The Impact of Learning Software on Improving Reading Comprehension Skills

Rina Azoulay[1*], Esther David[2], Mireille Avigal[3], Dorit Hutzler[3]

[1] Department of Computer Science, Jerusalem College of Technology, Jerusalem, Israel.
[2] Department of Computer Science, Ashkelon College, Ashkelon, Israel.
[3] The Open University of Israel, Raanana, Israel.

* Corresponding author. Email: rrinaa@gmail.com
Manuscript submitted November 24, 2019; accepted February 4, 2020.
doi: 10.17706/ijeeee.2020.10.3.235-248

Abstract: One of the challenges of an intelligent tutoring system (ITS) is adapting the difficulty level of the questions posed to the student to suit the student's academic level. Our study examines the task of adjusting the system's level of challenges to the level of the learner and addresses the questions of how best to do so and whether there is any benefit from such adjustment. To answer these questions, we developed reading comprehension courseware that includes three adaptive algorithms for adjusting the level of the questions presented to the students: the random selection algorithm, the Q-learning based algorithm, and the Bayesian inference algorithm. We conduct a real-world experiment in which real high school students used the courseware to improve their reading comprehension skills. In order to compare and evaluate the performance of the algorithms, the courseware used by each student utilized one of the three adaptive algorithm alternatives. Our results demonstrate that when considering all of the students, there was significant improvement (learning gain) using each of the methods.

Key words: Educational technology, computer aided instruction, intelligent tutoring systems.

1. Introduction

Intelligent tutoring systems (ITSs) are based on artificial intelligence methods that aim to teach a student specific subjects or improve particular skills [1]. An ITS must be adaptive with respect to the student's capabilities which dynamically change over the tutoring period. To accomplish this, an ITS should have knowledge about the student's capabilities (referred to as the student model) and a set of pedagogical rules. Ma, Adesope and Nesbit [2] define the main functions of an ITS as follows: (1) performs a tutoring function for each student, (2) learns the student model based on the student's responses, and (3) applies the learned student model in order to adapt the tutoring function accordingly. ITSs have been found to improve students' skills in various domains [3]-[5].

Teachers often face a dilemma in knowing how best to engage students and foster their academic achievement without discouraging them by presenting challenges that students perceive as too difficult [1], [3]. The same dilemma exists when designing a courseware capable of adapting itself to the student's level. The research question addressed in our study concerns how to adjust the level of challenges and practices of the system to the level of the learner and whether there is any benefit from such adjustment. Consequently, in this research we aim to develop courseware that presents questions which are suited to the student's academic level, encouraging the student to advance academically and enabling the student to

realize his/her potential without causing frustration or discouragement. Previous studies [4]-[8] have considered this problem and suggested several methods of dealing with it.

In this study, we concentrated on developing courseware aimed at improving the reading comprehension skills of high school students. Our courseware utilizes three adaptation methods, each of which uses a different algorithm to choose the questions to present to the student: the random selection algorithm, the Q-learning based algorithm, and the Bayesian inference algorithm. In order to evaluate this courseware and the adaptation methods employed, we conducted an experiment in which real students utilized our reading comprehension courseware on the Web. Each student manually answered an initial reading comprehension questionnaire. Then he/she was randomly assigned to one of the three adaptation algorithms and worked on the courseware that was adapted to his/her level using the assigned algorithm, and he/she had to answer at least 50 reading comprehension questions on about 10 essays. Finally, when finishing the courseware, the student received a final manual reading comprehension questionnaire. Given the grades of the manual test, the learning gain of each student was evaluated by the difference between the student's grade in the final written questionnaire and his/her grade in the initial written questionnaire.

The results of this experiment show that all of the examined methods yield significant learning gains. The Bayesian inference algorithm outperformed the random method and the Q-learning algorithm in terms of the learning gains, although no significant differences were revealed. We did, however, observe a difference for the weaker students (with grades lower than 60 out of 100 on the initial test) – weaker students using the Bayesian Inference algorithm obtained significantly higher scores than those using the Q-learning algorithm, while none of the weaker students using the random mode complete his process at all.

The rest of the paper is organized as follows. Section 2 describes related work. The basic student model is presented in Section 3, and Section 4 describes the question adaptation methods compared in this study. Section 5 describes our experimental study, Section 6 describes the experimental results, and in Section 7 we discuss our conclusions and provide some directions for future work.

2. Related Work

The idea of creating intelligent systems adapted to students' different levels and needs has fascinated researchers from the early days of AI [9] to the present. The motivation behind these systems is clear. Not all pupils are capable of receiving the support they need in conventional classroom settings. Customized teaching software can help stronger students learn faster, while providing weaker learners assistance in overcoming their difficulties and allowing them to progress at their own pace. Woolf [10] presents the principles of how to build ITSs and evaluate the AI technologies used in these systems. Woolf emphasizes that machine learning techniques can help intelligent tutors acquire new knowledge about students, identify their skills, and use new teaching approaches. She describes practical and theoretical issues addressed by machine learning techniques, including increased tutor flexibility, reduced tutor costs, and adaptation to new student populations. In our study, we developed an ITS to improve the reading comprehension skills of high school students, and we used machine learning techniques to adapt the difficulty of the questions to the current abilities of each student.

