Recommendation for Effective Standardized Exam Preparation

Hyunbin Loh, Dongmin Shin, Seewoo Lee, Jineon Baek, Chanyou Hwang, Youngnam Lee,
Yeongmin Cha, Soonwoo Kwon, Juneyoung Park, Youngduck Choi
Riiid! AI Research
Republic of Korea
{hb.loh,dm.shin,seewoo.lee,jineon.baek,cy.hwang,yn.lee,ymcha,sw.kwon,juneyoung.park,youngduck.choi}@riiid.co

ABSTRACT

Finding an optimal learning trajectory is an important question in educational systems. Existing Artificial Intelligence in Education (AiEd) technologies mostly used indirect methods to make the learning process efficient such as recommending contents based on difficulty adjustment, weakness analysis, learning theory, psychometric analysis, or domain specific rules.

In this study, we propose a recommender system that optimizes the learning trajectory of a student preparing for a standardized exam by recommending the learning content(question) which directly maximizes the expected score after the consumption of the content. In particular, the proposed *RCES* model computes the expected score of a user by effectively capturing educational effects. To validate the proposed model in an end-to-end system, we conduct an A/B test on 1713 real students by deploying 4 recommenders to a real mobile application. Result shows that *RCES* has better educational efficiencies than traditional methods such as expert designed models and item response theory based models.

CCS CONCEPTS

• **Applied computing** → **E-learning**; *Computer-assisted instruction.*

KEYWORDS

Education; Personalized learning; Intelligent tutoring system; Recommender Systems

ACM Reference Format:
Hyunbin Loh, Dongmin Shin, Seewoo Lee, Jineon Baek, Chanyou Hwang, Youngnam Lee, Yeongmin Cha, Soonwoo Kwon, Juneyoung Park, Youngduck Choi. 2021. Recommendation for Effective Standardized Exam Preparation. In *LAK21: 11th International Learning Analytics and Knowledge Conference (LAK21), April 12–16, 2021, Irvine, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3448139.3448177

1 INTRODUCTION

Standardized testing is an integral part of modern K-12 educational system around the world. As of today, it is estimated that more than 1 billion students are enrolled in the K-12 system worldwide according to the UIS Unesco dataset. In the case of the Scholastic Assessment Test (SAT), more than 2.1 million students in the class of 2018 took the exam. This number is the largest number ever and also a 25% increase from the number of students in the class of 2017. The Test of English for International Communication (TOEIC) is a standardized test that around 6 million people take every year. More than 160 countries are using the TOEIC score to evaluate English skills as a quantified value. For high performance in these standardized tests, it is important for students to follow an efficient learning process. Conventional approaches to the preparation of standardized examinations, such as enrolling in a class, or using textbooks, are often ineffective. Since students have different needs and abilities (knowledge and learning curves), personalized learning provides greater learning efficiency than conventional approaches. However, expert based personalized learning is not accessible to most students since it requires high cost and is not scalable.

It is therefore natural to envision a new large scale Artificial Intelligence system that makes personalized test preparation affordable specifically to the students facing these examinations. Artificial Intelligence in Education (AIEd) aims to combine models for the target study domain, pedagogical knowledge and the student into a system ,which provides real-time feedback to the student. Existing AIEd approaches for standardized testing focus on analyzing the students' abilities. For instance, models in Item Response Theory and Knowledge Tracing predicts the correctness probability when a question is given to a student [6, 10]. Also, models in Computerized Adaptive Testing focus on designing assessment scores and finding efficient questions to estimate the score [31]. However, analyzing the students' ability is insufficient to achieve the end goal of optimizing the learning process. Correctness probabilities and assessment scores doe not directly tell which learning content is the optimal choice to study at the moment for each student. As in most areas, deep learning models applied [16, 17, 22] in AIEd are showing high performance, but these models also focus on the analysis of students' abilities, not recommendations for efficient learning.

We introduce the *Santa AI system* that recommends the most efficient learning content to a student preparing for the TOEIC test. *Santa* is an AI tutor for English education, which is available via Android and iOS applications and has more than 2 million signed up users. To maximize the actual score of a student, we consider a question recommendation algorithm *RES* that suggests the question with highest expected score after response. We also present *RCES*, which is an effective and efficient recommender that suggests the question with highest expected score after consumption of the content. Our main contributions are summarized as follows:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
LAK21, April 12–16, 2021, Irvine, CA, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8935-8/21/04…$15.00
https://doi.org/10.1145/3448139.3448177

- We propose an educationally effective and computationally efficient recommender *RCES* by introducing the education effect term.
- We empirically show that *RCES* gives better student score increases compared to traditional methods.
- We conduct an A/B test on 1713 real students by deploying 4 recommenders to a real mobile application Santa.

2 RELATED WORKS

Standardized testing requires test takers to answer selected questions from the same question set, in the same environment, where the results are scored in a standardized rule [24]. In [25], the authors propose three main objectives for standardized tests: to evaluate and compare test takers, to improve teaching and evaluate the instruction process. Using results on student performances on standardized test, the instruction process of educational programs are evaluated as well [2]. However, applying a unified setting for all test takers of a standardized test draws problems such as unfairness of assessment for students from different backgrounds [7, 9], inefficient education where high costs are only used for score improvement [14, 20, 21] and unhelpful feedback to the tutors for teaching improvement [12].

