Learning with an algebra computer tutor: What type of hint is best?

Kyle Sale (kyle.sale@carleton.ca)
Institute of Cognitive Science, Carleton University
Ottawa, Ontario, Canada

Kasia Muldner (kasia.muldner@carleton.ca)
Institute of Cognitive Science, Carleton University
Ottawa, Ontario, Canada

Abstract

While there is substantial evidence showing that assistance provided to students during problem-solving activities influences learning outcomes, it is not yet clear how to best design educational technologies to maximize learning through various types of assistance. One common type of assistance corresponds to hints delivered by an educational technology. To date, however, there is little research on the impact of different types of hints, including high-level hints vs. specific bottom-out hints. Our research takes a step in filling this gap, through an experimental study with an intelligent tutoring system we implemented in the domain of algebra (N = 50). We did not find evidence that the type of hint, high level vs. bottom out, influenced learning, with both types of hints producing similar outcomes. We did, however, find support for the conclusion that the number of hints accessed interacted with the type of hint to influence learning, and specifically, that accessing more hints was correlated with learning but only in the high-level hint condition.

Keywords: Intelligent Tutoring Systems; high-level and bottom-out hints

Introduction

There is established evidence that instructional feedback and assistance, such as hints and explanations during instructional activities, influence student learning (Shute, 2008). An open question, however, is how explicit should this assistance be to facilitate learning?

Prior research suggests that students learn best when they engage in constructive behaviors as compared to ones that are merely active or passive. This is a key prediction made and confirmed by Chi's (2009) ICAP framework that distinguishes levels of student engagement during instructional activities. To illustrate in the context of human tutoring, when a tutor prompts their student with general suggestions and/or questions, this encourages the student to generate substantive contributions, namely domain-related utterances that are positively associated with learning (Chi, Roy, & Hausmann, 2008). As another example, Ferreira, Moore, and Mellish (2007) compared two common strategies human tutors used to respond to student errors and misconceptions, namely *giving-answer assistance* and *prompting-answer* assistance. They found that *giving-answer* type of assistance occurred more often, but *prompting-answer* type of assistance was more effective for learning. Thus, in the context of human tutoring, tutors don't encourage constructive student processing because they provide the answer instead of eliciting it from the student. When students are working on their own without a tutor, they also default to passive strategies. For instance, VanLehn (1991, 1998) showed that when students were given access to worked-out examples during paper and pencil problem-solving activities, they commonly missed learning opportunities because they copied from the examples rather than trying to generate the problem solution without the help of the example.

The findings on learning from human tutoring and related activities have influenced the design of educational technologies, including that of *tutoring systems*. These technologies rely on artificial intelligence techniques to personalize instruction, in some cases approaching the effectiveness of human tutors (Vanlehn, 2011). Based on research that students benefit from active processing and that reduced assistance may promote it, some work has examined the effects of manipulating assistance in computer tutors. For instance, in separate experiments, Borracci, Gauthier, Jennings, Sale, and Muldner (2019) and Lee, Betts, and Anderson (2015) found that students learn better from tutoring systems that provide reduced assistance as compared to high assistance. In these studies, assistance was operationalized through examples that aided problem solving, with the level of similarity between an example and its corresponding problem determining how much assistance the example provided (high similarity resulted in high assistance, low similarity in reduced assistance). We next review tutoring systems that provide assistance through hints.

A common way to integrate hints into tutoring systems is to use a specific progression of assistance, one that starts off general with hints that provide high-level suggestions, but that become more specific as students ask for more help (Arroyo, Mehranian, & Woolf, 2010; Roll, Aleven, McLaren, Ryu, Baker, & Koedinger 2006; Vanlehn, Lynch, Schulze, Shapiro, Shelby, Taylor, Treacy, Weinstein, &

Level 1: *Check your trigonometry* **Level 2:** *If you are trying to calculate the component of a vector along an axis, here is a general formula that will always work: Let qV be the angle as you move counterclockwise from the horizontal to the vector. Let qx be the rotation of the x- axis from the horizontal. (qV and qx appear in the Variables window.) Then: V_x = V*cos(qV-qx) and V_y = V*sin(qV-qx).* **Level 3: (bottom-out hint)** *Replace cos(20^o) with sin(20^o)*	**Level 1:** *Enter the value of the radius of circle A* **Level 2**: *How can you calculate the value of the radius of circle A given the value of the diameter of circle A*? **Level 3**: *The radius of a circle is half of the diameter* **Level 4: (bottom out hint):** *The radius of circle A ¼ 46.5*