When considering computational education systems, it is important to determine whether the integration of machine learning and advanced data-driven methods actually improves the system's ability to help the students. Koedinger *et al*. [11] suggest using the vast amount of data produced by the educational systems in order to advance the technologies of these systems. They believe that data-driven methods can contribute to the development of new ITSs. On the other hand, Baker [12] notes that most of the successful new intelligent systems currently in use do not include intelligent strategies or machine learning methods. He concludes that good intelligent tutors don't have to behave very intelligently, but they should

be designed intelligently,

Arroyo *et al*. [13] describe an intelligent adaptive tutor named Wayang Outpost (now called MathSpring) that models and supports cognitive, affective, and metacognitive (CAM) factors. A variety of features in Wayang Outpost were designed to help improve students' affective experience as they learn with the tutor. For example, affective learning companions trained attributes for success/failure and emphasized the malleability of intelligence and the importance of effort/perseverance. D'Mello and Grasser [14] present AutoTutor, which is an ITS that helps students compose explanations for difficult concepts in Newtonian physics, computer literacy, and critical thinking by interacting with them in natural language with adaptive dialogue which is similar to that of human tutors. Blanchard [15] notes that the learners' cultural background can affect their emotional behavior, motivation, collaboration attitudes, and the way in which the students should be rewarded. He presents a generic modular architecture for designing culturally adaptive e-learning systems and describes a rule based process for selecting culturally appropriate pedagogical resources.

Koedinger *et al*. [16] provide an overview of recent developments in the ITS area. They concentrate on the Cognitive Tutor [17], [18], an algebraic ITS produced by Carnegie Learning for high school mathematics which was used two days a week by around 600,000 students a year in 2,600 middle or high schools. Ritter *et al*. [17] conducted a year-long evaluation study showing that students who used Cognitive Tutor achieved higher gain of learning w.r.t. those taking traditional algebra courses.

Koedinger and Aleven [18] provide a brief review of cognitive tutorials and compare different types of tutorials. They consider the "assistance dilemma," which refers to the issue of what information should be provided to students so they can optimally learn. In order to address this issue as it pertains to the Cognitive Tutor, additional information about the problem solution is initially withheld, and information is interactively added only as needed, through yes/no feedback, explanatory hints, and dynamic problem selection. However, Koedinger and Aleven claim that additional research is required to characterize the qualitative conditions and quantitative threshold parameters that can aid instructional designers and instructors in making good decisions regarding the assistance dilemma.

Van der Kleij *et al*. [19] evaluate the effect of feedback provided by the ITS following each student response. Their meta-analysis includes 40 studies, published between 1968 and 2012, and compares the influence of various types of feedback on each experiment's effect size. They found that more elaborate feedback led to greater improvement in learning outcomes than simple feedback. Larger effect sizes were found for mathematics compared to social sciences, science, and languages, and effect sizes were negatively affected by delayed feedback timing and the type of school, i.e., primary or high school. In our experiment, the feedback included presenting the correct answers and highlighting their location in the text, both of which were aimed at helping the student better understand how the correct answer could be inferred.

Several studies have been performed to evaluate the benefits of using personalized and adaptive computerized learning systems [2], [12]. Kulik and Fletcher [20] provide a meta-analysis review, considering 50 controlled evaluations of intelligent computer tutoring systems, and their findings showed that students who received intelligent tutoring outperformed students studying in conventional classes in 46 (or 92%) of the controlled evaluations, with statistically significant findings in 39 (or 78%) of the studies. Their evaluations reveal that ITSs typically increase the students' performance well beyond the level of conventional classes and even beyond the level achieved by students who receive instruction from other forms of computer tutoring or human tutors.

The research question addressed in our study concerns how to adjust the level of challenges and practices of the system to the level of the learner and whether there is any benefit from such adjustment. Lomas *et al*. [21] examine the effect of the difficulty level of the challenges presented to players during a

math game on the players' motivation and their learning process. To test the Inverted U Hypothesis, which predicts that maximum game engagement will occur with moderate challenge, they performed two large-scale online experiments, finding that in almost all cases the subjects were more engaged and played longer when the game was easier. On the other hand, they also found that the most engaging design conditions produced the slowest rates of learning. Their results demonstrate the importance of adaptive learning in maintaining the two criteria that must be met by the learning system: ensuring that the tasks are not too difficult, so as to provide the students a successful experience and increase their motivation, while making sure that the tasks are not too easy, so that actual improvement in the students' level can be achieved. The results obtained in our study are different than the results obtained by Moeyaert *et al*. [22], in which no significant effect was found when different exercise levels were used. Moeyaert *et al*. examine six different item selection algorithms in the learning environment. The algorithms differed in their level of difficulty, and the last algorithm chose items randomly. According to their study, the degree of difficulty had no significant effect on either the learning outcome or motivation. Note, however, that the students in the experiment of Moeyaert *et al*. were all adults who may have been less sensitive to the emotional effects of failure during the study.