There has been various attempts to use AI technologies in education, which includes analyzing student knowledge (knowledge tracing) [26], providing appropriate educational contents (content recommendation) [18] and schematizing relations between various concepts (knowledge graphs) [34]. Item Response Theory (IRT) is one of the most widely used methods for examining student understandings, by questions that extracts the most student information [10, 33]. Knowledge Tracing (KT) is the approach to model and track the change of student knowledge status by time. Bayesian knowledge tracing [6] is a branch of KT that models student's understanding in a hidden Markov model as a set of binary variables, which is updated using the student's response to given questions. Collaborative filtering is a machine learning technique introduced in KT that extract knowledge of students based on their response correctness prediction [1, 3, 30]. To improve the performance of response correctness prediction, deep learning-based methods such as LSTM [23], memory network [35], Bi-LSTM with attention [17] and self-attention [22] are used. However, knowledge tracing models that require expert annotated knowledge components is not the scope of our research.

There has been some progress on item recommendation in education also with AI technologies. The works aim to enhance student's engagement, memory, or the learning efficiency of various educational contents. In [29], the authors used a Half-Life Regression (HLR) human memory model to suggest words that help students improve their long term memories. Based on the suggested memory model, the authors conducted experiments with real users of *Duolingo* and showed that their model succeeded to increase daily retention, such as new lesson engagement rate. There are also attempts to apply reinforcement learning techniques for instructional sequencing. In [8], the authors give a detailed review of reinforcement learning techniques applied to various domains in education. In [27], the expected number of words recalled or the likelihood of recalling all words in the database are used as rewards. The authors of [27] implemented student simulators based on three memory models, including HLR. They compared the performance of their scheduling algorithm with four other baseline recommendation systems, such as Leitner system. In [36], the authors developed a lecture recommendation model with Long Short Term Memory (LSTM) cell [11], which suggests an efficient learning trajectory for each student.

3 RECOMMENDATION FOR EFFICIENT STANDARDIZED EXAM PREPARATION

In this section, we present *Recommendation for Expected Score(RES)* and *Recommendation for Calibrated Expected Score(RCES)* which suggests the question with the highest expected score and calibrated expected exam score to the user at each time step respectively. The calibration term in RCES captures the educational effect from solving and studying the recommended question. The experiment shows that this term resolves adversarial sequences that can occur from RES.

3.1 Setup

Let Q be the total set of questions to be recommended to the students. We denote the interaction $I_u(q)$ as the response of the student u to the question q. The value of $I_u(q)$ is 1 if u responds correctly to q and 0 otherwise. Call a sequence of elements in $Q \times \{0, 1\}$ a *question-response sequence* and let Seq_{QR} be the set of all such possible sequences. A sequence

$$(q^1, r^1), \ldots, (q^t, r^t)$$

in Seq_{QR} denotes that a student answered the question q^i with response r^i in the successive order of index $1 \leq i \leq t$. The concatenation of two question-response sequences R_1 and R_2 is written as (R_1, R_2).

In this paper, the *score function* is a model that approximates the official TOEIC test score from the question-response sequence of the student. Likewise, given the question-response sequence I_u of student u, the *correctness probability function* $p(q|I_u)$ predicts the probability of the student responding correctly to q. Formally, the value $p(q|I_u)$ estimates the probability $\mathbb{P}[I_u(q) = 1|I_u]$.

Note that both RES and RCES depend on the score function S and the correctness probability function p. For p, we use the Matrix Factorization [13] model introduced in [15]. With the question-response sequences of the users, we find the decomposition $X = LR^T$ that minimizes the Binary Cross Entropy (BCE) loss with a Frobenius norm regularization, where $L = (L_{uj})$ represents the understanding of a student u on a hidden concept j and $R = (R_{qj})$ represents the contribution of a hidden concept j to the question q. The entry $X = (X_{uq})$ represents the understanding of a student u on a question q, and the response correctness probability $p(q|I_u) = P_{uq}$ is computed using X_{uq} based on the variation of the M2PR latent trait model in IRT [19].

We use two models for the score function S, where one model is used for recommendations in the experiment, and the other model is used for evaluation after the experiment. The first model is based on Bi-directional recurrent neural networks with Long Short Term Memory cell (Bi-LSTM). LSTM was first introduced in [11] for machine translation and Bi-LSTM [28] is an improved version of LSTM

Algorithm 1 Recommender with Expected Score

INPUT: student u, base question-response sequence I_u^0, the set of all questions Q, number of questions to be recommended k, subsampling size C.
$Q^0 \leftarrow Q(I_u^0)$
for $t \leftarrow 0$ to $k-1$ **do**
  Choose subset Q' of $Q - Q^t$ with size $|Q'| = C$ uniformly random
  $q^* \leftarrow \text{argmax}_{q \in Q'} [p(q|I_u^t)S(I_u^t, (q,1)) + (1 - p(q|I_u^t))S(I_u^t, (q,0))]$
  $Q^{t+1} \leftarrow Q^t \cup \{q^*\}$
  Ask the student question q^* and get response $r^* = I_u(q^*)$
  $I_u^{t+1} \leftarrow (I_u^t, (q^*, r^*))$
end for

which runs both forwards and backwards. For our purpose, instead of sentences, we use a sequence of question-response pairs as an input and the model predicts score by estimating relations between each pair. The second model is based on Transformer. Transformer was first introduced in [32], which replaced the recurrent layers in the encoder-decoder architecture by multi-head self-attention layers. The architecture is widely used in Natural Language Processing tasks since the training process can be parallelized, and the model shows higher performance than existing seq2seq models. The model used in this experiment is pre-trained by user-question response correctness and fine-tuned by score labels [4]. Both models use the latest 200 responses as input. The architecture of the models are illustrated in Figure 1 and 2.

