Does Knowing When Help Is Needed Improve Subgoal Hint Performance in an Intelligent Data-Driven Logic Tutor?

Nazia Alam, Mehak Maniktala, Behrooz Mostafavi, Min Chi, Tiffany Barnes
North Carolina State University
nalam2@ncsu.edu, mmanikt@ncsu.edu, bzmostaf@ncsu.edu, mchi@ncsu.edu, tmbarnes@ncsu.edu

Abstract

The assistance dilemma is a well-recognized challenge to determine when and how to provide help during problem solving in intelligent tutoring systems. This dilemma is particularly challenging to address in domains such as logic proofs, where problems can be solved in a variety of ways. In this study, we investigate two data-driven techniques to address the when and how of the assistance dilemma, combining a model that predicts *when* students need help learning efficient strategies, and hints that suggest *what* subgoal to achieve. We conduct a study assessing the impact of the new pedagogical policy against a control policy without these adaptive components. We found empirical evidence which suggests that showing subgoals in training problems upon predictions of the model helped the students who needed it most and improved test performance when compared to their control peers. Our key findings include significantly fewer steps in posttest problem solutions for students with low prior proficiency and significantly reduced help avoidance for all students in training.

Introduction

Intelligent tutoring systems (ITS) foster effective learning by providing necessary instructions, feedback, and hints to students as needed. Providing assistance can be useful to students but may also lead to shallow cognitive processing, and on the other hand, withholding assistance can encourage the students to learn on their own, but it can also generate frustration among students (Koedinger and Aleven 2007). Determining when a student needs help and how to help a student to optimize the learning process is an active research problem known as the assistance dilemma. Solving the assistance dilemma is particularly challenging in open-ended problem-solving domains such as logic and programming. There has been much research on how to help students learn better with the help of various types of feedback and hints such as on-demand hints and proactive hints. When students are given explicitly requested hints, this is known as on-demand hints. Proactive hints are unsolicited hints given to students by the tutor. The assistance dilemma arises from the finding that students often fail to display proper help-seeking behavior(Aleven et al. 2006). Sometimes students avoid seeking help although they are in need of it (known as help avoidance). Other times the other extreme is observed, where students request help unnecessarily (known as help abuse). When students fail to seek help in an effective way, it can hinder the overall learning process.

The assistance dilemma consists of two parts: the first part is to determine if and when students need help, and the second part is to determine how and what help to provide. In this study, we aim to address the assistance dilemma by developing a new pedagogical policy that tackles both parts.

A recent study (Maniktala et al. 2020b) presented the HelpNeed model which predicts *when* students need help. It has been shown to improve student performance by predicting when to provide assistance to students and providing proactive next-step hints [1] accordingly in a logic tutor. However it did not explore other kinds of hints such as high level hints. [2] There are several studies indicating that high level hints such as problem-solving subgoals can improve student learning and performance. According to Catrambone (Catrambone 1998), "A subgoal represents a meaningful conceptual piece of an overall solution procedure". Cody et al. in a recent study (Cody et al. 2022) derived data-driven subgoals for logic tutor called Waypoint hints, that are a few steps ahead of the current problem state. However, these hints were provided periodically to students and not necessarily when a student needs them. Waypoint hints alone, when given in a random manner did not show any significant student performance improvement. However, based on correlation analysis Waypoint hint usage was both found to be associated with shorter and faster proofs. In this study, we intend to explore the use of problem level subgoals as hints, and evaluate whether knowing when help is needed when providing subgoal hints can lead to good performance. We hypothesize that combining the *when* from HelpNeed model and the *what* from Waypoints may be effective in providing necessary assistance to student and lead to better training behavior and posttest performance.

Our main contributions in this study are as follows: we introduce a new pedagogical policy to address the assistance

[1] Next-step hints are hints which provide the immediate next step of the solution.

[2] High level hints are those which take more than one step to derive.

dilemma. We investigate the impact of the pedagogical policy in an intelligent tutor teaching the open-ended domain of logic proofs. We present a study analyzing the impact of the new Adaptive policy against a Control policy without the adaptive components. We also explore the role of prior proficiency in student outcomes and assess the hint usage in the two conditions.