Murray and Perez [23] examine completion rates and exercise scores for students assigned adaptive exercises and compare them to completion rates and quiz scores for students assigned objective type quizzes in a university digital literacy course. In their study, no significant difference in the test scores of students using adaptive learning and students learning in a setting of traditional instruction was seen. Similar to the work of [22] and [23] described above, in our study no significant difference in learning gain was seen among students using adaptive algorithms and those using a random algorithm. However, further analysis of the results of our study, showed that the ability to adapt the questions to the student's reading comprehension level was important, and even critical, for those that were initially considered weaker learners, as described in Section 6.

Pavlik *et al.* [24] study the effect of efficient practice scheduling on the learners' results. They present a practice scheduling algorithm based on ACT-R (Adaptive Control of Thought – Rational). ACT-R's equations theoretically model the effect of practice history on both success and latency of later performance, and the practice scheduling algorithm uses these equations to choose the next item to learn in order to optimize long-term learning efficiency. They tested the scheduling algorithm in a university course in China. Their results show that the students preferred the adaptive ACT-R policy over the random selection policy and that the students using the adaptive policy had greater learning achievements.

The abovementioned issue of whether an adaptive tutoring system improves learning results is strongly related to the question of how to assess the student. Pelánek [25] considers the issue of how to assess the student's skills during the ITS session, based on the student's performance. He compared time decay functions and the Elo rating system, and found that these two approaches provide good and consistent improvement in the student's skills.

Further to the above studies, the question discussed in our research concerns the question of which adaptive method to use in order to adapt the ITS challenges and tasks to the student's abilities.

3. The Level Selection Process

In our study, we considered the level selection problem in a courseware where students can practice the courseware in order to improve their reading comprehension skills. We assume the following student model. Each student has a distribution reflecting his/her level at each particular time. The distribution of the student's level is Gaussian, with a given mean and standard deviation. At a given time, the student's current level can be any continuous level, derived from the student's level distribution. Note that each

student has his/her own Gaussian distribution for his/her level at each given time, which is unknown to our courseware. The courseware can only learn the students' abilities by observing their answers to previous questions.

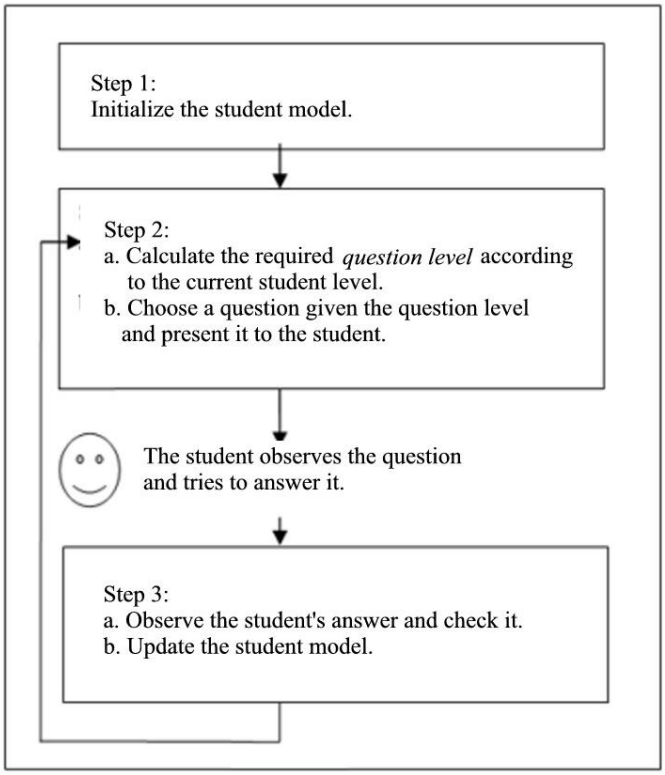

Fig. 1. The main steps of the level selection process.

When working with the courseware, each student obtains a set of N questions. The simplified level selection process is presented in Figure 1 and is based on three steps: 1) initialization of the student model, 2) the process of choosing the next appropriate question, 3) examining the student's answer and saving the information for the rest of the learning session. Note that Figure 1 also describes the CAT (Computer Adaptive Testing) process, since it enables the successful selection of questions for the purpose of maximizing the precision of the exam based on what is known about the student from previous questions. However, the maximization criteria when choosing the level of the next question will be different in an ITS than in a CAT model. In a CAT model, the aim is to maximize the knowledge about the student, while in an ITS, the goal is to efficiently train the student, while maintaining an adequate success rate. In addition, in a CAT model, the test terminates when the student's standard error of measurement falls below a certain value, whereas in an ITS, learners can continue to learn, until they obtain the required skills or the learning process has been completed.