Figure 1. The hint progression sequence in two established tutoring systems: Andes (left) and the Cognitive Geometry tutor (right)

Wintersgill, 2005). The final hint in the progression is commonly referred to as a *bottom-out* hint, and this type of hint essentially provides the answer (e.g., the solution step the student needs to produce to make progress in solving the problem). To illustrate, Figure 1 shows an example of such a hint progression from two established tutoring systems: (1) the *Andes* tutor in the domain of physics (Vanlehn et al., 2005) and (2) the *Cognitive Geometry* tutor (Aleven, McLaren, Roll, & Koedinger, 2006).

The rationale behind using a hint progression that starts with high-level hints is to encourage students to be constructive and so generate the answer with minimal assistance from the high-level hint; if students continue asking for help, they are given more specific assistance. While this type of design mirrors what *expert* human tutors do (i.e., start off more general in their assistance and only provide the answer if students are truly stuck), in the context of tutoring systems students often abuse help functionalities (Aleven et al., 2006; Muldner, Burleson, Van de Sande, & VanLehn, 2011; Peters, Arroyo, Burleson, Woolf, & Muldner, 2018), a behavior referred to as *gaming* (Baker, Corbett, Koedinger, & Wagner, 2004). In the context of systems that make hints available, students who “game” tend to quickly and repeatedly ask the tutoring system for a hint, without reading the high-level hints, until they reach the bottom-out hint in the hint progression, at which point they copy the answer the hint provides into the problem they are working on.

Skipping high-level hints in tutoring systems is a well-documented event (e.g., Arroyo et al., 2010, Muldner et al., 2011). How does this behavior impact learning? Some argue that students still learn because they use the bottom-out hints as worked examples, which may promote learning in ways that abstract hints do not (Shih, Koedinger & Scheines, 2011). This conclusion was reached through a data mining analysis. Others have found more mixed findings on the utility of either type of hint. To illustrate, Muldner et al. (2011) used exploratory methods corresponding to Bayesian parameter machine learning to investigate the utility of high-level and bottom-out hints. Specifically, to model learning from hints, a knowledge-tracing Bayesian network was used that included nodes representing student actions, knowledge of domain principles (rules), and hints. The network encoded the probability that students will learn a rule given that they saw a certain type of hint (high-level vs. bottom-out). To obtain those probabilities, machine learning was applied to learn the parameters from data corresponding to students interacting with the Andes tutoring system. The findings showed that neither type of hint was very effective at promoting learning and there was little difference between the two types of hints. Specifically, the probability of a rule being learned was only at about 25% when a hint was used and this value was similar for both bottom-out and high-level hints.

The work cited above used exploratory methods to investigate the utility of different types of hints. The motivation for the present study is that to date there is very little experimental work comparing the effect of different types of hints on student learning. One exception is the study by Chi et al. (2001), albeit this work involves human rather than artificial tutors. Specifically, Chi et al. (2001) manipulated the type of hint human tutors were allowed to give: high-level prompts only vs. detailed hints. The results indicated a lack of a difference in learning between the two conditions, with similar posttest scores. In contrast to this study, our work investigates the effect of different types of hints provided by a computer tutor, as we now describe.

The Present Study

To test how different types of hints influence learning from a tutoring system, we created a computer tutor using the Cognitive Tutor Authoring Tools (CTAT) framework (Aleven, McLaren, Sewall, & Koedinger, 2006). CTAT facilitates the construction of tutoring systems by providing tools that a human author uses to create the tutor interface and specify the tutor’s behavior. For the latter, a human author creates a *behavior graph* for each problem that specifies the tutor’s behavior for that problem (e.g., what kinds of hints to show, what feedback to provide on solution entries, what to do if a student wants to move on to the next problem).