3.2 Recommendation with Expected Score

We describe the question recommender with expected score (*RES*) using the computations of the correctness probability function p and the score function S. Let a student u have solved t questions and the responses are recorded as a questions-response sequence

$$I_u^t = ((q^1, r^1), \ldots, (q^t, r^t)).$$

For a question-response sequence $I_u^t = ((q^1, r^1), \ldots, (q^t, r^t))$, we denote the sequence of questions (with no responses attached) as $Q(I_u^t) := (q^1, \ldots, q^t)$. When a question $q = q^{t+1}$ is provided to the student u, then the expected score of u is computed by:

$$\begin{aligned} &\mathbb{E}_p[S(I_u^t, (q^{t+1}, r^{t+1}))] \\ &= p(q^{t+1}|I_u^t)S(I_u^t, (q^{t+1}, 1)) + (1 - p(q^{t+1}|I_u^t))S(I_u^t, (q^{t+1}, 0)) \end{aligned} \quad (1)$$

for given score function S and a correctness probability function p. We compute the expected score for all possible candidates of q^{t+1}, and choose the one with the highest expected score. Since the number of possible questions to recommend can be large, we randomly and uniformly sample C questions from Q to make the computation feasible. The algorithm is illustrated in Algorithm 1.

3.3 Recommendation with Calibrated Expected Score

In the previous section, we introduced the question recommendation algorithm RES that suggests the question with the highest expected score after consumption. However, RES does not accurately capture the educational effect from the user experience. In Santa, the students learn not only by solving questions, but also by reading the explanations to the responses. Therefore, maximizing the expected score is to maximize the score before reading the explanations. Then, RES can tend to recommend questions that increase the expected score, but has less educational effect. For instance, the recommender can recommend only easy questions so that the student can respond correctly with high probability and raise the score, but has less effect on the true skill of the student.

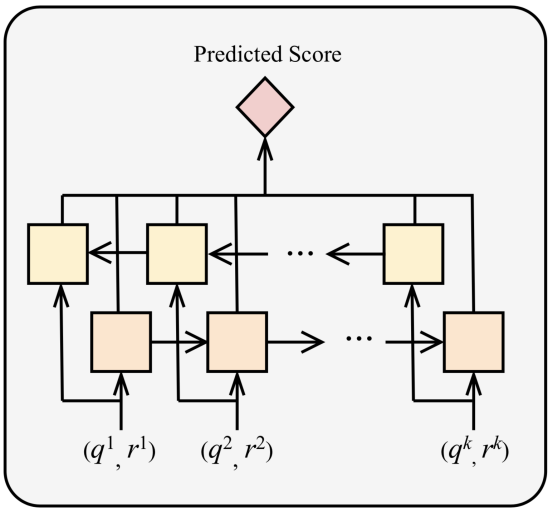

Figure 1: Score prediction model with Bi-LSTM cells

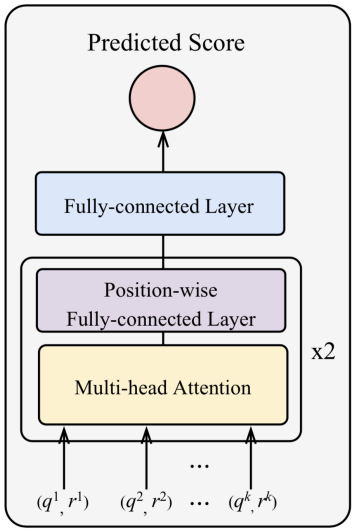

Figure 2: Score prediction model based on Transformer

We adjust the RES by introducing an additional constant term α, so that the new model can maximize the score after reading the comments. α is a constant that represents the educational effects that occur after solving questions, such as reading the explanations. The value of the model α is determined by the user and item pair. In this experiment, we heuristically estimate α as a constant value by observing the average correctness rate of responses on questions that are given more than one time to the same student. From this point of view, we define *Calibrated Expected Score* (CES) of a given