3.1. The Internal Utility Function

The aim of the courseware is to present questions that the student is capable of answering correctly, but the questions should also be at the highest level for the student, so as to utilize and sharpen the learner's skills.

In our courseware, we define the internal utility function presented below, which sums the level of the questions that were correctly answered by the student, where the higher the level of questions answered correctly, the higher the total evaluation.

Utility function:

$$Utility = \sum_{question_i=1}^{N} f_1(question_i) \quad (1)$$

where

$$f_1(\text{question}_\text{i}) = \begin{cases} \text{Level}(\text{question}_\text{i}) & \text{question}_\text{i} \text{ was answered correctly} \\ 0 & \text{question}_\text{i} \text{ was answered incorrectly} \end{cases}$$

Implemented in the courseware, the evaluation function considers the student's results for each question.

It is worth noting that the student is unaware of the internal utility function value, which is an internal value used by the courseware. The measurement the users are aware of is based on their percentage of success, where the score for an incorrect answer is zero and a correct answer is 100; this is reduced if a hint was required or if it took a relatively long time for the student to answer the question.

4. The Adaptive Algorithms Compared in Our Study

In this section, we provide a detailed description of the Q-learning and Bayesian inference algorithms used for the task of adapting the level of the courseware questions. Note that the algorithms are used in order to determine the level of the next question given to the student: in turn, the courseware will present a question at that level to the student.

4.1. Q-Learning

The Q-learning algorithm is a reinforcement learning (RL) algorithm. RL algorithms consider situations, in which there is a set of states, S, and a set of actions, A, which an agent can take. At each time step, the reinforcement learning agent chooses an action $a \in A$ from the set of actions available. The environment moves to a new state $s \in S$, and the reward associated with the transition is determined. The goal of the agent is to collect as many rewards as possible, and the main dilemma of the agent, called "the exploration-exploration dilemma," is whether to choose the currently best known action or to try using other actions in order to gather more information that might lead to greater rewards in the future.

The Q-learning algorithm [26], [27] saves a value Q for each pair $<s,a>$, where s is the current state, and a is the particular action taken by the agent given the current state. Given the Q values with a probability of ε, the algorithm explores and randomly chooses an action, and with a probability of 1-ε, the algorithm exploits and chooses the action with the highest Q value. In our framework, a denotes the question level. After the action is taken (i.e., the question has been asked), it is checked whether the question was answered correctly. Then, the courseware internal utility function is calculated given the level of the question and whether it was answered correctly). Then, the Q value of action a is updated using formula (2).

$$\text{x } Q(a) \leftarrow Q(a) + \alpha(r + \gamma \cdot max_{a'} Q\,(a') - Q(a)\,) \quad (2)$$

where α defines the learning rate, i.e., the speed of convergence of the Q values; α=0 results in the agent not learning anything, while α=1 results in the agent considering only the most recent information. Finally, r is the reward value, and γ defines the discount factor, which determines the importance of future rewards, where γ=0 will cause the agent to consider just the current rewards.

In the courseware used in our study, only one state exists, and the actions are the possible question levels. Each question level is associated with a certain Q value. Once the student provides an answer, the relevant Q value is updated according to the reward, which indicates the success or failure in answering the question. Fig. 2 presents the Q-learning algorithm process.

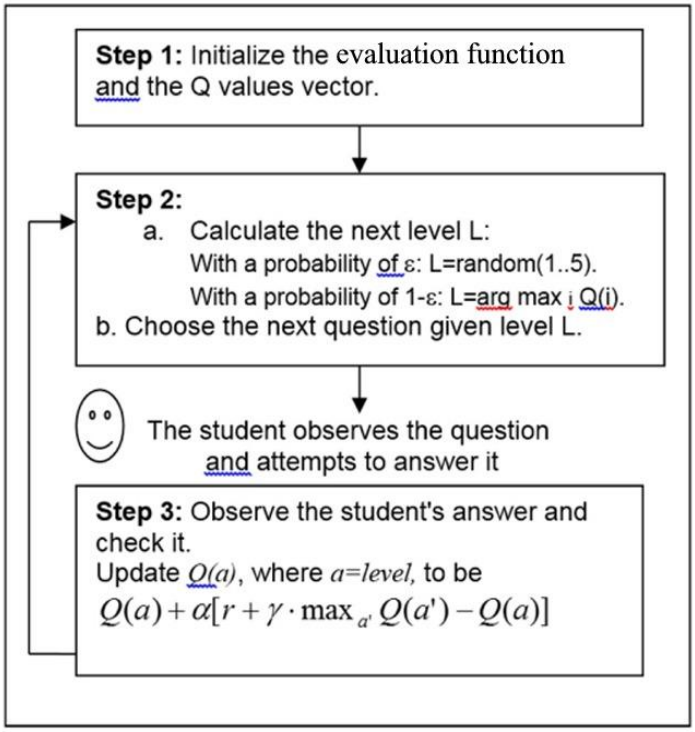

Fig. 2. The Q-learning algorithm process.