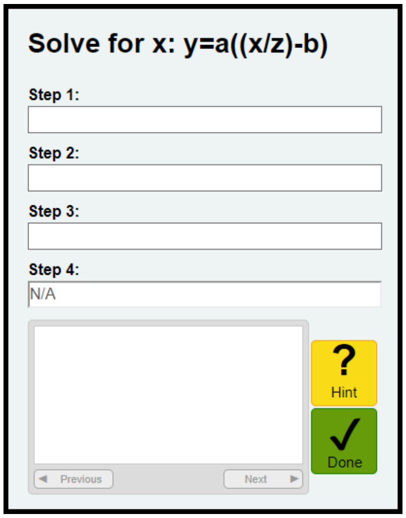

Figure 2. A problem in the algebra tutor

Table 1
Examples of hints used in the algebra tutor

Hint Type	Example
Bottom-out Hint	Enter $ya=(x/z)+b$ into the highlighted field.
High-Level Hint (Level 1)	x isn't isolated (alone on one side of the equal sign). So we must reverse the operations acting on the variable(s), starting with the outermost ones (i.e. a in $y=xa$)
High-Level Hint (Level 2)	For this step, you need to move b to the opposite side of the question using addition.

The algebra hint tutor

Our tutor provided students with problems to solve in the target domain of algebra (see Figure 2). The problem format was adopted from prior work and used variables instead of numeric constants (e.g., similar to the approach used by Cooper and Sweller, 1987).

All problems in the tutor required three to four solution steps for the final solution; each step was produced by applying algebraic manipulations (i.e., rules) to the prior step (or specification if the current step was the first one). A single algebraic manipulation corresponded to moving a variable from one side of the equation to the other. For example, given the equation $y = (a+x)b$, a manipulation required to solve the problem for the variable x involves moving the b variable to the other side of the equation, resulting in the equation $y/b = a+x$. Each solution step had its own input box in the tutor's interface that the student could type into. The tutor provided two forms of support: (1) feedback for correctness and (2) on-demand hints.

Feedback for correctness was realized by having the tutor color a student's entry as red (incorrect entry) or green (correct entry) directly after students indicated they were done with the entry by hitting the *return* key. The tutor was flexible in terms of accepting various forms of solutions, e.g., recognized $x = yab$ as equivalent to $x = a *y(b)$. This flexibility was accomplished through functionality we added to the tutor following the algorithm proposed by Shapiro (2005). This algorithm involves using mathematical calculations to check for equivalence without pre-storing all possible versions of a solution, thus saving significant development effort as well as computational cost of evaluating student solutions. To further scaffold the solution entry process, the solution steps had to be entered in the order required by the algebraic process and steps could not be skipped. Once a problem was done (all steps were correctly generated), students clicked the *Done* button (see Figure 1) to move on to the next problem.

As they were solving problems, students could ask for a hint, done by clicking on the *Hint* button in the interface (see Figure 1). We created two different versions of the tutor: one version provided only *bottom-out* hints and the other provided only *high-level* hints. To design the wording of the hints, we consulted existing tutoring systems as well as online educational sites specific to algebra. To check the wording of the hints was appropriate, we conducted several rounds of pilots.

Bottom out hints Bottom-out hints told students the exact equation they had to enter (see row 1, Table 1), and thus provided high assistance to problem solving. These hints were context specific, meaning that if the student entered a part of the solution and then asked for a hint, the hint would correspond to the next step they had to enter.

High-level hints High-level hints provided reduced assistance because they only prompted the student without giving the answer away. There were two levels of this type of hint: level 1 prompted the student about the next goal they needed to fulfill, but in contrast to a bottom-out hint did not specify exactly how to do that (see row 2, Table 1). If the student wanted further help, they could click the hint button again to access a level 2 hint. This type of hint specified the required operation and the variable that would be moved as a result (see row 3, Table 1).

Like the bottom-out hints, the high-level hints were context specific, and tailored to the student's problem-solving progress. For example, if the next step that had to be entered corresponded to the equation $y\text{-}a = (x+b)/c$, the hint would tell the student to move the variable c over to the other side of the equation using multiplication. In instances where two different manipulations were possible, the tutor would pick one at random (students could enter the steps in whichever order they wished). To avoid the hints sounding repetitive, we created several variations of each and the tutor cycled randomly through these variations. We chose to have the high level hints include prompts for both the variable and the operation because both were integral to the solving the problem.

