Learning by Teaching SimStudent – An Initial Classroom Baseline Study Comparing with Cognitive Tutor

Noboru Matsuda[1], Evelyn Yarzebinski[2], Victoria Keiser[1], Rohan Raizada[1], Gabriel J. Stylianides[3], William W. Cohen[1], and Kenneth R. Koedinger[1]

[1] School of Computer Science, Carnegie Mellon University
5000 Forbes St. Pittsburgh PA 15213 USA
[2] University of Pittsburgh
[3] Department of Education, University of Oxford

Abstract. This paper describes an application of a machine-learning agent, *SimStudent*, as a teachable peer learner that allows a student to learn by teaching. SimStudent has been integrated into APLUS (Artificial Peer Learning environment Using SimStudent), an on-line game-like learning environment. The first classroom study was conducted in local public high schools to test the effectiveness of APLUS for learning linear algebra equations. In the study, learning by teaching (i.e., APLUS) was compared with learning by tutored-problem solving (i.e., Cognitive Tutor). The results show that the prior knowledge has a strong influence on tutor learning – for students with insufficient training on the target problems, learning by teaching may have limited benefits compared to learning by tutored problem solving. It was also found that students often use inappropriate problems to tutor SimStudent that did not effectively facilitate the tutor learning.

Keywords: Learning by teaching, teachable agent, SimStudent, machine learning, inductive logic programming.

1 Introduction

The goal of our current project is to investigate cognitive and social theories of the *effect of tutor learning* [1]. Although it is well known that students learn when they teach others, little is known about the underlying cognitive principles. Part of the difficulties of studying the effect of tutor learning is its cost and human factors. For example, to conduct an empirical study on learning by teaching in an authentic classroom setting, students must switch their role (tutor vs. tutee). To overcome this challenge, we developed a synthetic pedagogical agent (called *SimStudent*) that acts as a peer learner [2]. We then developed an on-line game-like learning environment (called APLUS) where students learn algebra equations by teaching SimStudent.

The aim of this paper is to first introduce SimStudent and APLUS. The paper then describes a classroom study, in which the effectiveness of learning by teaching SimStudent was evaluated by comparing APLUS/SimStudent with Cognitive Tutor.

G. Biswas et al. (Eds.): AIED 2011, LNAI 6738, pp. 213–221, 2011.

2 SimStudent and the APLUS Learning Environment

2.1 SimStudent: A Synthetic Peer Learner

SimStudent learns procedural skills from examples. In the current context, individual students *interactively* tutor SimStudent. Namely, the examples are given as a combination of feedback and hint in the context of tutored-problem solving.

When tutoring SimStudent, the student poses a problem for SimStudent to solve. SimStudent then attempts to perform one step at a time and asks the student about its correctness. If the student provides negative feedback, SimStudent attempts an alternate action. If SimStudent cannot perform a step "correctly," it asks the student for a hint. The student then *demonstrates* the step as a hint. SimStudent inductively generalizes the examples using *background knowledge*, and generates a set of production rules that represent learned skills.

One of the unique characteristics of SimStudent as a teachable agent is its ability to *model human learning*. We are particularly interested in modeling errors that human students make from inappropriate inductions [3]. We hypothesize that students make such errors when they rely on shallow problem solving features instead of domain principles. One example is to identify '3' in '3x' as a *number* instead of a *coefficient*, as when students "divide both sides by 3 for $3x = 6$." A student who perceives such a shallow feature would likely divide both sides of $3/x = 6$ by 3 as well, which is one of the most frequently observed student errors. To model this type of learning, we modified SimStudent's background knowledge by dropping the concept of coefficient and adding more perceptually grounded background knowledge (e.g., "get a number before a variable"). This particular functionality provides us with the opportunity to investigate the impact of differences in the tutee's competency during tutor learning.

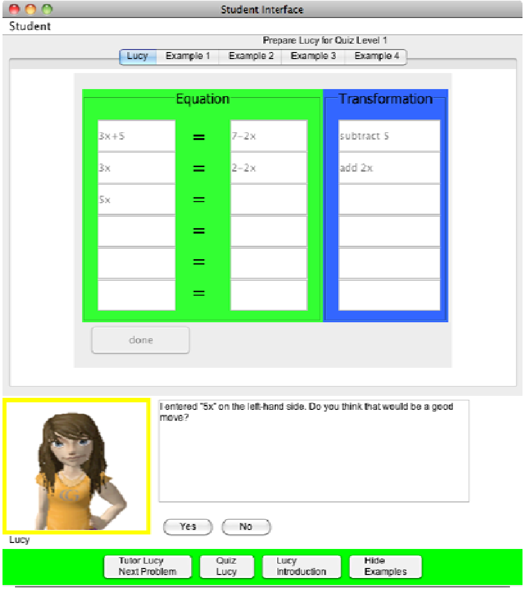
2.2 APLUS: An On-line Learning by Teaching Environment

