the same across

skills. Across the 3rd-8th opportunities, DKVMN's average RMSE was similar to BKT and 0.02 lower than PFA (0.44 versus 0.44 and 0.46). These trends can be seen in Figures 4-6.

Table 3: Average RMSE values for all models in each instance

Model Type	1	2	3	4	5	6	7	8	3-8	All (1-8)
BKT	0.49	0.46	0.44	0.44	0.44	0.44	0.45	0.44	0.44	0.45
PFA	0.51	0.48	0.46	0.46	0.47	0.46	0.48	0.46	0.46	0.47
DKVMN	0.47	0.45	0.44	0.43	0.44	0.44	0.45	0.43	0.44	0.45

Figure 4: RMSE results for BKT model in each instance

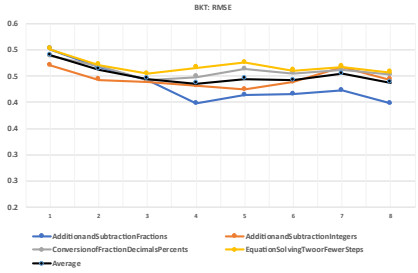


Figure 5: RMSE results for PFA model in each instance

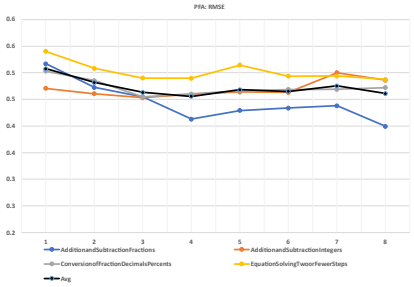
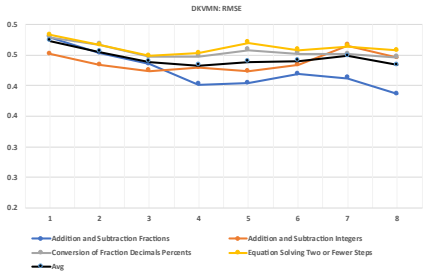


Figure 6: RMSE results for DKVMN model in each instance



4. CONCLUSION AND DISCUSSION

In the last few years, there has been an explosion of interest in new variants to knowledge tracing that achieve higher predictive performance using neural networks. However, this work has generally not yet explored where and when these algorithms perform better, and what the implications are for using these models in practice. More specifically, previous practices have averaged predictions across students' entire learning history, ignoring the difference between the earliest work and later work on a skill.

In this study, we examined the performance of three KT models, BKT, PFA, and DKVMN, across students' history of work on specific skills, and compared how the three models differ in predictive accuracy during the earliest and later opportunities to practice each skill. With all eight opportunities considered together, DKVMN outperformed BKT and PFA in both AUC and RMSE. However, DKVMN's better performance appears to be largely due to its initial prediction on the first attempt on a skill, in which DKVMN 's AUC was 0.16 higher than BKT and 0.13 higher than PFA, and RMSE was 0.02-0.04 better. After the first attempt, BKT and PFA's predictive performance improved substantially, and model performance became closer across the three algorithms after the third attempt, though DKVMN remained slightly better.

The results suggest that much of the difference in performance between these algorithms is due to DKVMN's ability to make more accurate initial predictions by using factors other than mastery of the current skill, such as past performance on other skills and other students' performance on the same item. In other words, a substantial amount of the difference between algorithms appears to be due to factors other than estimating mastery of the current skill the student is working on, from their performance on that skill. This may be especially true in datasets where students stop-out on specific skills [3], or where the skill model is added to or modified after the system is built. In these cases, many student/skill combinations may only occur once or twice and having relatively higher performance on the first attempt will inflate AUC and RMSE values for models such as DKVMN. This raises the question of what the application is for having better knowledge prediction at the first time when a student sees a new skill. This type of improvement in prediction may be useful to systems that decide which skill a student should work on next (i.e., [6, 28]) but less useful in systems that have a predefined order of skills for the student to work on (i.e. [5, 8]) and the student does not move on until they have demonstrated mastery on the current skill.

Given the difference in predictive performance between situations, it may be appropriate to separate cold start situations (for a given student and skill) from situations where the model has sufficient data to estimate the current skill by itself when comparing KT algorithms. Specifically, we propose that the calculation of predictive metrics should separate the predictions on the initial two opportunities to practice each skill from the rest. Adopting this approach will increase our ability to interpret the difference between algorithms and understand how much better a specific algorithm will be for specific use cases.

Two limitations to the current analyses can be addressed in future work. First, our recommendations may not be meaningful for all learning systems where contemporary KT is used. In specific, some systems may not have skill models at all, and may never intend to make inferences at the level of interpretable skills. Although these systems typically use an entirely different family of KT models (i.e. [16, 21]), our recommendations would not be relevant in these cases. Second, we have only investigated these issues in the context of a single system and a set of skills for which there is extensive data, and for three algorithms; the generalizability of the findings presented here should be further investigated, using data from other learning systems where, for instance, the granularity of the skills differs. However, only a limited effort is needed to separate practice on early learning opportunities from later learning opportunities when calculating model AUC/RMSE. Therefore, it may be warranted to adopt this approach and see whether practical differences are found for other contexts and algorithms as well.