Algorithm 2 Recommender with Calibrated Expected Score

INPUT: student u, base question-response sequence I_u^0, the set of all questions Q, number of questions to be recommended k, subsampling size C.
$Q^0 \leftarrow Q(I_u^0)$
for $t \leftarrow 0$ to $k-1$ **do**
  Choose subset Q' of $Q - Q^t$ with size $|Q'| = C$ uniformly random
  $q^* \leftarrow \text{argmax}_{q \in Q'} [(\alpha + (1-\alpha)p(q|I_u^t))S(I_u^t, (q,1)) + (1-\alpha)(1 - p(q|I_u^t))S(I_u^t, (q,0))]$
  $Q^{t+1} \leftarrow Q^t \cup \{q^*\}$
  Ask the student question q^* and get response $r^* = I_u(q^*)$
  $I_u^{t+1} \leftarrow (I_u^t, (q^*, r^*))$
end for

question-response sequence I_u^t as

$$\begin{aligned} &\alpha S(I_u^t, (q^{t+1}, 1)) + (1-\alpha)\mathbb{E}_p[S(I_u^t, (q^{t+1}, r^{t+1}))] \\ &= (\alpha + (1-\alpha)p(q^{t+1}|I_u^t))S(I_u^t, (q^{t+1}, 1)) \\ &+ (1-\alpha)(1 - p(q^{t+1}|I_u^t))S(I_u^t, (q^{t+1}, 0)) \end{aligned} \tag{2}$$

when a new question q^{t+1} is suggested. The CES is a weighted sum of the score with correct response of q^{t+1} and the expected score, with weights α and $1-\alpha$. The first term represents the case when a student answers q^{t+1} correctly by probability α after consuming the educational feedback followed by the question.

Note that the suggested recommenders use greedy approaches to optimize the recommendation. However, searching multiple steps is computationally infeasible, because the search space grows exponentially. In our case, there is a time constraint of 1 second for each recommendation. Using our *Bi-LSTM* model on an *EC2 C5.2Xlarge AWS instance with 4vCPUs and 8GM memory*, a single time step recommendation takes an average of 0.521 seconds. This cost constraint forces the recommender algorithm to search only a single step for each request.

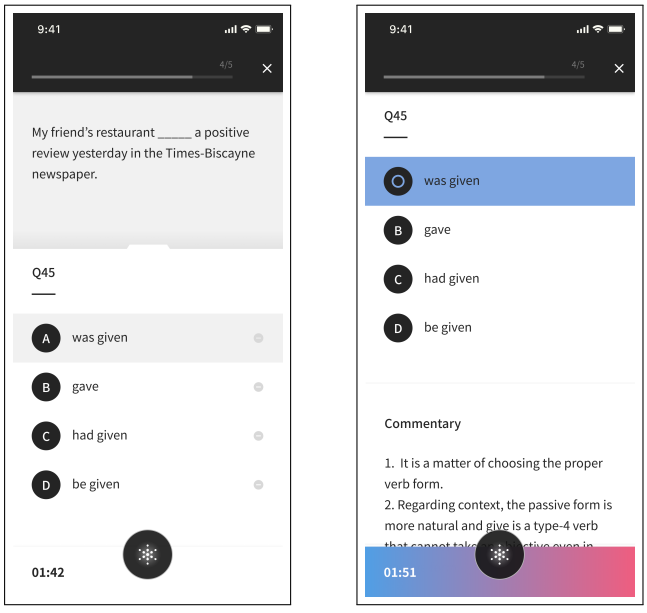

Figure 3: User interface of Santa

4 DEPLOYMENT AND RESULTS

We deployed RCES to *Santa*, which is a mobile application with 1M users for preparing the TOEIC exam. The actual flow chart of the system is described in Figure 4. The score and correctness probability models S and p are computed from the question-response sequence I_u of a student in real time. Using the inferred values, the question q^* is selected by Algorithm 2. When the student responds to the question q^* with response r^*, the question-response pair (q^*, r^*) is added to I_u and the whole process is repeated.

4.1 Setups

4.1.1 TOEIC and Santa. *Test of English for International Communication* (TOEIC) is an exam established by Educational Testing Service (ETS). In particular, *TOEIC Listening and Reading Test* measures English listening and reading abilities by 200 multiple-choice questions. The score ranges from 0 to 990, with a score gap of 5. The exam consists of seven parts with different types of questions, where Part 1 to 4 are Listening Comprehension (LC) and Part 5 to 7 are Reading Comprehension (RC) questions.

Santa is a mobile AI tutor service for the TOEIC Listening and Reading Test preparation. Figure 3 shows the user interface of Santa, where users study by solving questions and reading educational feedback (such as explanations or commentaries on questions) after solving each question. Currently, 2 million users have signed up for the service from Android and iOS. The Table 1 shows the number of questions for each part in Santa.

Part	1	2	3	4	5	6	7
# of questions	1476	1695	1983	1941	5789	579	1714

Table 1: Number of questions in Santa

For user response and score prediction, Santa uses two types of data described in Table 2 and Table 3. The first data is the user-question response data, which includes the following columns: user_id, question_id, response and timestamp of the response. This data is called *EdNet* and is open to the public [5]. The second data is the TOEIC exam score data reported by users of Santa. The columns used for this paper are: user_id, score and timestamp_test.