Related Work

Assistance Dilemma and Proactive Hints

Students often use the assistance at hand in an unproductive manner (Roll et al. 2006). Aleven et al. (Aleven et al. 2006) defined help avoidance as "situations in which the student could benefit from asking for a hint but chose not to", and help abuse is categorized as "situations in which the student misuses the help facilities or uses them unnecessarily". A number of intelligent tutoring systems provide proactive hints or feedback to prevent help avoidance. Fossati et al. (Fossati et al. 2010) developed a linked list tutor which generates proactive feedback using a procedural knowledge model automatically constructed from prior student data. Results showed that the tutor is effective in helping student learn, and after including proactive feedback its performance moved closer to that achieved by human tutors. Muir et al. (Muir and Conati 2012) investigated how students react when the hints are provided unsolicited during interaction with an educational game, showing that the *way* hints are given sometimes leads students to ignore them. In another study (Maniktala et al. 2020a), authors developed a hint interface called assertions for a logic tutor and found that the interface significantly improved unsolicited hint usage for all students. When and whether to provide support to students is a well-known challenge. The HelpNeed model discussed in section 1 (Maniktala and Barnes 2020) (Maniktala et al. 2020b) aimed to predict *when* students need help learning efficient strategies. More details about this model are discussed in section 3. Another study (Cody 2020) investigated the effects of unsolicited hints where hints were periodically added to the student's workspace without any element of student choice. Result showed that unsolicited hints can effectively ensure that more help is delivered when it is needed, called possible help appropriateness.

Problem-Solving Subgoal Hints

There are a number of studies that tried to use problem-solving subgoals for improving student learning and performance. One study (Margulieux and Catrambone 2014) found that subgoal labeled instructional text when paired with subgoal labeled examples can improve performance in a computer-based learning environment. Margulieux (Margulieux, Catrambone, and Guzdial 2016) applied subgoal labeled worked examples to explore whether it would improve programming instruction and found that the intervention improves undergraduate learners' problem-solving performance and affects how learners approach problem-solving. In another study (Margulieux, Morrison, and Decker 2020), the authors investigate how subgoal labeled instructions affect problem-solving performance throughout a semester-long programming class. They found that the group who learned with subgoals performed better than the group who learned with conventional instructional materials on quizzes within a week of learning new problem-solving procedures. One study (Harley et al. 2017) found that when subgoals were set collaboratively between learners and the pedagogical agent, they generally lead to higher proportional learning gains when compared to less collaboratively set goals. In a study from Chen et al. (Chen and Catrambone 2019), participants tried to learn to perform task in two domains: cryptarithmetic and Rubik's Cube. Results from the study suggest that subgoal labels increased learning for the Rubik's Cube domain, and they sometimes increased workload in cryptarithmetic domain. Morrison et al. (Morrison et al. 2016) conducted a study where students were asked to solve a Parson's problem [3] and it was found that students who were given subgoal labels performed significantly better than the groups that did not receive subgoal labels or were asked to generate subgoal labels. In another study Margulieux et al. (Margulieux, Guzdial, and Catrambone 2012) utilized subgoal-labeled instructional materials to promote the creation of mental models when teaching novices to program. Results showed that subgoal-labeled materials help novices learn subgoals which reduces the extraneous cognitive load and helps in developing mental models early in the learning process.

However, to the best of our knowledge, our study is the first attempt to both predict when students need help and provide proactive subgoal hints based on that.

Context

In this section, the tutoring system used for the experiment, and the HelpNeed and Waypoint models are discussed. In this study, we use the HelpNeed model and Waypoint hints supported by data-driven next-step hints available on-demand so that students have support in what subgoals to derive and how to derive them.

Tutor Structure

The data is collected from a data-driven intelligent logic tutor where students practice constructing formal propositional logic proofs for a Discrete Mathematics course. The tutor user interface is displayed in figure 1. The interface contains the workspace on the left, and the rules available on the right. There is a 'Get Hint' button (bottom-left) to ask for on-demand hints. For each problem, there is a set of given logic statements presented as premises, and a conclusion to derive. Students go through four tutor sections: introduction, pretest, training, and posttest. The introduction section presents two worked examples to familiarize students with the tutor interface. Next, a *pretest* section, comprising two easy and short problems, is presented to students. Next, students go through the *training* section comprising of five levels. Each level teaches students new rules with increasing difficulty. After completing three training problems per

[3]Parsons problems is a type of assessment in which correct code is broken into code fragments that have to be put in the correct order.