4.2. Bayesian Inference Algorithm

A number of assumptions are made when using the Bayesian inference method [12] algorithm [12]: 1) We assume that at each particular time, the actual level of the student depends on different factors; 2) The student's level has a normal distribution with a mean and a variance, and the ability of the student to successfully answer a particular question depends on his/her actual level at that particular time. 3) Each student is defined by a mean and variance for his/her level distribution. 4) Both the student's level and the standard deviation of the student's level can be integers between zero and five.

Given a particular student, the algorithm associates a constant probability with each set of parameters (μ and σ) representing the student's level distribution. In each step, the algorithm considers all possible combinations of parameters for the student, and for each candidate question's level, the algorithm calculates the expected internal utility function value of the student receiving a question at this level, given all possible distributions of students. Then, the algorithm chooses the level with the highest expected utility, using formula (3) below.

$$ChosenLevel=argmax_{level=1..5}\ \Sigma_{\mu,\sigma}\ prob(_{\mu,\sigma})* \\ (pwins(level\mid \mu,\sigma)*util\ (level)+(1-pwins(level\mid \mu,\sigma))*UtilFailure) \quad (3)$$

where *pwins(level | μ,σ)* is the probability of a question *n* from level *level* to be chosen, *util (level)* is the utility of a successful answer to a question from this level, and UtilFailure is the negative utility (penalty) from failure to answer a question from this level. Once a question has been chosen, and the student's response has been observed, the probability of each distribution of the student is updated using the Bayesian rule. Further details on the Bayesian iInference algorithm are provided in [28]. The Bayesian inference process is described in Fig. 3.

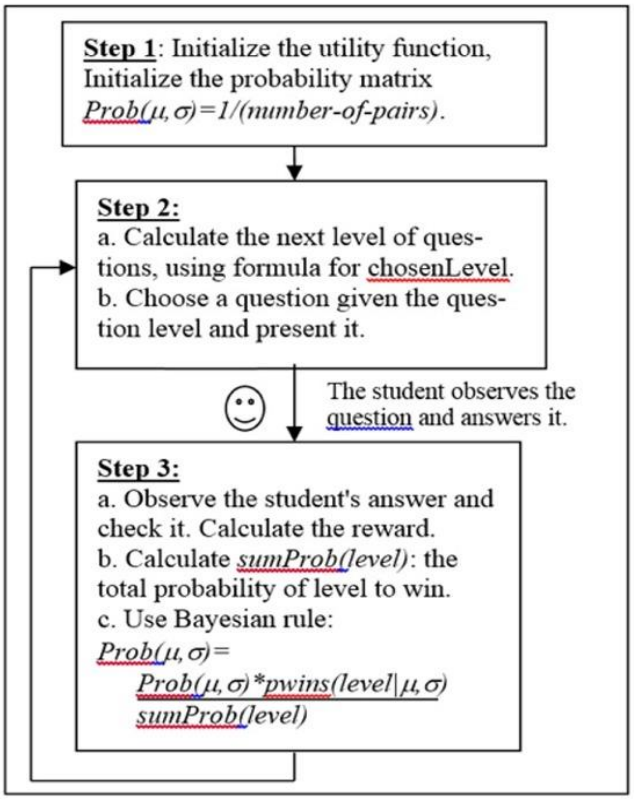

Fig. 3. The Bayesian inference algorithm process.

Next, in Section 5, we describe the results of our experimental study, using the adaptation algorithm described above.

5. Experimental Setup

In order to examine the efficacy of the proposed courseware and the three adaptation methods it employed, a real-world study was conducted, in which a group of high school students used our courseware in order to improve their reading comprehension skills. Using this courseware, we compared the Bayesian inference algorithm, the Q-learning algorithm, and a random method in which the questions for each essay were arbitrarily chosen without any adaptation to the student's level. In this section, we describe the courseware used by the students and its life cycle and provide a detailed description of the study and the results of the experiment.

The courseware used in this study is comprised of three modules: a domain module, a pedagogical module, and a student module.

The domain module contains a bank of 18 essays and 277 multiple choice questions concerning the essays. Each question consists of the question itself, four possible answers, the correct answer, the difficulty level of the question (graded on a scale from one to five), a hint (providing some direction regarding where and how to find the answer to the question), and the location of the answer in the essay. The difficulty level of each question was determined by a panel of experts using a set of criteria based on a taxonomy customized for the reading comprehension domain. The detailed decision criteria used for the questions' difficulty classification are described in [29].

The pedagogical module determined the algorithm used to choose the courseware questions. Upon logging into the system, each student was assigned to one of the three adaptation methods described above. The student module maintained the data about the student, saving the current level of the student. The way

this level was saved depended on the algorithm used in the pedagogical module.