We did not include bottom-out hints in the high-level hint tutor version because we wanted to investigate whether general prompts alone would be sufficient to foster learning.

Each of the two versions of the algebra tutor were populated with the same 12 algebra problems (all required 3-4 steps for their solutions, of the type shown in Figure 2). Both tutor versions logged all student actions in the tutor. We used a basic python script to extract the salient information from the log files (e.g., number of hints, number of errors).

Participants

The participants (N = 50) were undergraduate students at a Canadian University recruited via Sona and compensated with course credit. To be eligible for the study, participants could not have taken or be currently enrolled in any university-level math courses.

Materials

To assess algebra knowledge, we used a paper and pencil algebra pretest and posttest from our prior research that included 11 questions (Borracci et al., 2019). The tests were equivalent (only variable names were changed between them). The tests were scored out of 40, with the points for a given question corresponding to the number of rule applications needed for the question's canonical solution. For instance, if a question required three rule applications for its solution, its point value was three. This scoring method is more sensitive than marking a question as correct or incorrect, given that each question required multiple rule applications.

Several other questionnaires were used in the study to measure personality traits but we do not describe them as we do not include analysis from their data here.

Design

We used a between-subjects design with two conditions: *high-level hints* (participants used the version of the alebra tutor that included only high-level hints) and *bottom-out hints* (participants used the version of the tutor that included only bottom-out hints). As noted above, the problems solved in both conditions were identical, and the only difference between the two conditions was the type of hint available in the tutor.

Procedure

Each session was conducted individually and lasted approximately 90 minutes (the duration varied slightly based upon the amount of time participants spent on the various components). The procedure for the two conditions was the same.

Participants first completed the algebra pretest (they had up to 20 minutes to do so). They then filled in a demographics questionnaire and were assigned to their condition. Participants initially were assigned to a given condition in a round robin fashion; after about 10 participants, we began using a matching procedure based on pretest score with the goal of equalizing pretest scores between the two conditions, while maintaining similar sample size between the two conditions[1]. The experimenter then introduced participants to the algebra tutor, and explained its various features (e.g., that feedback for correctness was provided, and that all solution steps had to be correctly generated for a given problem before moving on to the next problem). Participants were told to treat this part of the study as if it were a homework situation: they had some problems to solve and were doing so to prepare for an upcoming test. Once participants confirmed they understand how to use the tutor, they were given 40 minutes to complete the 12 problems in their respective tutor version. Participants then completed the algebra posttest (20 minutes), and the personality questionnaire (10 minutes).

Results

The analysis is based on 47 participants. We excluded from the analysis three participants who were at ceiling on pretest, i.e., 95% or higher.

Does type of hint influence learning?

The descriptives for the pretest and posttest are in Table 1. Before checking if the type of hints influenced how much students learned from pretest to posttest, we verified there was no significant difference in pretest scores between the two conditions – this was the case (p = .24).

A between-subjects ANCOVA with *pretest* as the covariate, *posttest* as the dependent variable, and *condition* (high-level hints, bottom-out hints) as the independent variable did not find a significant effect of condition, $F(1, 44) = .1$, $p = .75$ and the effect size was very small, $\eta_p^2 < .01$. As shown in Figure 3, the mean posttest scores adjusted by the pretest through the ANCOVA were very similar in the two conditions. There was also no significant effect of condition on performance as measured by the number of errors made during problem solving (we extracted this information from the log files). Specifically, as expected on average participants made more errors with high-level hints, $M = 29.2$, $SD = 20.1$, as compared to with bottom-out hints, $M = 23.0$, $SD = 15.7$, but this difference was not significant, $t(45) = 1.2$, $p = .25$.

Thus, we did not find evidence that the type of hint provided influenced either learning from the algebra tutor or overall performance. However, it may be the case that that the number of hints participants accessed influenced learning differently depending on the condition. The next analysis investigates this possibility.