SimStudent is embedded into an online, game-like learning environment, called APLUS (Artificial Peer Learning environment Using SimStudent). Fig. 1 shows a screenshot of APLUS. SimStudent is visualized at the lower left corner and, in this example, is named Lucy. There is a *Tutoring Interface* taken from a Cognitive Tutor that allows the student and SimStudent to collaboratively solve problems. In the figure, SimStudent entered "5x" and is asking the student if it is correct or not. The student responds by clicking on the [Yes/No] button. Because the student is also learning how to solve equations, he/she may get stuck. In such a situation, students are encouraged to review examples provided in the [Example *n*] tab shown on top of the screen.

The student is told that his/her goal is to have Lucy pass the quiz. When the student clicks on the [Quiz Lucy] button, Lucy takes a quiz without feedback from the student. The summary of the quiz results appears in a separate window (see Fig. 2).

3 Related Works on Teachable Agent

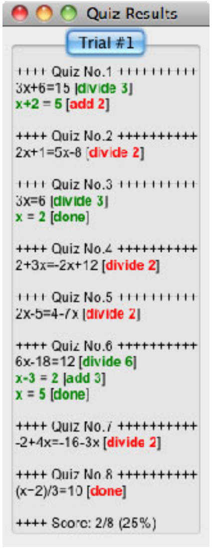
Using a pedagogical agent as a peer learner to study the effect of tutor learning is not a new idea [4-8]. Such a pedagogical agent is often called a *teachable agent*.



3x+5	=	7−2x	subtract 5
3x	=	2−2x	add 2x
5x	=		
	=		
	=		
	=		

I entered "5x" on the left-hand side. Do you think that would be a good move?

Fig. 1. A screenshot of APLUS – Artificial Peer Learning environment Using SimStudent. There are four examples available in the [Example] tabs to review. The [Quiz Lucy] button is to initiate a quiz. In this figure, Lucy just entered “5*x*” on the left-hand side and asked the student for feedback on the correctness.



```
++++ Quiz No.1 ++++++++++
3x+6=15 [divide 3]
x+2 = 5 [add 2]

++++ Quiz No.2 ++++++++++
2x+1=5x-8 [divide 2]

++++ Quiz No.3 ++++++++++
3x=6 [divide 3]
x = 2 [done]

++++ Quiz No.4 ++++++++++
2+3x=-2x+12 [divide 2]

++++ Quiz No.5 ++++++++++
2x-5=4-7x [divide 2]

++++ Quiz No.6 ++++++++++
6x-18=12 [divide 6]
x-3 = 2 [add 3]
x = 5 [done]

++++ Quiz No.7 ++++++++++
-2+4x=-16-3x [divide 2]

++++ Quiz No.8 ++++++++++
(x-2)/3=10 [done]

++++ Score: 2/8 (25%)
```

Fig. 2. A summary of the quiz. SimStudent is pre-trained with inappropriate background knowledge hence may make some common human errors such as adding 2 to *x*+2=5.

Some teachable agents employ a machine-learning algorithm to carry out genuine learning [5], whereas other systems do not require sophisticated machine-learning modules [4, 6, 7]. For some agents, the knowledge is directly transferred from the student to the teachable agent using a shared knowledge representation [4, 6, 7]. Conversely, other agents rely on an indirect knowledge transfer, meaning the agent’s knowledge is not visible to the student [5]. Some teachable agents are capable of interacting with the student in a similar way as humans do – learning by tutored-problem solving [5, 8]. Other teachable agents “learn” knowledge in a different mode than students do, which somewhat limits the tutoring interactions [4].

SimStudent deploys inductive logic programming to interactively learn problem-solving skills. Because of this genuine learning mechanism, SimStudent may learn incorrect rules much like human students do (details follow in the next section) when its background knowledge is controlled as described in Section 2.1. Since the skills learned by SimStudent are not directly visible, the student must reason about the competency of SimStudent from its behavior. These characteristics provide more ecological validity for our purposes of studying theories of learning by teaching, bacause the students are able to teach SimStudent interactively in the same way as they teach their friends.