4.1.2 Correctness probability model and Score model. The proposed recommenders in this experiment are based on computations of the correctness probability model and the score model. We describe the details and performances of models that compute the correctness probability $p(q|I)$ of a question q and the predicted score $S(I)$ from

user_id	question_id	response	timestamp
1	1	A	03170511
2	5	B	03170511
1	42	A	03170512
1	53	D	03170523
5	36	C	03170537

Table 2: User-question response data

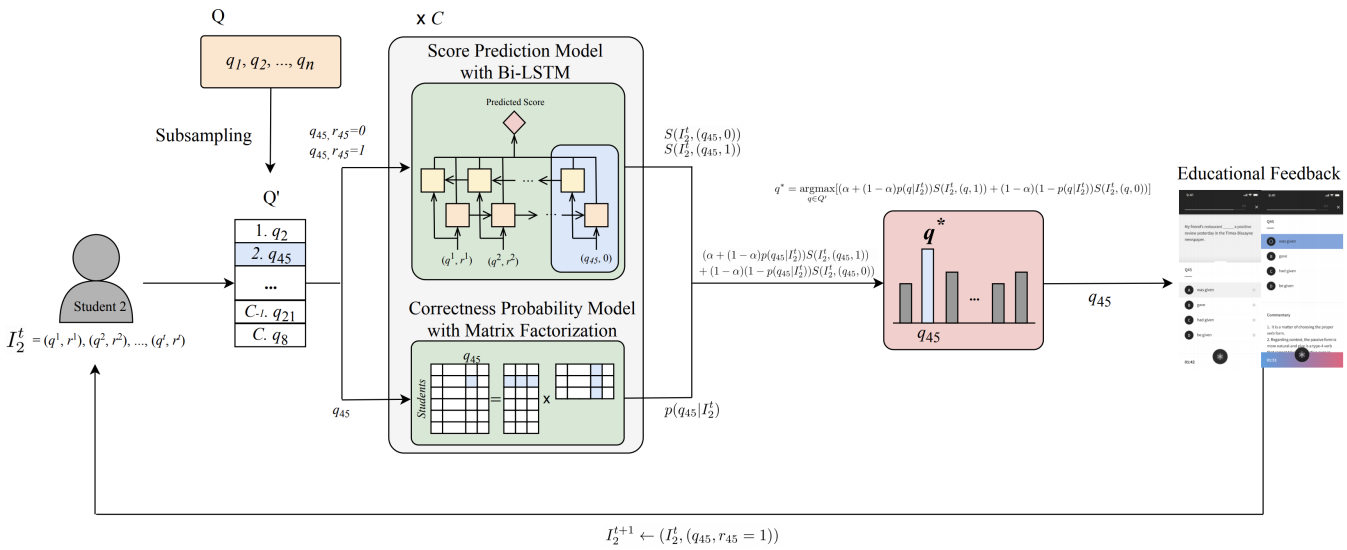

Figure 4: Flowchart

user_id	score	timestamp
1	810	03120900
2	785	03120900
1	900	03260900
1	885	04030900
5	640	04170537

Table 3: Score data

the user's question-response sequence I. The models are trained with the data described in Section 4.1.1.

In this experiment, we compute the correctness probability $p(q|I_u) = \phi(X_{uq})$ by the equation

$$\phi(x) = \phi_a + \frac{1-\phi_a}{1+e^{-\phi_c(x-\phi_b)}}$$

for $(\phi_a, \phi_b, \phi_c) = (0.25, 0.5, 10)$. Then we solve the optimization problem

$$\min_{L,R} \sum_{(u,q)\in\Omega} \mathcal{L}_{uq} + \frac{\mu}{2}(||L||_F^2 + ||R||_F^2)$$
$$\text{s.t. } 0 \le L_{uj} \le 1, 0 \le R_{qj} \le 1, p(q|I_u) = \phi(X_{uq}), \sum_j R_{qj} = 1$$

where Ω is the set of student index-question index pairs that are observed. Here $\mathcal{L}_{uq}$ is a BCE loss

$$\mathcal{L}_{uq} = -Y_{uq}\log(p(q|I_u)) - (1-Y_{uq})\log(1-p(q|I_u))$$

and $||\cdot||_F$ is a Frobenius norm of a matrix. Y_{uq} is an actual response of student u on a question q, which is 1 if correct and 0 otherwise. We use the projected Stochastic Gradient Descent (SGD) method to solve the matrix factorization problem. This model is trained over 100M responses of 1M users. The AUC and ACC of the model are 0.7163 and 0.6882, respectively.

For score prediction in the experiment, we use a Bi-LSTM model that predicts the score from the responses of a user. The model has 5 layers with 32 hidden nodes, and we compute the average of all outputs of each step and predict the score by using two fully-connected layers with dimension 32 and 16. For training, we used 5K *real* TOEIC exam scores. The Mean Absolute Error (MAE) of the model is 61. We also evaluate the final score differences using another model based on Transformers, with MAE 65.

4.1.3 Recommenders. We deploy the following four recommenders to Santa with subsampling size $C = 200$ for RES and RCES:

(A) (Expert) Domain expert weighted sample
(B) (RIRT) IRT (Maximize Fisher Information)
(C) (RES) Maximize expected score
(D) (RCES) Maximize calibrated expected score

Expert is a random question sampler based on the allocation of studying time for each part on Santa suggested by TOEIC experts. Such weights are given in the Table 4.

