Allowing Revisions While Providing Error-Flagging Support: Is More Better?

Amruth N. Kumar

Ramapo College of New Jersey,
Mahwah, NJ 07430, USA
amruth@ramapo.edu

Abstract. In this study, we studied whether the number of revisions allowed per problem when error-flagging feedback is provided has a significant effect on learning. We used a partial cross-over study and analyzed the data collected by two adaptive tutors on while loops and for loops over six semesters. We found that when students were unfamiliar with the concepts, they solved fewer problems and therefore, learned significantly less when they were provided more opportunities for revision with error-flagging feedback. But, once they became more familiar with the concepts, allowing for more revisions had no deleterious effect on learning.

Keywords: Error-flagging feedback, Revisions, Adaptive tutor.

We had conducted several studies of the effect of providing error-flagging feedback, i.e., error-detection but not error-correction support, in the context of code-tracing tutors. In the first study [1], we found that students scored better on tests with rather than without error-flagging support even though the tests did not use multiple-choice format. In a follow-up study [2], we found that when error-flagging feedback was provided, students saved time on the problems that they already knew how to solve, and spent additional time on the problems for which they did not know the correct solution. But, we also found that students may abuse error-flagging support to find the correct solution by trial and error. In a subsequent study [3], we compared not providing error-flagging feedback against providing it with a limit placed on the number of revisions during testing. We found that even with a limit placed on the number of revisions per problem, students revised more often and scored higher with rather than without error-flagging feedback. We found that placing a limit on the number of revisions may discourage students from using error-flagging feedback as a substitute for their own judgment during tests.

In the current study, we wanted to study whether the number of revisions allowed per problem when error-flagging feedback is provided has a significant effect on *learning*. So, we compared error-flagging feedback with 3 revisions allowed per problem versus 5 revisions. We conducted the study using two tutors that did not use multiple-choice format. So, students could not guess the correct answer merely through brute-force trial-and-error in the presence of error-flagging feedback.

The two adaptive problem-solving tutors were on `while` loop and `for` loop. `while` loop tutor covered 9 concepts and `for` loop tutor covered 10 concepts in C++/Java/C#. The tutors presented code-tracing problems on these concepts: in each

problem, they presented a complete program and asked the student to identify the output of the program, one output at a time.

The tutors provided error-flagging feedback while the student was entering the solution to the problem (See bottom right panel in Figure 1). Once the student submitted the solution, if it was incorrect, the tutors provided step-by-step explanation of the correct solution in the style of a worked example [4,7].

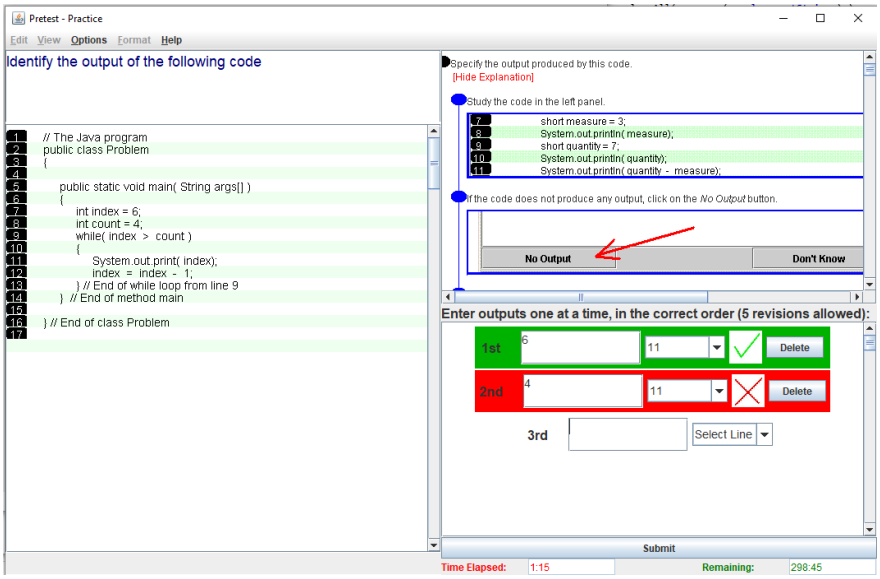

Figure 1: Error-Flagged answers in bottom right panel

The two tutors were configured to administer pre-test-practice-post-test protocol during each session [5]. During pretest, they administered one problem per concept. During adaptive practice that followed [6], they administered problems on only the concepts on which the student had solved the pretest problem incorrectly. They did so until the student demonstrated mastery of the concept by solving at least 60% of the problems correctly. During posttest, they administered problems on only the concepts mastered during practice. The tutors administered all three stages back-to-back online without interruption.

