Educational Robotics and the Individual Tutorial: Exploring the Effects of Personalization on Attention and Help-Seeking Behavior

Kevin Jiang
Computer Science
Yale University
51 Prospect St
New Haven, CT 06520
k.jiang@yale.edu

ABSTRACT
The individual tutorial, consisting of a one-on-one interaction between student and teacher, remains the gold standard of education, as tutored students achieve learning gains two standard deviations above those of their non-tutored peers. Robotic tutoring systems provide an opportunity to emulate these learning benefits in a highly scalable manner. This paper examines the effects of (1) shaping productive help-seeking behavior and (2) providing breaks in the context of an individual tutorial led by a physically present robot. First, a preliminary investigation for a personalized system of shaping help-seeking behaviors is proposed. Then, an implementation of a full robot tutoring system is described. This platform will be used to perform a break timing study, which aims to examine the efficacy of two different personalized break strategies in terms of student learning outcomes, perceptions of the robot tutor, and overall engagement.

1. INTRODUCTION
The advantages of one-on-one tutoring when compared to standard classroom instruction in student academic performance has been well documented in education research— in general, students perform one to two standard deviations better when receiving one-on-one tutoring [2]. The benefits of one-on-one tutoring are not only empirically supported but seem intuitively salient. It is easy to cognize how the individual tutorial setting— which offers a space where a tutor can highly individualize their pedagogical style based on a particular student, provide immediate feedback, corrections, and reinforcement for particular actions, and supply quality interaction with a student—can lead to disproportionately high learning gains when compared to a conventional classroom setting [3]. However, it is financially unrealistic for most schools to provide this kind of personal instruction on a consistent basis for all or even some of its students. Initial investigative research in the field of robotics has been conducted on ways to provide a more scalable education experience that produces similarly impressive student achievement results. The ultimate goal in this field of research is to be able to provide a personalized, individual tutoring experience for students without the need for extensive human capital in the form of skilled tutors and teachers.

Two particular aspects of the individual tutorial will be examined in this investigation: the effects of shaping help-seeking behaviors and break timing conditions.

1.1 Help-Seeking Introduction
Emulating the efficacy of expert human tutors on a holistic level is a complex problem with many different aspects worth pursuing and considering. There exists extensive research in intelligent tutoring systems (ITS) concerning cognitive learning gains. One feature of many intelligent tutoring systems is that they offer on-demand help services where, theoretically, a student who is struggling with material can prompt the system for help [1]. However, this kind of on-demand help is not congruent with the actions and best practices of expert human tutors, who (1) often withhold help when they think it is necessary and (2) do not provide hints at students’ every whim [3].

Given that up to 72% of help requests are unproductive in on-demand tutoring, shaping a student’s help-seeking behavior shows great promise as an aspect of this congruency problem to pursue [7]. While there are a variety of approaches to building an intelligent tutoring system, research has indicated that (1) the physical presence of a robot tutor can be effective in increasing learning gains [4] and (2) a robot tutor utilizing simple personalization strategies can have a significant positive impact on student achievement[7].

Further discussion of this help-seeking research and the preliminary research study proposed will be described later in this investigation.

1.2 Break Timing Introduction
A second aspect of the expert tutoring problem involves the challenge of maintaining a student’s attention throughout the tutoring interaction. Students, especially younger students, can have an attention span in traditional human-teacher-led learning settings that spans as short as five min-
utes [6]. Thus, one would expect that, once the novelty of a robot interaction wears off, a student-robot learning setting might suffer from similar or even worse attention deficit scenarios.

Expert tutors employ various strategies to combat the inattention of students, but many of these strategies—such as providing nuanced feedback, allowing students to actively reflect on their experience, and contextualizing material [3]—are currently infeasible to generalize into an easily actionable model for robot tutoring systems. However, one method of accommodating for short attention spans in student-robot interactions that has not been well-documented in social robotics learning situations is to introduce short breaks into a tutoring session. A break is considered any activity that is unrelated to the task at hand in the learning setting interaction. Breaks have been empirically shown to improve performance on various tasks—for instance, one, five, or ten minutes disruptions were shown to have positive performance effects on human performance on auditory tasks [5].

Further, robot behaviors personalized to particular student interactions have been shown to be more effective in shaping student behavior than simple, fixed behaviors [4]. Thus, we hypothesize that a break setting contingent upon student actions that is introduced in a student-robot learning interaction could positively effect a student’s learning outcomes, perceptions of the robot, and overall user experience.

The use of breaks during a learning interaction as a tool for motivation and increased engagement is not yet well documented in the literature. In a traditional setting, expert tutors may use breaks as either a positive activity to reward good or improved performance or as a tool to combat student frustration, encourage the student, or simply refresh the student’s affective state. The study proposed in section 3 seeks to compare these two different break trigger conditions to provide a better understanding of which kinds of break-triggering mechanisms are most effective in a student-robot learning interaction.