[1] The pretests were graded during the experimental session but in a separate room to avoid making participants uncomfortable. To save time, we used a coarser grading scheme than for the present analysis (where each question was assigned one point it was fully correct and zero points otherwise).

Table 1
Descriptive statistics for each condition

	Bottom-out hints (n = 24)		High-level hints (n = 23)	
	M	*SD*	*M*	*SD*
Pretest (/40)	10.8	14.2	15.7	14.3
Posttest (/40)	26.2	11.3	28.1	9.5

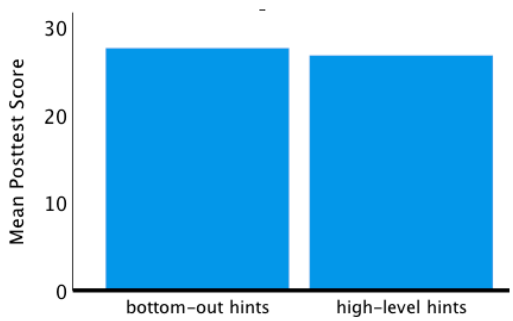

Figure 3. Posttest scores in the two conditions (adjusted by the pretest covariate); posttest was out of 40

What is the relationship between number of hints accessed and learning in each condition?

On average, participants requested more hints in the bottom-out hint condition (M = 15.6, SD = 17.5) than in the high-level condition (M = 13.4, SD = 15.1). This finding is not surprising given that the bottom-out hints facilitated problem solving by telling the students precisely what to do. To get a preliminary view of how the number of hints accessed influenced learning in each condition (operationalized as posttest score – pretest score), we plotted the relationship between these two variables for each condition. As shown in Figure 4, the relationship between the number of hints accessed and learning in the high-level hint condition is positive: the more participants accessed the high-level hints, the more they learned. In contrast, the slope of the line characterizing this relationship in the bottom-out hint condition is almost flat, suggesting there is little association between learning and number of bottom-out hints accessed.

We formalized this analysis by conducting a regression. In preparation, we dummy coded the condition variable so that the bottom-out hint condition was assigned the value 0 and the high-level hint condition the value 1 (the choice of which variable to assign the value 1 is arbitrary and does not impact the results). We proceeded with the regression by entering *posttest* as the outcome variable, and the following four predictors: *pretest*, *condition*, *number-of-hints requested*, and *number-of-hints requested x condition*.

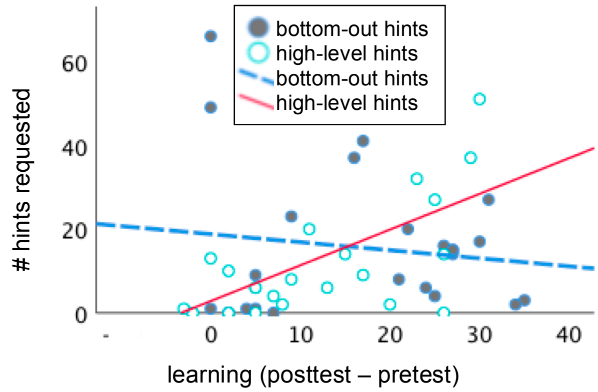

Figure 4. Relationship between number of hints accessed and learning for each condition

The overall model we obtained, shown in Table 2, was significant, $F(4, 42) = 11.3$, $p < .001$, $R^2 = .52$. Of primary interest is the interaction term (i.e., *number-of-hints x condition*), which informs on whether condition influenced the impact of number of hints requested on posttest score. Since the interpretation of the other coefficients is affected by the interaction term (Braumoeller, 2004), which essentially renders them "baseline" slopes (Grace-Martin, 2000), they are not discussed here. The interaction is modest but significant and indicates that overall, the number of hints accessed had a stronger positive relationship with posttest for high-level hints, as compared bottom-out hints. This conclusion is based on the fact that the coefficient for the interaction term is positive, indicating that when students were given high-level hints (recall this was dummy-coded as 1), their posttest score increased by the corresponding amount, controlling for the influence of the other predictors.