Part	1	2	3	4	5	6	7
Weight	0.05	0.15	0.18	0.15	0.22	0.05	0.20

Table 4: Expert-annotated weights

RIRT chooses questions in each part that maximizes the *Fisher information function*

$$I(p) = \sum_{r=1}^{c} p_r(1-p_r) = 1 - \sum_{r=1}^{c} p_r^2$$

where p_i is the probability that a user choose i-th choice. This is a common method used by recommenders based on IRT. Note that

the Fisher information function is maximized when $p_i = 1/c$ for all $1 \leq i \leq c$.

RES chooses the question that maximizes the expected score function, as introduced in Section 3.2. *RCES* uses the Algorithm 2 that maximizes the calibrated expected score function described in Section 3.3. The constant α in RCES is computed from the user response data of Santa. For new users, Santa only recommends questions that are not solved before. However, once a user solves every question in one of the seven parts, any question from that particular part is recommended. We observe the average correctness rate when users face a question for the second time. Table 5 shows the basic statistics. α in RCES is a model that is computed from the pair u and i. In this experiment, we use the mean of user correctness rates for reviewed questions, which is 0.7889.

statistic	value
count	937
mean	**0.7889**
std	0.1196

Table 5: α statistics

4.1.4 Experiment. We randomly assigned one of the four recommenders to 1713 users. To maintain a similar user experience for the users, the users were given 20 percent of the questions from the assigned recommender from the four recommenders. The other questions were given by the same default recommender for all 1713 users. These users in total solved 277649 questions, where 54165 questions were solved from the recommenders and 223484 from the default recommender. Then, we filter users who finished solving 600 questions, where the first 200 responses are used as the inputs for the score models and the next 400 responses to track the score trajectory. The first 200 responses are required since we use score models that use the latest 200 responses as input for the experiment.

4.2 Results

The experiment was conducted from 26th Jul 2019 to 11th Aug 2019. Four different recommenders introduced in Section 4.1.3 were distributed to 1713 real users of Santa. Table 6 shows the number of participants that solved more than 600 questions from each group.

4.2.1 Experiment Results. We evaluate the user scores from the first 200 responses as the initial score and compare with the scores after solving 400 additional questions. The results are shown in the Table 6. We use a Bi-LSTM model S_{BiLSTM} and a *Transformer*-based model S_{TF} to compute the estimated average increase of the user score after recommendation. We use the score model S_{BiLSTM} for the recommendation process in Algorithm 1 and 2. Then, we use a different score model S_{TF} based on Transformer that was developed independently. The model S_{TF} is developed to make an evaluation of the experiment that is independent to the recommendation method which maximizes the score function, but not the true skill of the student. The architecture of S_{TF} is depicted in Figure 2. This model S_{TF} uses the self-attention mechanism to put more emphasis on relevant interactions for prediction.

Recommender	Expert	RIRT	RES	RCES
total participants	373	653	344	343
total responses	8241	29628	9032	7264
users with 600 responses	34	100	33	30
avg ΔS_{BiLSTM}	-8.9942	10.3263	15.5580	**35.9618**
avg ΔS_{TF}	-14.9841	12.2864	10.9489	**40.1164**
area under S_{TF} curve	-3902.1766	2512.0631	1640.5096	**6552.4028**

Table 6: Improvements by Score Model

Table 6 shows that RCES outperforms the other three recommenders in terms in both metrics. Note that RES showed good performance for the Bi-LSTM model, but not for the Transformer model. This shows that Algorithm 1 only increases the score from a specific Bi-LSTM model. We describe the details of this adversarial effect.

4.2.2 Adversarial effects of RES. RES maximizes the score of the student after solving the question, and RCES maximizes the score after reading the explanations additionally. Therefore, RCES is a recommender which is designed for the particular user experience of the system. We tracked the change of average scores over responses for each recommender. Figure 5 shows the score changes until the users solves 400 questions, using the score function used for recommendation. The results are compared to two other baseline recommenders, where each are 1) domain expert designed random sampler and 2) item response theory based recommender. Then, we also track the score changes of the 4 models using a different score model. The score changes are shown in Figure 6. The result shows that RES, RCES works better than the baseline methods for maximizing the score function S using the Bi-LSTM model. This is a natural result since RES and RCES literally gives questions with highest expected S after the consumption of the question and after reading the comments. However, RES does not show significant results when using the Transformer model.

Recommender	Expert	RIRT	RES	RCES
0-0.1	0.0541	0.0847	0.0532	0.0635
0.1-0.2	0.0494	0.1291	0.0486	0.0481
0.2-0.3	0.0463	0.1860	0.0578	0.0417
0.3-0.4	0.0583	0.2097	0.0416	0.0546
0.4-0.5	0.0548	0.2054	0.0399	0.0725
0.5-0.6	0.0696	0.1116	0.0497	0.0888
0.6-0.7	0.0848	0.0206	0.0700	0.1166
0.7-0.8	0.1093	0.0204	0.1139	0.1285
0.8-0.9	0.1871	0.0275	0.2244	0.1950
0.9-1.0	0.2862	0.0049	0.3007	0.1906

Table 7: Correctness probability distributions

We also investigate the predicted correctness probability of questions when the questions were recommended, for each recommender. We divide the suggested questions into 10 groups with correctness probabilities in $0 \sim 0.1, 0.1 \sim 0.2, \ldots, 0.9 \sim 1$, and check the proportion of questions for each group. The results are