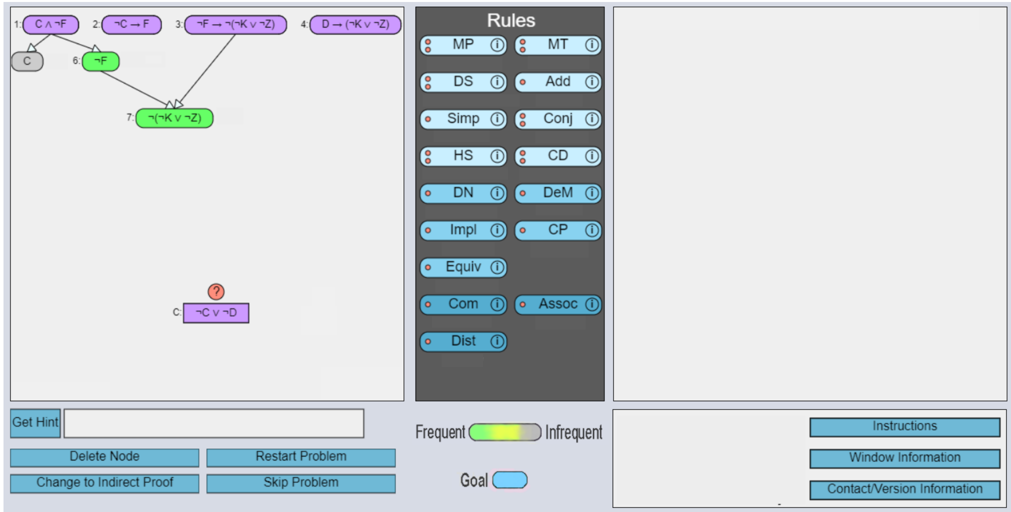

Figure 1: Tutor user interface

level, students are shown a *level end posttest* that assesses students on the rules taught during the level. Finally, students go through a more difficult *final posttest* section comprising six problems. The students get the proactive and on-demand hints only during the training section excluding the level end posttest part. No hints are provided during introduction, pretest, and final posttest section. Note that the tutor is designed to provide immediate feedback on rule application errors in all the sections.

HelpNeed: *When* to Provide Help

The HelpNeed model showed promising results (Maniktala et al. 2020b) by providing a way to determine when to provide proactive hints to students. The model has a HelpNeed observation method which uses a step's duration and efficiency to identify whether the student is possibly using sub-optimal strategies in that step, where step efficiency is a measure of how much a student's most recent step contributes to an efficient (short) solution. It extends the hint factory model (Stamper et al. 2008) which used historical student solutions to form Markov decision processes (MDP). Problem solving state is the snapshot of a solution at any given time, and steps are defined as transitions between states. The helpNeed model leverages the Hint Factory approach to define local and global state quality. Local quality is a measure of how far a state is from the closest goal state. Global quality is a measure of state quality which gives higher values to states on efficient solution paths. The HelpNeed model also defines relative and absolute progress which are two measures of progress. Relative progress is the change in state quality from the previous problem-state, and absolute progress is the change in state quality from the start state. A step is considered efficient if the progress of its post-state using any of the quality measure is non-negative, and inefficient otherwise. The step efficiency and step duration is used by the HelpNeed observation method to identify unproductive steps post hoc. There is a separate HelpNeed predictor to identify unproductive steps at their start based on previous steps so that proactive hints can be given. HelpNeed predictor uses two types of classifiers: state-based, and state-free. A state-based classifier is used when a student's problem-solving state can be matched to historical data and state-free classifier is used in the other cases.

Waypoint: *What* to Provide as Help

When a student on a step is identified as needing help, high-level hints representing problem-solving subgoals are provided at the start of the step. The hints usually require 2-3 steps to derive them. The hints are generated using a modification of the Hint Factory to select a target statement that is 2-3 steps away from the current state. Logic proofs are usually 5-12 steps long, therefore to create subgoal hints it is hypothesized that target statement 2-3 steps away should be a good fit. Among states that are 2 or 3 steps away, the state with a higher use-frequency within prior correct solutions in the historical dataset is selected.