Table 2
Linear regression coefficients

Predictors	*B*	*β*	*t*	*p*
# hints x condition	**.34**	**.4**	**2.3**	**.022**
# hints	-.37	-.57	3.6	.001
condition	-5.4	-.26	1.7	.091
pretest	.32	.43	3.2	.003

B = Unstandardized Coefficients
β = Standardized Coefficients

Do high-level hints promote more active processing than bottom-out hints?

High-level hints offer reduced assistance because they don't tell the student the answer directly. Thus, these types of hints should promote more constructive processing on the part of the student. One way to check this is to analyze the amount of time students spent on a solution entry after they saw a high-level hint and compare that to the other condition in which students only saw bottom-out hints. Students spent longer generating a solution entry after seeing a hint in the high-level hint condition (M = 18.0 sec, SD = 6.1) than after seeing a hint in the bottom-out hint condition (M = 21.2 sec, SD = 7.1). This trend did not reach significance but approached it after controlling for pretest score, $F(2, 34) = 2.8$, $p = .1$, $\eta_p^2 = .08$. While this result is somewhat expected as the high-level hints provided less information, it does open up the possibility that students in the bottom-out condition were not actively processing the contents of the hint before asking the tutor to check their answer (i.e., by pressing the enter button as soon as they finished entering the solution step).

Another way to check if hints are influencing student processing is to analyze how long students waited after entering a solution step (and receiving feedback on it) before pressing the hint button. If there are differences between conditions, this may suggest different levels of processing taking place for each group. Note that the alternative action after entering a solution step is to enter another solution step – here we focus on the subset of actions after a solution entry pertaining to hints only because are interested in conditional effects of hints. When students requested a hint after generating a solution entry, they waited significantly longer to do so in the high-level condition (M = 17.6 sec, SD = 23.9) as compared to the bottom-out condition (M = 6.3 sec, SD = 3.5), $F\ (2, 37) = 5.2$, $p = .029$, $\eta_p^2 = .12$ (controlling for pretest does not affect this result). The large standard deviation in the high-level condition implies there is a lot of variability in this condition. To ensure extreme values were not affecting the result, we removed 3 outliers flagged by SPSS and re-ran the analysis. The results remained significant and so the outliers were not influential.

Discussion

The present study investigated the utility of two types of hints in the context of a tutoring system: bottom-out hints that told students exactly what step was needed to proceed with problem solving, versus high-level abstract hints that merely suggested at what was needed to generate the corresponding problem solution step. Thus, the two types of hints provided high vs. reduced assistance to problem solving, respectively. We did not find evidence that either type of hint had a differential impact on learning and in fact the learning outcomes were very similar between the two conditions. While we recognize that conclusions can not be drawn from non-significant results, these findings echo prior experimental results (e.g., Chi et al., 2011). Our findings also echo exploratory studies using machine learning to investigate student learning from different types of hints (Muldner et al., 2011) - this latter work also did not find a difference in learning from the two types of hints.

If high-level hints are not more effective for learning, are they a less *efficient* instructional tool because they take longer to process and thus increase time on task? When we checked total time spent in each condition, we did find a trend that students overall took longer in the high-level hint condition (while this did not reach significance, that may be due to lack of power given the high variability). If high-level hints do not produce more learning than bottom out-hints but are less efficient, then that is an argument for not using them. Our subsequent analysis, however, suggested a more nuanced view of each type of hint's impact, where the number of hints students accessed interacted with the type of hint available to influence learning. It may be that students benefited from both types of hints, but that if they accessed too many bottom-out hints, they failed to learn effectively because they could not resist passively copying from the hints. Prior research in example-based learning found this type of pattern, with students copying indiscriminately from examples (VanLehn, 1998). In contrast to bottom-out hints that promote more passive cognitive processing, high-level hints in general may encourage learning because they promote active processing of the hint content, needed to infer the additional information not provided by the hint. While we did not find strong evidence in this regard, we found some indications: (1) the number of hints accessed was positively associated with learning in the high-level hint condition, and (2) students waited longer in the high-level condition to request a hint, suggesting they were less reliant on assistance provided by the tutoring system and thus more constructive. Promoting constructive processing is generally important, but may be especially challenging to realize when students are interacting with tutoring systems rather than human tutors due to accountability (i.e., students may feel less accountable with technologies than humans), although this conjecture awaits validation through future studies.