Overall, we find initial evidence that one key factor leading to better performance for DKVMN compared to earlier algorithms is its performance in situations before a student has had a significant opportunity to work on a skill. This result leads to recommendations in how to better evaluate KT algorithms and suggests that the benefits of this algorithm may be greater for some applications (deciding which skill a student should work on next) than others (deciding if a student has reached mastery in the current skill they are working on). From the results of this study, future studies conducting research involving KT models may find it useful to calculate performance separately for a student's initial performance and their later performance on a skill; this would provide researchers with more information on how their models are working, and where their greatest benefits and potential are.

5. REFERENCES

[1] Baker, R.S.J. d. et al. 2010. Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor. *User Modeling, Adaptation, and Personalization* (Berlin, Heidelberg, 2010), 52–63.

[2] Baker, R.S.J.D. et al. 2008. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. *Intelligent Tutoring Systems* (Berlin, Heidelberg, 2008), 406–415.

[3] Botelho, A.F. et al. 2019. Refusing to try: Characterizing early stopout on student assignments. *Proceedings of the 9th International Conference on Learning Analytics & Knowledge* (New York, NY, USA, Mar. 2019), 391–400.

[4] Coetzee, D. 2014. Choosing sample size for knowledge tracing models. *CEUR Workshop Proceedings* (2014), 117–121.

[5] Corbett, A.T. and Anderson, J.R. 1995. Knowledge Tracing : Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*. (1995), 253–278.

[6] Craig, S.D. et al. 2013. The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors. *Computers and Education*. 68, (2013), 495–504.

[7] Feng, M. et al. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*. 19, 3 (2009), 243–266.

[8] Feng, M. et al. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*. 19, 3 (Aug. 2009), 243–266.

[9] Galyardt, A. and Goldin, I. 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining*. 7, 2 (2015), 83–108.

[10] Gervet, T. et al. 2020. When is Deep Learning the Best Approach to Knowledge Tracing? *Journal of Educational Data Mining*. 12, 3 (2020), 31–54.

[11] Gong, Y. et al. 2010. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. *Intelligent Tutoring Systems* (Berlin, Heidelberg, 2010), 35–44.

[12] Khajah, M. et al. 2016. How deep is knowledge tracing? *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016* (2016), 94–101.

[13] Minn, S. et al. 2018. Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing. *2018 IEEE International Conference on Data Mining (ICDM)* (Singapore, Nov. 2018), 1182–1187.

[14] Montero, S. et al. 2018. Does deep knowledge tracing model interactions among skills? *Proceedings of the 11th International Conference on Educational Data Mining* (2018).

[15] Pardos, Z.A. and Heffernan, N.T. 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. *In International Conference on User Modeling, Adaptation, and Personalization* (2010), 255–266.

[16] Pavlik, P. et al. 2008. Using Optimally Selected Drill Practice to Train Basic Facts. *Intelligent Tutoring Systems* (Berlin, Heidelberg, 2008), 593–602.

[17] Pavlik, P.I. et al. 2009. Performance Factors Analysis – A New Alternative to Knowledge Tracing. *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (Brighton, England, 2009), 531–538.

[18] Pelánek, R. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*. 27, 3–5 (2017), 313–350.

[19] Piech, C. et al. 2015. Deep knowledge tracing. *Advances in Neural Information Processing Systems*. 2015-Janua, (2015), 505–513.

[20] Reye, J. 2004. Student Modelling based on Belief Networks. *International Journal of Artificial Intelligence in Education,*. 14, (1) (2004), 63–96.

[21] Settles, B. and Meeder, B. 2016. A trainable spaced repetition model for language learning. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*. 4, (2016), 1848–1858.

[22] Slater, S. and Baker, R.S. 2018. Degree of error in Bayesian knowledge tracing estimates from differences in sample sizes. *Behaviormetrika*. 45, 2 (Oct. 2018), 475–493.

[23] Xiong, X. et al. 2016. Going Deeper with Deep Knowledge Tracing. *Proceedings of the 9th International Conference on Educational Data Mining*. (2016), 545–550.

[24] Yang, T.-Y. et al. 2019. Active Learning for Student Affect Detection. *Proceedings of The 12th International Conference on Educational Data Mining* (2019), 208–217.

[25] Yudelson, M. V. et al. 2013. Individualized bayesian knowledge tracing models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 7926 LNAI, (2013), 171–180.

[26] Zhang, J. et al. 2017. Dynamic key-value memory networks for knowledge tracing. *26th International World Wide Web Conference, WWW 2017* (2017), 765–774.

[27] Zhang, J. et al. 2017. Dynamic Key-Value Memory Networks for Knowledge Tracing. *Proceedings of the 26th International Conference on World Wide Web* (Perth Australia, Apr. 2017), 765–774.

[28] Zou, X. et al. 2019. Towards Helping Teachers Select Optimal Content for Students. *International Conference on Artificial Intelligence in Education* (Cham, 2019), 413–417.