Methods

In this section we discuss the pedagogical policy, study design, performance measures and other details of the analysis.

Pedagogical Policy and Study Design

The proposed pedagogical policy combines two data-driven techniques- HelpNeed model and Waypoint subgoal hints. In each step of the tutor, the HelpNeed model makes a prediction on whether students need help in the current step. Based on the prediction of the HelpNeed model on whether students need help, students are provided proactive problem solving subgoal hints on that step.

The HelpNeed model with next-step hints was found to improve student performance compared to a condition which did not receive any tutor intervention. Now we want to evaluate whether the HelpNeed model with subgoal hints can deliver a good performance. Hence, our study setup was similar to the HelpNeed study. To understand the impact of the proposed pedagogical policy, we conducted a controlled study where the *Adaptive* condition received subgoal hints upon predictions of HelpNeed, while the *Control* condition worked without proactive hints. Students in both conditions could request next-step hints on-demand during training.

Hypotheses

We have the following two hypotheses:

- **H1 - Posttest Performance**: The Adaptive condition will have better performance with shorter (more optimal) solution lengths, and less time than the Control.
- **H2 - Training Productivity**: The Adaptive condition will exhibit better training behavior than the Control, having (a) fewer training HelpNeed steps, and (b) lower possible help avoidance, and higher possible help appropriateness.

To evaluate our hypothesis we will compare the student performance by using the metrics described in the performance metrics subsection. In addition, we will evaluate the performance of high and low prior proficiency groups separately to better understand the impact of proactive problem-solving subgoals on student performance. We will examine whether there is any interaction between the conditions (Adaptive and Control) and the prior proficiency (high and low). In order to get insight about whether there is any relation between proactive hint justification and student performance, we will conduct correlation analysis. To examine the training behavior we will inspect the HelpNeed steps as well as help avoidance and help appropriateness of students in the two conditions. The definition of these metrics can be found in the performance metrics subsection. In addition to the above mentioned analyses, We will look into the hint usage of the students in both the conditions to see if the students are actually taking notice of the hints given and using them in their solution.

Participants

The tutor was given as a homework assignment to 149 undergraduate students in a Discrete Math course at a large university in the USA in Fall 2020. The participants were partitioned in the two conditions using random sampling. 140 participants (80 in Adaptive and 60 in Control)[4] completed the tutor as shown in Table 1. Two Adaptive group participant's data were removed because of log errors.

Performance Metrics

We measure performance using solution length and problem-solving time. In open-ended well-structured[5] domains such as logic, forming shorter (more efficient) proofs, taking less time, and making fewer mistakes reflect more expert-like problem-solving. *Length* of a solution is the number of statements derived in that solution. Length over a tutor section (e.g. posttest) is defined as the sum of lengths of solutions to the problems solved in that section. Next, similar to other studies (Kardan and Conati 2015), we assess students on their problem-solving *time*. To account for times when students may leave the web-based system, we cap each action (any click performed) time to five minutes, and take the sum of the times for each action to determine the total (capped) time. The pretest and posttest scores are calculated using a function of solution length, problem solving time, and accuracy, where accuracy is calculated by taking number of correct rule applications divided by the total number of rule applications.

We use hint justification as the measure of hint usage. The process of hint justification involves selecting existing statements and rules to derive the hinted statement. If a hinted statement is derived by the student using the existing statements and rules, the hint is marked as justified. We measure the *hint justification rate (HJR)* defined as the number of hints given (on-demand or proactive) that are correctly justified divided by total number of hints given. We also investigate student help seeking behaviors such as possible help avoidance, possible help abuse, and possible help appropriateness using the HelpNeed observation and predictor as defined by the HelpNeed study (Maniktala et al. 2020b). Note that the prefix *possible* is added to these terms because HelpNeed does not represent ground truth as classified by experts but rather a heuristic. *Possible help avoidance* is percentage of total training steps with observed HelpNeed but no hints (requested or received). *Possible help abuse* is defined as percentage of total training steps with hint requests but no prediction or observation of HelpNeed. Finally, *possible help appropriateness* is defined as percentage of total training steps with predicted HelpNeed and hints received.

[4] The sampling was set up to result in a larger sample size for the Adaptive condition to gather more data on how the intelligent policy was carried out.

[5] Well-structured domains have problems with a clear goal, end states, or constraints.