A limitation of our study is that we only measured short-term learning. High-level hints require students to process the material, possibly using common-sense or overly general reasoning to infer new rules (Vanlehn, 1991). The benefit of these types of hints may not show up until some time has passed, and so a delayed post-test would be beneficial to include in future studies to measure retention in each condition. Another limitation is the modest sample size, highlighting the need for replication. In general, given the relatively little research on what types of hints best promote learning in tutoring systems, more work is needed to validate and extend our findings.

Acknowledgements

This work was supported with an NSERC Discovery Grant #1507 and a NSERC USRA grant.

References

Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education, 16*(2), 101-128.

Aleven, V., McLaren, B., Sewall, J., van Velsen, M., Popescu, O., Demi, S., ... & Koedinger, K. R. (2016).

Example-tracing tutors: intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education,* 26(1), 224-269.

Arroyo, I., Mehranian, H., Woolf, B. (2010) Effort-based Tutoring: An Empirical Approach to Intelligent Tutoring. In *Proceedings of the 3rd International Conference on Educational Data Mining*, 10 pages.

Baker, R., Corbett, Koedinger, K., & Wagner, A. (2004). Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". In *Proceedings of the ACM CHI 2004: Computer-Human Interaction (CHI'04)*, 383- 390.

Borracci, G., Gauthier, E., Jennings, J., Sale, K., & Muldner, K. (2019). The Effect of Assistance on Learning and Affect in an Algebra Tutor. *Journal of Educational Computing Research*.

Braumoeller, B. F. (2004). Hypothesis testing and multiplicative interaction terms. *International Organization,* 58, 807–820.

Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science*, 25, 471–533.

Chi, M. T. H., Roy, M., & Hausmann, R. G. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science*, 32(2), 301–341.

Chi, M. T. H. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1), 73–105.

Cooper, G. & Sweller, J. (1987) Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, 79(4), 347–362.

Ferreira, A., Moore, J., & Mellish, C. (2007). A study of feedback strategies in foreign language classrooms and tutorials with implications for intelligent computer-assisted language learning systems. *International Journal of Artificial Intelligence in Education, 17*(4), 389-422.

Grace-Martin, K. (2000). Interpreting interactions in regression. Retrieved from http://www.cscu.cornell.edu/ news/statnews/ stnews40.pdf

Lee, H. S., Betts, S., & Anderson, J. R. (2015). Not taking the easy road: When similarity hurts learning. *Memory & Cognition*, 43(6), 939–952.

Muldner, K., Burleson, W., Van de Sande, B., & VanLehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. *User modeling and user-adapted interaction,* 21(1), 99-135

Peters, C., Arroyo, I., Burleson, W., Woolf, B, & Muldner, K. (2018). Predictors and outcomes of gaming in an intelligent tutoring system. In *Proceedings of the Conference on Intelligent Tutoring Systems,* 366-372.

Roll I., Aleven V., McLaren B., Ryu E., Baker R., & Koedinger K. (2006) The Help Tutor: Does Metacognitive Feedback Improve Students' Help-Seeking Actions, Skills and Learning. In proceedings of *Intelligent Tutoring Systems Conference*, 10 pages.

Shapiro, J. (2005). An Algebra Subsystem for Diagnosing Students' Input in a Physics Tutoring System. *International Journal of Artificial Intelligence in Education,* 15(3), 205-228

Shih, B., Koedinger, K., & Scheines, R. (2011). A response time model for bottom-out hints as worked examples. *Handbook of educational data mining,* 201-212.

Shute, V. (2008). Focus on formative feedback. *Review of educational research*, 78(1), 153-189.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem solving strategies. *Cognitive Science*, 15(1), 1-47.

VanLehn, K. (1998). Analogy events: How examples are used during problem solving. *Cognitive Science,* 22(3), 347-388.

Vanlehn, K., Lynch, C., Schulze, A. Shapiro, Shelby, Taylor, Treacy, Weinstein, & Wintersgill. (2005). The Andes Physics Tutoring System: Lessons Learned. *Int. Journal of Artificial Intelligence in Education*, 15(3),147-204.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist,* 46(4), 197-221.