In this controlled study, the tutors allowed control group to revise the solution of each problem no more than 3 times and experimental group to revise the solution up to 5 times per problem. The interface always displayed the remaining number of revisions allowed for each problem (Title bar of bottom right panel in Figure 1). The duration of the tutoring session was set to 30 minutes for control group and 32 minutes for experimental group in order to accommodate additional revisions. It was also a partial cross-over study: students who were assigned to control group on `while` loop were assigned to experimental group on `for` loop and vice versa.

We used the data collected by the two tutors over six semesters: Fall 2014-Spring 2017. The tutors were used by students in introductory programming courses in C++, Java and C#. Typically, students used the tutors as after-class assignments. Students

could use the tutors as often as they pleased. Table 1 lists the number of students and the number of times they used the two tutors with each of the two treatments.

Table 1. Number of tutor users and uses in each treatment

	`while` loop	`for` loop
Max 3 revisions	1185 / 2162	1550 / 2957
Max 5 revisions	1647 / 2991	1141 / 2034

If a student solved the pretest problem incorrectly on a concept, solved sufficient number of problems during practice to demonstrate mastery, and went on to solve the post-test problem on the concept with a normalized score of at least 0.8, the student was categorized as having **learned** the concept. For our study, we used the number of concepts learned as a dependent variable.

In `while` loop tutor, students who were allowed 5 revisions learned significantly fewer concepts per session (1.30) than those who were allowed 3 revisions (1.40, $p = 0.02$). They solved significantly fewer problems than those who were allowed 3 revisions during practice (4.44 ± 0.32 with 3 revisions versus 3.87 ± 0.20 with 5 revisions, $p = 0.003$). One explanation for the differences is that students who were allowed more revisions engaged in more revisions and therefore, took longer to solve problems.

No such differences were found between treatments for `for` loop tutor. One explanation is that since students used `for` loop tutor after `while` loop tutor and the concepts covered by the two tutors were similar, students had less need for revisions in `for` loop tutor. *Students may revise their answers more when allowed more revisions when the concepts are unfamiliar to them. This may lead them to initially learn fewer concepts per session. But, with increased familiarity of concepts, students do not find the need to revise their answers as much, and any deleterious effect of allowing more revisions on the amount of learning fades.*

Mixed factor ANOVA analysis of `while` loop data of learned concepts with pretest and post-test score and pretest and post-test time as repeated measures and treatment (3 versus 5 revisions allowed) as between-subjects factor yielded:

- Significant within-subjects effect for score [$F(1,2349) = 3803$, $p < 0.001$]: mean score increased from 0.57 ± 0.01 on pretest to 0.99 ± 0.002 on post-test;
- Significant within-subjects effect for time [$F(1,2349) = 13.66$, $p < 0.001$]: time decreased from 94.95 ± 15.23 seconds on pretest to 66.64 ± 2.19 seconds on posttest;
- No significant between-subjects effect of treatment on score [$F(1,2349) = 1.67$, $p = 0.20$] or time [$F(1,2349) = 0.48$, $p = 0.49$] and no significant interaction between pre-post change in score and treatment [$F(1,2349) = 0.82$, $p = 0.37$] or pre-post change in time and treatment [$F(1,2349) = 1.3$, $p = 0.25$].