Total students	149
Students who completed the tutor	140
Adaptive condition	78
Control condition	60
High prior proficiency adaptive condition	39
Low prior proficiency adaptive condition	39
High prior proficiency control condition	30
Low prior proficiency control condition	30

Table 1: Number of participants in different groups

Results

In this section we present our experimental results. Specifically, we look at the student performance, help seeking behavior, and hint usage. We use Mann-Whitney test for significance analysis.

Student Performance

We report the performance of the students in the Adaptive and the Control conditions in terms of time and solution

	Mean time (Std Dev) in minutes		Mean solution length (Std Dev)	
	Adaptive	Control	Adaptive	Control
Final posttest	59 (36)	65 (52)	94 (42)	105 (58)
Level end posttest	83 (55)	108 (91)	105 (59)	127 (96)
Training	77 (40)	78 (43)	123 (33)	130 (38)
Total tutor	275 (107)	311 (176)	355 (106)	393 (165)

Table 2: Performance of students in the Adaptive and Control condition

	Mean time (Std Dev) in minutes		Mean solution length (Std Dev)	
High prior proficiency students	Adaptive	Control	Adaptive	Control
Final posttest	53 (34)	50 (32)	94 (39)	90 (41)
Level end posttest	72 (50)	73 (54)	100 (55)	100 (62)
Training	67 (31)	71 (40)	118 (21)	126 (36)
Total tutor	224 (89)	219 (93)	333 (82)	333 (103)
Low prior proficiency students	Adaptive	Control	Adaptive	Control
Final posttest	65 (38)	80 (63)	**95 (46)****	**119 (71)****
Level end posttest	94 (58)	143 (108)	109 (64)	154 (117)
Training	87 (46)	85 (46)	128 (41)	143 (40)
Total tutor	326 (102)	403 (195)	376 (123)	453 (196)

Table 3: Performance of high and low prior proficiency students in the Adaptive and Control condition

length. We look at the time and solution length in the final posttest section, level end posttest, training section, and finally the total tutor (includes the introduction, pretest, training, level end posttest, and final posttest section).

Time: Table 2 displays the average and standard deviation of time (in minutes) taken by students in both Adaptive and Control conditions. We find that the students in the Adaptive condition took less time than the Control condition in different sections of the tutor, however the reported differences were not statistically significant.

Solution length: In this tutor, shorter solutions demonstrate better performance, so we look at the length of solutions in the Adaptive and Control conditions. Table 2 displays the average and standard deviation of solution length in both Adaptive and Control conditions. While none of these differences are statistically significant, the solution length for the total tutor in the Adaptive condition is on average 38 steps less than the Control condition.

Impact of proactive hints on low and high prior proficiency student groups. We aim to analyze the impact of proactive subgoal hints on low and high proficiency pretest group performance separately. The students of the Adaptive and Control condition were separated into low and high incoming proficiency groups based on the pretest performance scores. All students having score higher than or equal to median pretest performance score were put into the high prior proficiency group, and the rest of the students were put into the low prior proficiency group.

Time. Table 3 shows the performance of high and low prior proficiency students. The Adaptive condition students with low prior proficiency spent less time in the different tutor sections, however the differences between Adaptive and Control were not significant. Similarly, there was no significant difference in time for the high proficiency groups when comparing time between conditions.

Solution length. There were no significant differences in the average solution length of high proficiency students in the different tutor sections between Adaptive and Control conditions. However, Table 3 shows that the low prior proficiency students in the Adaptive condition had significantly (p-value = 0.047) shorter solution length in the final posttest section than those in the Control. This means that the Adaptive condition helped low prior proficiency students achieve significantly more efficient proofs in similar time when compared to Control condition.

Interaction analysis: We analyze the interaction between the conditions (Adaptive and Control) and the prior proficiency (High and Low). We perform 2-factor interaction analysis using two-way ANOVA. We find marginal significant interactions between the conditions and prior proficiency for total tutor time (df=1, F=3.683, p=0.057), and for level-end test time (df=1, F=4.058, p=0.046). We observe marginally significant interactions between the conditions and prior proficiency for total tutor steps (df=1, F=2.885, p=0.092) and level-end test steps (df=1, F=2.988, p=0.086). From the analysis, we find that there is some interaction existing between the conditions and prior proficiency. This existence of aptitude-treatment interaction (ATI) can mean that the high proficiency students are less affected by the treatment than the low proficiency student (Cronbach and Snow 1977).