After the initial algorithm assignment stage, done in the pedagogical module. the courseware was implemented using the following five-step process:

1) Essay selection – An essay was chosen from the bank stored in the domain module.
2) Question selection – A question was selected according to the student's current level and the adaptive algorithm he/she is assigned to.
3) Response to the student's answer – Upon receiving the student's answer, the system recorded the response time and correctness of the answer, as well as whether the student requested a hint. An incorrect answer resulted in a score of zero. A correct answer without a delay or hint resulted in a score of 100 for the question. If a hint was given, the grade was reduced by 20%. In addition, if the answer was provided after more than five minutes, the grade was reduced by 20%.
4) Level adjustment – After completing each set of five questions, the student's level was adjusted in accordance with the relevant algorithm.
5) If the required total number of questions was reached, the process terminated. If five questions were already provided to the student for the essay, the process continued with step one, and a new essay was presented to the student; if not, the process continued with step two, and the student was presented with another question for the essay.

The subjects' (namely a group of high school students) participation in the experiment took place in the three stages described below.

Stage I: In the initial stage, the student received a reading comprehension questionnaire that included an essay and questions pertaining to the text. This was done in order to establish a reference point for each student's level. The test was manually checked by a reading comprehension teacher, and these test results did not influence the courseware.

Stage II: The student practiced using the courseware (with one of the three adaptation algorithms) in his/her spare time during 2-3 weeks. The student was not aware of the adaptation method utilized by the courseware.

Stage III: In the final stage, each student received a new essay (not seen in the previous stages) and questions about it and had to answer the questions within a set period of time, thereby determining the student's post-training level.

The students participating in the experiment received a payment of 100 NIS (28$) for meeting the requirements and fully participating in the experiment (completing the initial questionnaire, using the ITS courseware for to answer at least 50 questions, and completing the final questionnaire). The length of the learning session varied for different students, ranging from 2-3 days to 2-3 weeks. After completing the task with the courseware learning session, i.e., answering at least 50 questions, the learners received the final questionnaire and answered it. Students could complete the post-test any time, from one day to two weeks after finishing the courseware part of the experiment.

Given the students' scores, in order to examine the success of each student and the the degree of improvement attained during training, we calculated the students' learning gain as the average difference between the students' grades on the questionnaire taken of Stage I and those obtained in Stage III for each type of algorithm (the random method, Q-learning algorithm, and Bayesian inference method).

6. Experimental Results

During the experiment, a total of 151 students completed the initial questionnaire and started the reading comprehension courseware. Each student who performed the initial questionnaire was assigned (without his/her knowledge) to one of three algorithms, which determined how the courseware operated

for that student. A third of the students were assigned to the randomized algorithm, a third to a Q-learning algorithm for selecting the levels of questions, and a third to a Bayesian inference based algorithm.

In Table 1, we present a summary of the starting point of the total set of students who used the courseware, and the starting point of those students who fulfilled all of the experiment's requirements (completed the initial questionnaire, answered at least 50 questions presented by the courseware, and completed the final questionnaire). As depicted in Table 2, the initial scores of the students using the three different methods were very similar, however less than 50% of the students actually proceeded to stage II where they worked on the courseware and answered 50 questions). We believe that the reasons for this were that performing the experiment was time consuming and students were able to stop using it at any time. Furthermore, the payment the students received was not enough of an incentive given the amount of time and effort they were required to invest, namely: 90 minutes*2 for completing the initial and final questionnaires, 10*20 minutes + 50*5 minutes to answer 50 questions from 10 essays. Consequently, the students were required to invest over nine hours of their time to perform all of the requirements, and only received payment equivalent to $3 per hour for their efforts. Nevertheless, when considering the students using the random selection algorithm, all the weak students (students with low grade in the initial questionnaire) did not fulfilled their requirements, and only relatively stronger students were more likely to be able to successfully complete their tasks (answer the questions presented by the courseware). As a result, we can observe in Table 1, when considering the students that fulfilled their requirements, the initial average score of the students assigned to the random algorithm was the highest.

A possible explanation for this can be the fact that when using the random method, the questions assigned to the students could have been variously too hard or too easy for the student. Consequently, while this caused the motivation of the weaker students decreases.

Table 1. Summary of the Experiment

	Random	**Q-learning**	**Bayesian Inference**
Total number of students	55	49	50
Number of students who fulfilled all of the requirements	20	18	22
Average score on the initial questionaire for all students	72.2	71.8	70.6
Average score on the initial questionnaire for students who fulfilled all of the requirements	78.5	69.7	69.5

The satisfaction of the students was reflected by their verbal comments at the end of the study. Most of the students did not leave any verbal comments at all. One student assigned the random algorithm wrote, "It was hard and annoying, and the answers were confusing." Students using the Q-learning algorithm wrote, "There are too many questions," "This stage is too long," "You should improve the design," and "reasonably precise " and students using the Bayesian inference algorithm wrote, "The questions are somewhat fair," "Improves reading comprehension ability," "The articles are interesting; the aesthetic aspect can be improved," and "The system is very convenient for learning!" Based on the feedback we received, it appears that the students who were assigned the Bayesian inference algorithm enjoyed a good learning experience which strengthened their results.