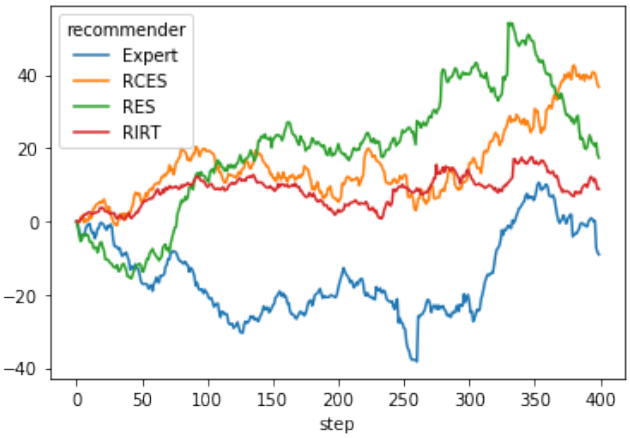

Figure 5: Average score changes by Bi-LSTM

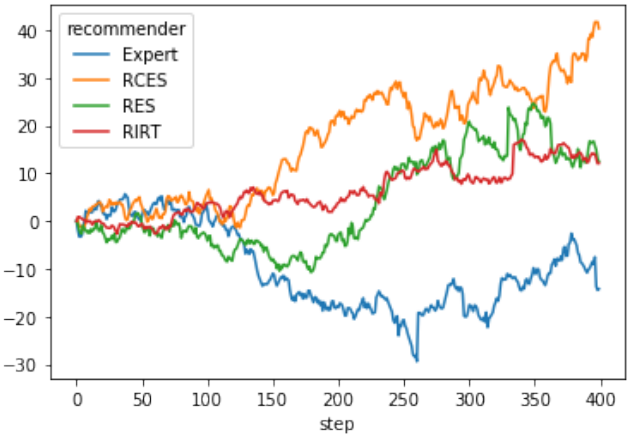

Figure 6: Average score changes by Transformer

shown in Figure 7 and Table 7. RIRT recommends items with lower correctness probability as expected since those questions have the highest entropy, which is computed by $\sum_{r=1}^{c} p_r(1-p_r)$. (Note that Part 3 questions have 3 choices and the other questions have 4 choices.) This fits with the theory that the Fisher information is maximized when probabilities for selecting possible choices are equally distributed. Expert, RES and RCES tends to choose easier questions with correctness probabilities higher than 0.7. In particular, more than half of the questions that RES suggests have expected correctness probability higher than 0.8. This result supports the claim in 3.3 that Algorithm 1 operates adversarially to only maximize the expected score by recommending questions with high expected correctness rate. RCES recommends questions which are considered to be harder than the ones that RES gives.

5 CONCLUSION AND FUTURE WORK

In this paper, we suggested two recommender algorithms RES and RCES that chooses the question with highest expected score and calibrated expected score to capture the learning effect within a test-prep application. These recommender algorithms were then

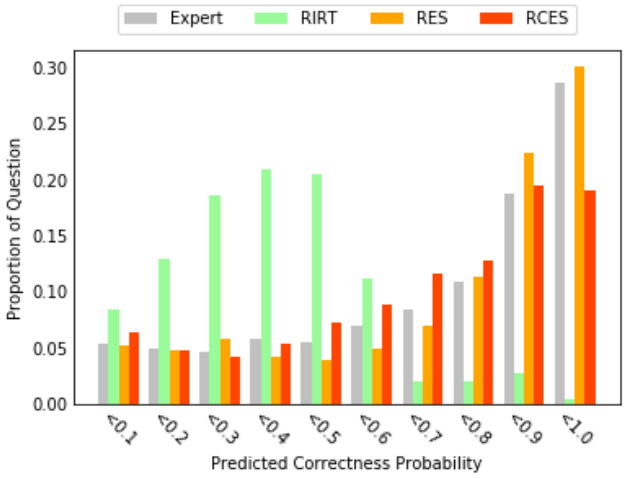

Figure 7: Correctness probability distributions

compared with baseline algorithms such as the expert-annotated random recommender (*Expert*) and the IRT-based recommender (*RIRT*) through an experiment with 1713 active users. The results show that RCES outperforms other recommenders in learning gain.

Also, adversarial effects of RES recommending easy questions was also explored using the predicted correctness probabilities of the recommended questions. Compared to RES, RCES showed a more uniform distribution of correctness probabilities in recommended items, where RES tended to recommend questions with high predicted correctness probabilities.

While this study displays the potential strength of solving question recommendation as a score optimization problem, there are limitations to the current work. Firstly, the current experiment only implemented the recommender for 20% of the questions solved(confided within a single separate section within the application) in the application. While this approach was to not disrupt the original application flow for existing users, a full-scale experiment would have highlighted the results more clearly. It is also notable that the learning effect from question recommendation was not distinctively measured using explicit pre-tests and post-tests. Even though the pre-test and post-test framework does not fit well with the mobile learning environment, it would be beneficial to consider such approaches for a more elaborated result in future works. Secondly, more exploration of the mathematical properties of the recommender algorithm could be beneficial for further improvements. For example, the constant value α, which is the average value of correctness rate for reviewed questions, in RCES may change over time and vary for each users. Capturing such property would greatly benefit the algorithms ability to calibrate the expected score. Also, various properties of the score functions such as monotonicity and submodularity could compute effective bounds for the greedy approach of RES and RCES.