Correlation analysis: We calculate the correlation of proactive hint justification rate in the Adaptive condition

	Adaptive	Adaptive (Low)	Adaptive (High)
Posttest time	-0.02 (p = 0.83)	-0.09 (p = 0.572)	0.086 (p = 0.601)
Training time	-0.10 (p = 0.35)	-0.19 (p = 0.242)	-0.006 (p = 0.97)
Total tutor time	-0.01 (p = 0.99)	-0.215 (p = 0.188)	0.230 (p = 0.158)
Posttest steps	-0.14 (p = 0.194)	-0.201 (p = 0.219)	-0.099 (p = 0.547)
Training steps	**-0.22 (p = 0.04)****	**-0.328 (p = 0.041)****	-0.119 (p = 0.469)
Total tutor steps	-0.14 (p = 0.22)	-0.215 (p = 0.188)	-0.07 (p = 0.665)

Table 4: Correlation of proactive hint justification rate with student performance

with the student performance in different sections of the tutor using Spearman correlation coefficient as shown in Table 4. We found a significant negative correlation between proactive hint justification rate and number of training steps in the Adaptive condition. The same phenomenon is also observed for the lower proficiency student group of the Adaptive condition.

Hint Usage

In this section we look at the hint usage of the students. Table 5 provides the average and standard deviation of the number of hints given and justified by the students in the Adaptive and Control condition, rounded to the nearest integer. In the Adaptive condition students were given both proactive and on-demand hints, while in the Control condition received only on-demand hints. It can be seen that students on average justified 91% of the proactive hints in the Adaptive condition. This indicates that the students indeed took notice of the proactive hints and used them. The students also on average justified 95% of the on-demand hints in the Adaptive condition, and 97% of the on-demand hints in the Control condition.

		Mean hints (Std Dev)	
		Adaptive	Control
Number of Hints given	Proactive	27 (10)	
	On-demand	16 (18)	17 (18)
	Total hints	43 (18)	17 (18)
Number of hints justified	Proactive	24 (9)	
	On-demand	15 (18)	16 (17)
	Total hints	39 (17)	16 (17)
Hint justification rate	Proactive	91%	
	On-demand	95%	97%
	Total hints	92%	

Table 5: Hint usage in Adaptive and Control condition

Help Seeking Behavior of Students

In this section, we inspect and compare the help seeking behavior of students in the training section between the Adaptive and the Control condition. Table 6 shows the average and standard deviation of number of steps that were predicted to need help (HelpNeed step) by the HelpNeed prediction model in the Adaptive and Control condition. Although in the Adaptive condition there are fewer training HelpNeed steps on average per student compared to the Control condition, this difference is not statistically significant.

	Mean # of training steps (Std dev)	
	Adaptive	Control
HelpNeed steps	21 (19)	26 (26)
Other	102 (20)	104 (16)
Total	123 (33)	130 (38)

Table 6: Distribution of students' training steps in the two conditions

Help Avoidance and Appropriateness: Figure 2 shows a comparison of help seeking behavior of students in the Adaptive and the Control condition. These behaviors have been defined in the performance metrics subsection of section . We find that the Adaptive condition has 3.59% help

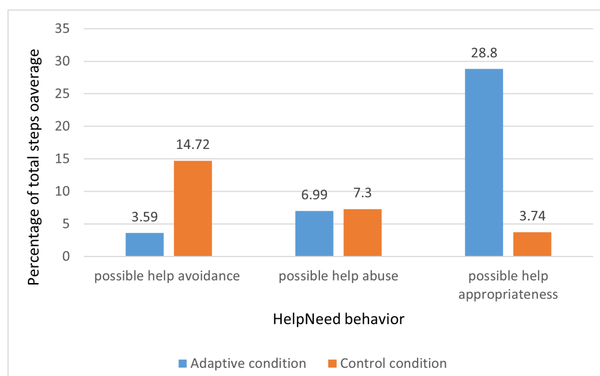

Figure 2: Comparison of possible help avoidance, abuse, and appropriateness between Adaptive and Control condition

avoidance on average (Std Dev = 3.19%) while the Control condition has 14.59% help avoidance on average (Std Dev = 11.01%). From comparison of help avoidance in the Adaptive and Control condition, they were found to be significantly different ($p - value < 0.01$). For help abuse, we see that both the conditions are close in this measure with Adaptive condition having 6.99% (Std Dev = 11.51%), and Control condition having 7.3% (Std Dev = 9.99%). Next for help appropriateness, we find that the Adaptive condition has a high value of 28.8% (Std Dev = 9.7%) and Control condition has 3.74% (Std Dev = 4.66%). A significant difference was found in the help appropriateness in the Adaptive and Control condition ($p - value < 0.01$).