So, *students solved the post-test problem significantly more correctly and faster than pre-test problem, but there was no difference between treatments*. We found no significant main effect of treatment on the number of practice problems solved on the learned concepts, or the mean score per practice problem. But, we found a significant main effect of treatment on the mean time per practice problem solved [$F(1,2683] =$

8.29, p = 0.004]: students spent 68.73 ± 2.46 seconds per problem with 5 revisions compared to 63.80 ± 2.15 seconds per problem with 3 revisions. *So, students who were allowed 5 revisions spent significantly more time per practice problem than those who were allowed 3 revisions.*

Mixed factor ANOVA analysis of `for` loop data of learned concepts with pretest and post-test score and pretest and post-test time as repeated measures and treatment (3 versus 5 revisions) as between-subjects factor yielded:

- Significant within-subjects effect for score [F(1,2165) = 5140.84, p < 0.001]: mean score increased from 0.52 ± 0.01 on pretest to 1.00 on post-test;
- Significant within-subjects effect for time [F(1,2165) = 269.30, p < 0.001]: time decreased from 106.95 ± 5.80 seconds on pretest to 55.80 ± 1.95 seconds on posttest;
- Significant between-subjects effect of treatment on score [F(1,2165) = 5.33, p = 0.02]: Students who were allowed 3 revisions scored a mean of 0.75 ± 0.009 whereas, those who were allowed 5 revisions scored 0.77 ± 0.01. The interaction between pre-post and treatment was also significant [F(1,2165) = 5.09, p = 0.02]: students who were allowed 3 revisions improved from 0.51 on pretest to 0.997 on post-test whereas those who were allowed 5 revisions improved from 0.54 on pretest to 0.997 on post-test. We discounted this result because of ceiling effect, 1.0 being the maximum normalized score per problem.
- No significant between-subjects effect of treatment on time [F(1,2165) = 1.83, p = 0.18] or interaction between pre-post time and treatment [F(1,2165) = 0.53, p = 0.47].

Again, *students solved the post-test problem significantly more correctly and faster than pre-test problem, but the difference between treatments was minimal*. We found no significant main effect of treatment on the number of practice problems solved on the learned concepts, the mean score per practice problem or the mean time per practice problem solved. In contrast, treatment had a significant effect on mean time per practice problem solved on `while` loop tutor, the first tutor to be used by students. This once again reinforces that any negative effect of allowing for more revisions wears out with increased familiarity with the concepts.

Students did not score more per problem when allowed more revisions – so, allowing for revisions with error-flagging feedback was not a substitute for knowing the concepts underlying problems. They did not score less per problem either, although they spent more time per problem on `while` loop tutor. This might suggest that allowing for more revisions with error-flagging by itself may not invite gaming of the system by students, especially when solutions to problems are not of multiple-choice nature.

In this study, we evaluated the effect of allowing a limited number of revisions (as saliently displayed in the user interface of the tutor), not the effect of the number of revisions actually undertaken by students. In the future, we plan to analyze the data to check whether allowing for more revisions invites students to revise more, and if not, the effect of the number of revisions actually undertaken by students on the learning of students.

Acknowledgments. Partial support for this work was provided by the National Science Foundation under grant DUE-1432190**.**

References

1. Kumar, A.N. Error-Flagging Support for Testing and its Effect on Adaptation. In: Proc. Intelligent Tutoring Systems (ITS 2010), LNCS 6094, pp 359-368. (2010)
2. Kumar, A.N. Error-Flagging Support and Higher Test Scores. In: Proc. Artificial Intelligence in Education (AI-ED 2011), LNAI 6738, pp 147-154. (2011)
3. Kumar, A.N. Limiting the Number of Revisions While Providing Error-Flagging Support During Tests. In: Proc. Intelligent Tutoring Systems (ITS 2012). LNCS 7315, pp 524-530. (2012)
4 Kumar, A.N.: Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors. Technology, Instruction, Cognition and Learning. (TICL) J. Special Issue on Problem Solving Support in Intelligent Tutoring Systems, 4(1) (2006)
5 Kumar, A.N., A Model for Deploying Software Tutors, IEEE 6th International Conference on Technology for Education (T4E), Amritapuri, India, 12/18-21/2014, 3-9.
6 Kumar, A.: A scalable solution for adaptive problem sequencing and its evaluation. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 161–171. Springer, Heidelberg (2006)
7 Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V. and Salden, R. The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior.* Vol 25(2). March 2009. 258-266.