REFERENCES

[1] Solmaz Abdi, Hassan Khosravi, and Shazia Sadiq. 2018. Predicting Student Performance: The Case of Combining Knowledge Tracing and Collaborative Filtering. (2018).

[2] Mary J Allen. 2004. *Assessing academic programs in higher education*. Anker Publishing Company Bolton, MA.

[3] Yoav Bergner, Stefan Droschler, Gerd Kortemeyer, Saif Rayyan, Daniel Seaton, and David E Pritchard. 2012. Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory. *International Educational Data Mining Society* (2012).

[4] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Dongmin Shin, Seewoo Lee, Youngmin Cha, Byungsoo Kim, and Jaewe Heo. 2020. Assessment Modeling: Fundamental Pre-training Tasks for Interactive Educational Systems. *arXiv preprint arXiv:2002.05505* (2020).

[5] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Byungsoo Kim, and Youngjun Jang. 2019. EdNet: A Large-Scale Hierarchical Dataset in Education. arXiv:1912.03072 [cs.CY]

[6] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[7] Neil J Dorans and Linda L Cook. 2016. *Fairness in educational assessment and measurement*. Routledge.

[8] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. 2019. Where's the reward? *International Journal of Artificial Intelligence in Education* 29, 4 (2019), 568–620.

[9] Thomas L Good. 2008. *21st century education: A reference handbook*. Vol. 1. Sage.

[10] Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. 1991. *Fundamentals of item response theory*. Sage.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Alfie Kohn. 2000. Standardized testing and its victims. *Education Week* 20, 4 (2000), 46–47.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[14] Suh Keong Kwon, Moonbok Lee, and Dongkwang Shin. 2017. Educational assessment in the Republic of Korea: Lights and shadows of high-stake exam-based education system. *Assessment in Education: Principles, Policy & Practice* 24, 1 (2017), 60–77.

[15] Kangwook Lee, Jichan Chung, Yeongmin Cha, and Changho Suh. 2016. Machine Learning Approaches for Learning Analytics: Collaborative Filtering Or Regression With Experts?. In *NIPS Workshop, Dec.* 1–11.

[16] Youngnam Lee, Youngduck Choi, Junghyun Cho, Alexander R Fabbri, Hyunbin Loh, Chanyou Hwang, Yongku Lee, Sang-Wook Kim, and Dragomir Radev. 2019. Creating A Neural Pedagogical Agent by Jointly Learning to Review and Assess. *arXiv preprint arXiv:1906.10910* (2019).

[17] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. EKT: Exercise-aware Knowledge Tracing for Student Performance Prediction. *arXiv preprint arXiv:1906.05658* (2019).

[18] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. 2011. Recommender systems in technology enhanced learning. In *Recommender systems handbook*. Springer, 387–415.

[19] Robert L McKinley and Mark D Reckase. 1983. MAXLOG: A computer program for the estimation of the parameters of a multidimensional logistic model. *Behavior Research Methods* 15, 3 (1983), 389–390.

[20] Linda McNeil. 2002. *Contradictions of school reform: Educational costs of standardized testing*. Routledge.

[21] D Monty Neill and Noe J Medina. 1989. Standardized testing: Harmful to educational health. *Phi Delta Kappan* 70, 9 (1989), 688–97.

[22] Shalini Pandey and George Karypis. 2019. A Self Attentive model for Knowledge Tracing. In *EDM*. International Educational Data Mining Society (IEDMS).

[23] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[24] W James Popham. 1999. Why standardized tests don't measure educational quality. *Educational leadership* 56 (1999), 8–16.

[25] W James Popham. 2016. Standardized Tests: Purpose Is the Point. *Educational Leadership* 73, 7 (2016), 44–49.

[26] Joseph Psotka, Leonard Daniel Massey, and Sharon A Mutter. 1988. *Intelligent tutoring systems: Lessons learned*. Psychology Press.

[27] Siddharth Reddy, Sergey Levine, and Anca Dragan. 2017. Accelerating human learning with deep reinforcement learning. In *NIPS'17 Workshop: Teaching Machines, Robots, and Humans*.

[28] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[29] Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1848–1858.

[30] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. 2010. Recommender system for predicting student performance. *Procedia Computer Science* 1, 2 (2010), 2811–2819.

[31] Wim J Van der Linden, Cees AW Glas, et al. 2000. *Computerized adaptive testing: Theory and practice*. Springer.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[33] Runze Wu, Qi Liu, Yuping Liu, Enhong Chen, Yu Su, Zhigang Chen, and Guoping Hu. 2015. Cognitive modelling for predicting examinee performance. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[34] Yiming Yang, Hanxiao Liu, Jaime Carbonell, and Wanli Ma. 2015. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 159–168.

[35] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 765–774.

[36] Yuwen Zhou, Changqin Huang, Qintai Hu, Jia Zhu, and Yong Tang. 2018. Personalized learning full-path recommendation model based on LSTM neural networks. *Information Sciences* 444 (2018), 135–152.