Discussion

Here we investigate our two hypotheses about the Adaptive condition: H1 on improved posttest performance and H2 on (a) reduced HelpNeed steps and (b) reduced possible help avoidance and increased possible help appropriateness during training.

H1: Student Performance

From comparing the posttest performance of students in the Adaptive and the Control condition, students in the Adaptive condition took less time and fewer steps compared to Control condition, however the difference in time and steps were not significant compared to the Control condition. Next, we compared the performance of high and low proficiency students of the Adaptive and Control condition. There were no significant differences in time or steps between conditions for those with high prior proficiency. However, for the low proficiency group, the Adaptive condition outperformed the Control overall, and had significantly fewer steps in their posttest. This group is of particular interest for us, because when starting the tutor this group had lower proficiency based on pretest performance score and was already lagging behind the other group of students. Therefore, these students may need more help, and in this case the proactive hints are filling in this gap and providing help to those who need it more. Also from literature review we find that students with low prior proficiency get more benefit from interventions (Kardan and Conati 2015). Since the significantly better results were for lower proficiency group posttest step analysis, it suggests that H1 is partially correct.

H2: Help Seeking Behavior of Students

From the comparison of HelpNeed steps we find that while the number of HelpNeed steps were less in Adaptive condition compared to Control condition, it is not significant. Therefore H2a could not be confirmed. From the comparison of possible help avoidance which is when HelpNeed was observed but no help was requested or received, we find that Adaptive condition has significantly lower help avoidance compared to Control condition. This suggests that when students were being unproductive, they either got a proactive hint or requested an on-demand hint in most cases, therefore the help avoidance value was lower in the Adaptive condition. Next we analyze the help appropriateness values of the Adaptive and Control condition. From section we find that the Adaptive condition has significantly higher help appropriateness compared to the Control condition. This indicates that when the predictor predicted that help was needed a hint was either requested or given proactively. These results suggest that students were more likely to get help when they most needed it in Adaptive condition compared to the Control condition. Overall the Adaptive condition has significantly reduced possible help avoidance and increased possible help appropriateness compared to the Control condition. This is strong evidence suggesting that hypothesis H2b is correct.

Assistance dilemma is a common problem over intelligent tutors and in particular open-ended intelligent tutors. We believe that the findings of this study should be generalizable to other ITSs, and the pedagogical policy of predicting when students need help and providing proactive subgoal hints based on that should be useful in addressing assistance dilemma in other ITSs.

Limitation and Future Work

One limitation of the study is that it was conducted using one ITS, so we could not validate the generalizability of the pedagogical policy. Therefore, in the future, we aim to use the pedagogical policy with other ITSs to investigate its generalizability.

The pedagogical policy utilized historical student solutions to generate hints. In the future, we plan to explore automatic hint creation for logic proofs (Lodder et al. 2021). Furthermore, we plan to build a policy that can differentiate between when students are ready to benefit from subgoal hints and when they need next-step hints.

In the future, we aim to take information such as the demographic of the participants into account while conducting study. We also plan to survey the students to evaluate the pedagogical policy from their viewpoint.

Conclusions

In this paper, we presented a new pedagogical policy combining two data-driven components to address the assistance dilemma in an intelligent logic tutor. We conducted a study to investigate the effectiveness of the pedagogical policy against a policy without these components. The study demonstrated that the new policy improved posttest performance for students who need it the most - those with low prior proficiency, and reduced help avoidance in training for all students. There was empirical finding of negative correlation between proactive hint justification rate and training steps of students, suggesting that students who achieved the subgoals are learning how to form more efficient solutions. Overall the findings provide insight into the impact of providing proactive subgoal hints when students need help in an intelligent tutor.