4 Evaluation Study

4.1 Methods

Two high schools in a rural area near Pittsburgh, PA, participated in the study under the supervision of Pittsburgh Science of Learning Center (www.learnlab.org). In both schools, the Algebra I Cognitive Tutor [9] (referred to as the Cognitive Tutor or CT hereafter) is intensively used. There are two Algebra I classes in one school (N=40), and two Algebra I (N=30) and two Algebra II (N=34) classes in another school. A total of 104 students participated from the six algebra classes.

The study was a randomized control trial in each class. We used the Cognitive Tutor as a control condition. Since its effectiveness is well known [9], we conjectured that this comparison would provide a good sense of the effectiveness of learning by teaching relative to tutored-problem solving. We targeted a unit of the Cognitive Tutor where students learn to solve *equations with variables on both sides*. The students in the experimental condition (the SimStudent condition, or SS, for short) were asked to tutor Lucy equations with variables on both sides. The quiz for Lucy was designed to measure the competency at this level.

There were two days for the intervention where students used either APLUS or Cognitive Tutor for one full class period (40 minutes with an exception of 54 minutes in one Algebra I class). The students' and SimStudents' activities during the intervention were all logged automatically by the software, including problems tutored, feedback provided, steps performed, examples reviewed, hints requested, and quiz attempts. The expert-model module taken from the Cognitive Tutor was embedded into APLUS, but was only used to automatically assess the student's and SimStudent's actions for the logging purposes (i.e., the students did not receive any feedback on the correctness of their tutoring activities).

Pre- and post-tests were performed immediately before and after the intervention to measure students' competency in algebra equation solving as described in the next section.

4.2 Tests

Three versions of isomorphic online tests were used to counterbalance the pre- and post-test.[1] All three versions showed decent reliability scores; Cronbach's alpha for Test A = 0.83, B = 0.76, and C = 0.84.

The test has five subsections: (1) six equation-solving items (EQ) – students were asked to show their work on a piece of paper. (2) Effective next step (EFFECT) – 12 yes/no multiple-choice items to identify if a given operation is appropriate for a given equation. (3) Demonstration of errors (DEMO) – five items with a mixture of multiple-choice and free response to identify and explain an incorrect step in an *incorrect* solution for a given equation. (4) 38 yes/no multiple-choice items. Identify constant and/or variable terms in given expressions, and identify if two given expressions are like terms. (5) Equivalent expressions – 10 yes/no multiple-choice items to identify if a pair of expressions are equal.

[1] There was a delayed-test implemented 2 weeks after the intervention hence the three versions. We have yet to analyze the delayed test scores, thus their exclusion from this paper.

In the following analysis, average scores of subsections are used. We also used the overall score, which is the average of the five subsection averages.

4.3 Results

Although the total of 104 students participated to the study, only 74 and 79 students took the pre- and post-test respectively. Of those, only 57 students took both pre- and post-tests. In this section, we only include students who took both pre- and post-test. However, we further excluded five students from the analysis for apparent patterns of "gaming" on the EQ section (e.g., entering a sequence of numbers as the answers for the six equation items). As a consequence, 52 students (25 in SS and 27 in CT) were included in the current analysis.

4.3.1 Overall Test Scores

We first ran a regression analysis predicting the post-test score with the condition as a fixed factor and the pre-test score as a covariate. The adjusted post-test scores for each condition are CT = 0.44 + 0.37 * pre-test, and SS = 0.27 + 0.56 * pre-test. The condition is not the main effect on the adjusted mean post-test score: SS = 0.65 vs. CT = 0.67; F (1, 50) = 0.97, p<0.34.

4.3.2 Individual Section Scores

A mixed-design ANOVA revealed that there is a trend of interaction between the condition and the test on the EQ score; F(1, 50) = 3.48, p<0.07. There is a trend of a main effect on the test only for the CT condition: pre-test EQ mean = 0.54, post-test EQ mean = 0.64; t=1.71, p<0.1.

There is a significant aptitude-treatment interaction on the EFFECT score, F(1, 48)=4.43, p<.05, although the test is not a main effect. The centered polynomial regression on the EFFECT score shown in Fig. **3** confirmed that the difference between the condition intercepts is not significant.[2] This implies that *the students who scored on and above the average at the pre-test EFFECT score performed equally well on the post-test EFFECT score, regardless of the type of intervention (CT vs. SS), but those who scored below the average on the pre-test EFFECT score performed worse on the post-test EFFECT when assigned to the learning by teaching condition.*

For all other subsections, there was no significant condition effect or the test effect.

4.3.3 Learning Curve

How did students improve the accuracy in applying knowledge components during the intervention? To answer this question, we analyzed the learning curves.