Next, we compared the learning gain (the difference in the scores on the written questionnaires before and after using the courseware) for the students who answered at least 50 questions. Note that the reason

the data of the students who answered less than 50 questions were excluded is the fact that the data obtained from those students would not adequately reflect the progress and improvement obtained during training.

Our results indicate that the students assigned to the random algorithm demonstrated the least amount of improvement in their reading comprehension in comparison to the other groups. The students assigned the Q-learning algorithm demonstrated greater improvement, and those assigned the Bayesian inference algorithm demonstrated the greatest degree of improvement in their reading comprehension grades, and the greatest increase in the final mean score, as indicated in Table 2.

Using the courseware using in any adaptation algorithm, significantly increased the mean scores obtained across all methods. When comparing the improvement of the students assigned to the three adaptation methods described, the learning gain of those assigned to the random method was not significant (t=1.47, p=0.157), and the improvement of the students assigned to both the Q-learning method and the Bayesian inference method was significant. For the Q-learning method, t=3.43 and p=0.003, and for the Bayesian inference method, t=2.43 and p=0.24. However, despite of the fact that the learning gains were significant, the difference of the learning gains of the different algorithms was, indeed, not significant. : running ANOVA on the learning gain of the three groups resulted in an F score of 1.21 and p=0.306.

However, when looking only at the weaker students (those who received a score lower than 60 on the initial questionnaire), only nine of the 26 students completed both the initial and final questionnaires and answered 50 questions or more, and none of them were associated with the random algorithm. The learning gain of those using the Bayesian inference algorithm was significantly greater than the learning gain of those using the Q learning algorithm (t=3.2, p=0.015). This demonstrates that the courseware adaptation, and more accurate adaptation, are more important for weaker students than it is for the other students.

Table 2. Details on the Students Who Fulfilled All the Requirements

	Random	Q-learning	Bayesian inference
Number of students who fulfilled all of the requirements	20	18	22
Average score on the initial questionaire	78.5	69.7	69.45
Average score on the final questionaire	82.5	79.2	81.54
Average learning gain (score on the final questionaire minus the score on the initial questionaire)	4	9.4	12.1
Standard deviation of learning gain	12.15	11.68	22.1

To summarize, we can conclude that the use of computerized courseware improves the reading comprehension skills of the students. In addition, in our experiments we found that the adaptation method employed was more important for students whose initial academic ability was weaker and that these students had a significantly greater learning gain when using the Bayesian inference algorithm.

7. Conclusions

At the core of the pedagogical model of the ITS is choosing the difficulty level of challenges presented to the student. The research question addressed in our study concerns how to adjust the level of challenges

and practices of the system to the level of the learner and whether there is any benefit from such adjustment. In order to answer this question, we developed courseware aimed at improving the reading comprehension skills of high school students. Our courseware utilizes three adaptation algorithms to choose the questions to present to the student: the random selection method, the Q-learning based method, and the Bayesian inference algorithm. We conducted a real-world experiment in which high school students were trained using the reading comprehension courseware we developed. The courseware utilizes three different adaptation methods: the random selection method, the Q-learning based method and the Bayesian inference method. Our results demonstrate that when considering all the students, there was significant improvement (learning gain) using each of the methods. While the students using the Bayesian inference method had the greatest learning gain, the difference in the improvement using the different methods was not found to be significant. However, students with low grades on the initial questionnaire (a grade lower than 60) showed a significantly greater learning gain when using the Bayesian inference method than when using the Q-learning method. These results confirm that a courseware that adapts the questions' difficulty level to the user's capabilities can improve a student's skills and show that adjusting the questions' difficulty level to the student's level is more crucial for students starting at a lower academic level.

Future research is needed to compare different adaptation methods on larger groups of students and for longer periods of time, in order to strengthen the conclusions reached. Additional experiments are also needed in order to evaluate other capabilities of ITSs, including teaching of new topics, were the ITS should decide when to move to the next topic. In addition, in future work we intend to exploit the abilities of the adaptation algorithms described in this study to train and improve various learning skills of children with special needs.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Rina Azoulay designs algorithm and analyzes results; Esther David supervises, edits paper, and does related work; Mireille Avigal supervises and designs experiments; Dorit Hutzler conducts the research and analyses the data. All authors had approved the final version.

References

[1] Schmidt, R. A., & Bjork, R. A. (1992). New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. *Psychological Science*, *3(4)*, 207-218.

[2] Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, *106(4)*, 901-918.

[3] Scager, K., Akkerman, S., Pilot, A., & Wubbels, T. (2017). Teacher dilemmas in challenging students in higher education. *Teaching in Higher Education*, *22(3)*, 318-335.