Acknowledgements

This research was supported by the NSF Grants: Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments(1726550) and Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502).

References

Aleven, V.; Mclaren, B.; Roll, I.; and 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.

Catrambone, R. 1998. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of experimental psychology: General*, 127(4): 355.

Chen, D.-W.; and Catrambone, R. 2019. Productive Failure and Subgoal Scaffolding in Novel Domains. In *International Conference on Human-Computer Interaction*, 282–300. Springer.

Cody, C. 2020. Does autonomy help Help? The impact of unsolicited hints and choice on help avoidance and learning. In *Proceedings of the 13th International Conference on Educational Data Mining*.

Cody, C.; Maniktala, M.; Lytle, N.; Chi, M.; and Barnes, T. 2022. The impact of looking further ahead: a comparison of two data-driven unsolicited hint types on performance in an intelligent data-driven logic tutor. *International Journal of Artificial Intelligence in Education*, 32(2): 263–296.

Cronbach, L. J.; and Snow, R. E. 1977. *Aptitudes and instructional methods: A handbook for research on interactions.* Irvington.

Fossati, D.; Eugenio, B. D.; Ohlsson, S.; Brown, C.; and Chen, L. 2010. Generating proactive feedback to help students stay on track. In *International conference on intelligent tutoring systems*, 315–317. Springer.

Harley, J. M.; Taub, M.; Azevedo, R.; and Bouchet, F. 2017. Let's set up some subgoals: Understanding human-pedagogical agent collaborations and their implications for learning and prompt and feedback compliance. *IEEE Transactions on Learning Technologies*, 11(1): 54–66.

Kardan, S.; and Conati, C. 2015. Providing adaptive support in an interactive simulation for learning: An experimental evaluation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 3671–3680.

Koedinger, K. R.; and Aleven, V. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3): 239–264.

Lodder, J.; Heeren, B.; Jeuring, J.; and Neijenhuis, W. 2021. Generation and Use of Hints and Feedback in a Hilbert-Style Axiomatic Proof Tutor. *International Journal of Artificial Intelligence in Education*, 31(1): 99–133.

Maniktala, M.; and Barnes, T. 2020. Extending the hint factory: Towards modelling productivity for open-ended problem-solving. In *In proceedings of the 13th International Conference on Educational Data Mining (DC paper)*.

Maniktala, M.; Cody, C.; Barnes, T.; and Chi, M. 2020a. Avoiding help avoidance: Using interface design changes to promote unsolicited hint usage in an intelligent tutor. *International Journal of Artificial Intelligence in Education*, 30(4): 637–667.

Maniktala, M.; Cody, C.; Isvik, A.; Lytle, N.; Chi, M.; and Barnes, T. 2020b. Extending the Hint Factory for the assistance dilemma: A novel, data-driven HelpNeed Predictor for proactive problem-solving help. *arXiv preprint arXiv:2010.04124*.

Margulieux, L. E.; and Catrambone, R. 2014. Improving problem solving performance in computer-based learning environments through subgoal labels. In *Proceedings of the first ACM conference on Learning@ scale conference*, 149–150.

Margulieux, L. E.; Catrambone, R.; and Guzdial, M. 2016. Employing subgoals in computer programming education. *Computer Science Education*, 26(1): 44–67.

Margulieux, L. E.; Guzdial, M.; and Catrambone, R. 2012. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the ninth annual international conference on International computing education research*, 71–78.

Margulieux, L. E.; Morrison, B. B.; and Decker, A. 2020. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7(1): 1–16.

Morrison, B. B.; Margulieux, L. E.; Ericson, B.; and Guzdial, M. 2016. Subgoals help students solve Parsons problems. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 42–47.

Muir, M.; and Conati, C. 2012. An analysis of attention to student–adaptive hints in an educational game. In *International conference on intelligent tutoring systems*, 112–122. Springer.

Roll, I.; Aleven, V.; McLaren, B. M.; Ryu, E.; d Baker, R. S.; and Koedinger, K. R. 2006. The help tutor: Does metacognitive feedback improve students' help-seeking actions, skills and learning? In *International conference on intelligent tutoring systems*, 360–369. Springer.

Stamper, J.; Barnes, T.; Lehmann, L.; and Croy, M. 2008. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, 71–78.