The knowledge component used for this analysis was determined based on the features of the equation. There are four knowledge components defined: addsub-pos, addsub-neg, muldiv-pos, and muldiv-neg. They represent skills for adding/subtracting a term to/from both sides, and multiplying/dividing both sides with/by a term. The postfix, pos and neg, is determined by *a term in a given equation* that motivated a

[2] There are two apparent outliers, one at the top left corner and one at the lower right corner). Their absolute z-scores are higher than three; hence it is probable that they are outliers. When these two points are removed, however, the significance of the interaction remains intact.

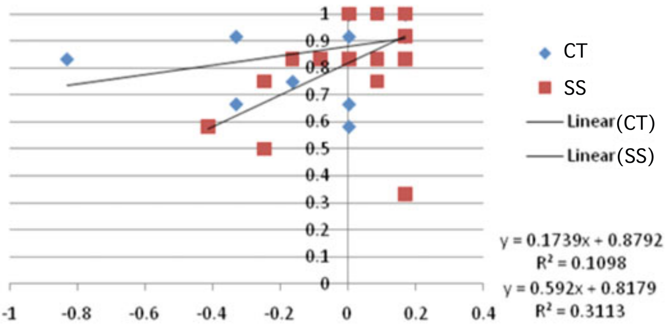


Fig. 3. A centered polynomial regression on the EFFECT score. The covariate (pre-test score) is normalized as a difference from the population mean.

particular operation. For example, "subtracting 2 from both sides of 3*x*+2=5*x*-1" is coded as addsub-pos, because it is arguably the *positive* term "+2" on the left-hand side that triggered this operation.

To compute a learning curve, we first coded the individual student's accuracy in applying knowledge components. For the Cognitive Tutor condition, for each step in solving an equation, the correctness of the application of a knowledge component is coded (correct or incorrect) based on the *first attempt* at a step. For the SimStudent condition, the correctness of the application of a knowledge component is coded based on the correctness of the student's feedback on a step performed by SimStudent or the correctness of the step performed by the student as a hint.

The learning curves shown in Fig. 4 are plotted by using the following regression model: $p_{iT} = \alpha + \beta K_i + \gamma K_T * T$ where p_{iT} represents the probability of making an *error* to apply knowledge component K_i on the T^{th} opportunity to apply K_i.

The significant decline of the probability of making errors shows that the students in both conditions improved the accuracy of applying knowledge components over

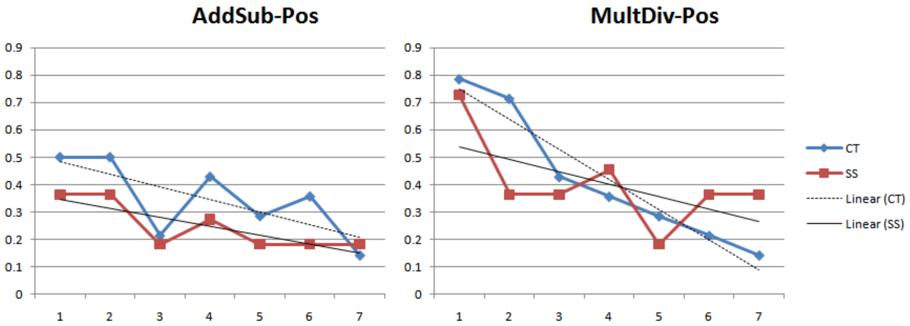


Fig. 4. A learning curve of student performance during the tutoring session. The x-axis shows the number of times the knowledge component was practiced to apply. The y-axis shows the probability making an *error* to apply the knowledge component.

time. This observation implies that the students could show better performance on the post-test than the pre-test, especially on the procedural test items (i.e., EQ, EFFECT, and DEMO; see 4.2). Yet, the data do not confirm such an improvement. The next section provides one hypothesis on this issue for the SimStudent condition.

4.3.4 Problems Used for Tutoring

For the SimStudent condition, one possible account for not seeing improvement on the test scores, even though the learning curves indicate improvement on the accuracy of applying knowledge components, is what might be called a *biased rehearsal effect*. Namely, *when tutoring SimStudent, the students might have repeatedly used only similar and perhaps "easy" problems, thus the overall accuracy of solving the problems improved as the tutoring session advanced.*

To test the hypothesis of this biased rehearsal effect, we analyzed the session log data to categorize the type of problems the students used for tutoring. There were 108 (17%) one-step equations, 259 (41%) two-step equations, and 256 (40%) equations with variable on both sides. A dominating number of problems (58%) were either one-step or two-step equations, which are not typical of the target for this current study. This observation provides positive support for the biased rehearsal effect.