[4] Beck, Woolf, B. P., & Beal, C. R. ADVISOR: A machine learning architecture for intelligent tutor construction. *Proceedings of AAAI 2000* (pp. 552-557).

[5] Martin, K. N., & Arroyo, I. (2004). AgentX: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. *Proceedings of Intelligent Tutoring Systems: 7th International Conference* (pp. 564-572). Springer Berlin Heidelberg.

[6] Vriend, N. J. (1997). Will reasoning improve learning? *Economics Letters*, *55(1)*, 9-18.

[7] Gittins, J. (2011). *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, Ltd.

[8] L2TOR, second language tutoring using social robots. (2017). Retrieved from the website:

http://www.l2tor.eu/researchers-professionals/

[9] Carbonell, J. R. (1970). AI in CAI: An AI approach to CAI. *IEEE Transactions on Man-Machine Systems*, *11*, 190.

[10] Woolf, B. P. (2007). *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing e-Learning*. Morgan Kaufmann Publishers Inc.

[11] Koedinger, K. R., Brunskill, E., Baker, R. S. J. D., McLaughlin, E. A., & Stamper, J. (2013). New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, *34*, 27-41.

[12] Ryan, S. B. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, *26(2)*, 600–614.

[13] Arroyo, I., Woolf, B. P., Burelson, W., Muldner, K., Rai, D., & Tai, M. (2014). A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *Int. Journal of Artificial Intelligence in Education*, *24(4)*, 387–426.

[14] D'Mello, S. K., & Graesser, A. C. (2012). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*, *2(4)*, 1–38.

[15] Blanchard, E. G. (2009). Adaptation-oriented culturally-aware tutoring systems: When adaptive instructional technologies meet intercultural education. *Handbook of Research on Human Performance and Instructional Technology*. Hershey: PA, Information Science Reference.

[16] Koedinger, K. R., Brunskill, E., Baker, R. S. J. D., McLaughlin, E. A., & Stamper, J. (2013). New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, *34*, 27-41.

[17] Ritter, S., Anderson, K., Koedinger, K., & Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, *14(2)*, 249-255.

[18] Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, *19(3)*, 239-264.

[19] Fabienne, M., *et al*. (2015). Effects of feedback in a computer-based learning environment on students' learning outcomes: A meta-analysis. *Review of Educational Research*, *85(4)*.

[20] Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems. *Review of Educational Research*, *86(1)*, 42-78.

[21] Lomas, D., Patel, K., Forlizzi, J. L., & Koedinger, K. R. (2013). Optimizing challenge in an educational game using large-scale design experiments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 89-98).

[22] Moeyaert, M., Wauters, K., Desmet, P., & Noortgate, W. (2016). When easy becomes boring and difficult becomes frustrating: Disentangling the effects of item difficulty level and person proficiency on learning and motivation, systems.

[23] Murray, M. C., & Perez, J. (2015). Informing and performing: A study comparing adaptive learning to traditional learning. *Informing Science*, *18*, 111-125.

[24] Pavlik, P., Bolster, T., Wu, S. M., Koedinger, K., & Macwhinney, B. (2008). Using optimally selected drill practice to train basic facts. *Proceedings of Int. Conf. on Int. Tutoring Systems* (pp. 593-602).

[25] Pelánek, R. (2014). Application of time decay functions and Elo system in student modeling. *Educational Data Mining,* 21-27.

[26] Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

[27] Michael, O. D. (1995). *Q-Learning for Bandit Problems*.

[28] Azoulay, R., David, E., Hutzler, E., & Avigal, M. (2014). Adaptation schemes for question's level to be proposed by intelligent tutoring systems. *Proceedings of ICAART 2014: 6th International Conference on Agents and Artificial Intelligence* (pp. 245-255).

[29] Hutzler, D., David, E., Avigal, M., & Azoulay, R. (2014). Learning methods for rating the difficulty of reading comprehension questions. *IEEE Int. Conf. on Software Science, Technology and Engineering*.

Rina Azoulay got the Ph.D. in computer science from Bar Ilan University in 2001. She is a senior lecturer in the Computer Science Department at the Jerusalem College of Technology, Israel.

Esther David got the Ph.D. in computer science from Bar-Ilan University in 2004. She is a senior lecturer in the Computer Science Department at the Ashkelon Academic College, Israel. She completed her post doc under Nicholas Jennings.

Mireille Avigal got the D.Sc. in mathematics from the Technion, Israel Institute of Technology in 1992. She joined the Computer Science Department of the Open University of Israel in 1989, where since then she is a course coordinator, takes part in developing courses in mathematics and computer science and a researcher. Her research interests include machine learning, evolutionary computation and multimedia analysis.

Dorit Hutzler got the M.Sc. in computer science from the Open University in 2015. She is a lecturer in the Computer Science Department and in the Mathematics Department at the Jerusalem College of Technology, Israel.