5 Discussion

The current study is the first classroom experiment using APLUS and SimStudent as a teachable agent. Thus, the technical immaturity in the system may have interfered with the results. Despite intensive pilot and usability studies, there were still technical glitches observed during the study. Since the study, we have been running more usability tests and the system has been revised. In a more recent classroom study where the effect of self-explanation for tutor learning was tested, the students in the control condition that used the same APLUS system as in the current study showed significant improvement from pre- to post-test (will be reported elsewhere).

The significant aptitude-treatment interaction (ATI) on the EFFECT score (to identify an effective next step for a given equation) is a particularly important finding. The impact of learner's readiness for learning is one of the central issues in the sciences of learning [10]. The current study suggests that the prior knowledge significantly influences the effect of *tutor learning*, and (more importantly) that when the student is not trained beyond a certain threshold, he/she might receive more benefit from tutored-problem solving than learning by teaching. This makes sense because no one would likely be able to teach without a certain amount of knowledge about the subject. The current study provides an indication of what that threshold might be and opens the space for further investigation on this area.

Despite a favorable trend on the effect of the teachable agent for tutor learning [1], the current study did not confirm such an effect. One potential account for the lack of reproduction of the positive effect is the difference in domains. Betty's Brain is an example of a recent study that showed a positive effect of tutor learning, but that target knowledge is a declarative causal knowledge. The current study investigated the tutor learning effect in algebra equation solving that has more *procedural* skills involved in nature. Notably, Walker et al. [11] found no significant benefit of peer tutoring (within students alternating between tutoring and being tutoring) over being tutored in a procedural domain. The presence of the ATI suggests that there might be more requirements for prior knowledge when teaching procedural skills than when teaching declarative knowledge. If so, then, *it would be worth testing the hypothesis that learning by teaching has more effect for declarative domains than procedural domains*.

Our hypothesis concerning the biased rehearsal effect must also be tested. One way to avoid the bias on the problem selection is to have SimStudent express boredom on solving too many similar problems. Another idea is to embed a meta-tutor into APLUS to provide feedback on student's problem selection. In our upcoming new study, a bank of problems is available for students to review. The problem bank shows a wide variety of problem types used by the students in the previous studies with difficulty levels reflecting a ratio of successful vs. unsuccessful attempts made by SimStudents.

Acknowledgement

The research reported here was supported by National Science Foundation Award No. DRL-0910176 and the Institute of Education Sciences, U.S. Department of Education, through Grant R305A090519 to Carnegie Mellon University. This work is also supported in part by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation Award No. SBE-0836012.

References

1. Roscoe, R.D., Chi, M.T.H.: Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. Review of Educational Research 77(4), 534–574 (2007)
2. Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W.W., Koedinger, K.R.: Learning by Teaching SimStudent: Technical Accomplishments and an Initial Use with Students. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 317–326. Springer, Heidelberg (2010)
3. Matsuda, N., et al.: A Computational Model of How Learner Errors Arise from Weak Prior Knowledge. In: Taatgen, N., van Rijn, H. (eds.) Proceedings of the Annual Conference of the Cognitive Science Society, pp. 1288–1293. Cognitive Science Society, Austin (2009)
4. Biswas, G., et al.: Learning by teaching: a new agent paradigm for educational software. Journal Applied Artificial Intelligence 19(3&4), 363–392 (2005)
5. Michie, D., Paterson, A., Hayes, J.E.: Learning by teaching. In: Proc. of Second Scandinabian Conference on Artificial Intelligence, Tampere, Finland, pp. 413–436 (1989)
6. Bredeweg, B., et al.: DynaLearn - Engaging and Informed Tools for Learning Conceptual System Knowledge. In: Azevedo, P.R.R., Biswas, G. (eds.) AAAI Fall Symposium, Cognitive and Metacognitive Educational Systems (MCES 2009), pp. 46–51. AAAI Press, Arlington (2009)
7. Nichols, D.M.: Intelligent Student Systems: an Application of Viewpoints to Intelligent Learning Environments. Lancaster University, Lancaster (1993)

8. Palthepu, S., Greer, J., McCalla, G.: Learning by teaching. In: Proceedings of the International Conference on Learning Sciences, Illinois (1991)
9. Ritter, S., et al.: Cognitive tutor: Applied research in mathematics education. Psychonomic Bulletin & Review 14(2), 249–255 (2007)
10. Koedinger, K.R., Aleven, V.: Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. Educational Psychology Review 19(3), 239–264 (2007)
11. Walker, E., Rummel, N., Koedinger, K.R.: To Tutor the Tutor: Adaptive Domain Support for Peer Tutoring. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S., et al. (eds.) ITS 2008. LNCS, vol. 5091, pp. 626–635. Springer, Heidelberg (2008)