2. HELP-SEEKING BEHAVIOR RESEARCH

The goal of this portion of the investigation is to lay the groundwork for a robot tutoring system that utilizes (1) data collected from previous research studies and (2) real-time data from student-robot interactions to autonomously provide effective help for students.  

2.1 Investigation and Literature Review

Initial research was conducted to gain motivation for implementing effective help-seeking techniques.

2.1.1 Expert Tutors

Many of the methods of effective human tutors seem, at first, to be counter-intuitive; however, in practice, these techniques were common to many if not all the practices of expert tutors. In particular, effective human tutors seemed to employ a decidedly Socratic approach in instructing the students as opposed to a more didactic approach. The Socratic approach meant that human tutors would constantly use questions, as opposed to directions or assertions, when working with tutees, which empirically resulted in more than 90% of remarks made in a tutoring session to be in the form of questions.

Additionally, tutors almost never offered hints that directly led to answers. Instead, they preferred hints that necessitated the student to make the next step on their own and often persisted in offering five or six such hints in succession, even if their initial efforts were unsuccessful, in an attempt to lead students to the right answer.

Finally, effective human tutors were able to distinguish between productive errors (i.e. errors that would provide opportunities for students to learn more) and nonproductive errors (i.e. errors that could only be corrected by explicit intervention and were leading students down a dysfunctional path) and act accordingly. Effective tutors would often let productive errors pass while quickly intervening when nonproductive errors occurred [9].

While these expert tutor findings seem promising and salient, many of the specifics are difficult to implement in a robot-student interaction due to platform limitations and social complexity. However, there are some important lessons to learn from this investigation. In particular, it is important to emphasize that a student’s affective state matters just as much, if not more, as their cognitive understanding of the subject material in a tutoring interaction. Much of ITS research has been conducted with the explicit or implicit goal of correcting cognitive errors. In subsequent sections, help-seeking behaviors, as well as break timing conditions, will be examined as potential solutions to shaping both the cognitive and affective state of students.

2.1.2 Simple Help-Seeking Behavior Shaping

Shaping help seeking behaviors in a simple manner has been shown to produce significant learning gains as well as decrease maladaptive help-seeking behavior over subsequent sessions. In a study involving fifth and sixth graders learning fractions problems, researchers found that students who received adaptive strategies targeting suboptimal help requests (i.e. asking for too much help or not asking for enough help) not only reduced their suboptimal behavior over time but also significantly improved their test performance when compared to a control group [7]. The magnitude of positive performance change derived from these simple personalization strategies was very encouraging and prompted this follow-up investigation to examine more in-depth personalization strategies to improve student learning gains and help-seeking behavior.

A preliminary proposal for a system involving more complex and personalized behavior shaping properties is detailed in the subsequent section. The hope is that the future implementation of such a system could result in even greater learning gains for students in robot-student tutoring interactions.

2.2 Proposed Framework

A preliminary framework to shape help-seeking behavior could potentially take as input two main sources of information that can be analyzed in real-time from the robot-tutor interaction. One source is the video feed of the interaction that can be used in conjunction with image processing soft-
will allow for a more personalized tutoring interaction with a student explicitly asking for help. Furthermore, this approach hints when the student is stuck, oftentimes without the student where a tutor autonomously guides the student with a holistic view. This type of help-providing behavior is meant incorporating historical student data pertaining specifically gathered from the subsequent tutoring interaction while (3) fine-tune this strategy with real-time information strategy based on (1) an initial evaluation of the student and (2) is active at moderate values of $h$ while (3) is active at moderate values of $h$ and (4) is active at high values of $h$. In general, low values of $h$ correspond to a state where the student should be able to solve the problem without additional help, whereas high values of $h$ indicate that a student needs additional help. Accordingly, the robot will act to shape the student’s help-seeking behavior more effectively. During a given question in a tutoring interaction, there are four distinct ranges where $h$ can lie. The ranges are labeled as follows: (1) block hint, (2) block answer, (3) suggest hint, and (4) force hint. (1) and (2) are active at low values of $h$ while (3) is active at moderate values of $h$ and (4) is active at high values of $h$. During a given question in a tutoring interaction, there are four distinct ranges where $h$ can lie. The ranges are labeled as follows: (1) block hint, (2) block answer, (3) suggest hint, and (4) force hint. (1) and (2) are active at low values of $h$ while (3) is active at moderate values of $h$ and (4) is active at high values of $h$. In general, low values of $h$ correspond to a state where the student should be able to solve the problem without additional help, whereas high values of $h$ indicate that a student needs additional help. Accordingly, the robot will act to shape the student’s help-seeking behavior more effectively.

- $w$: number of wrong attempts on current question. As this value increases, $h$ also increases.
- $b$: number of hint attempts blocked. As this value increases, $h$ also increases.
- $t$: elapsed time. As this value increases, $h$ also increases. Additionally, if $t$ is less than the previous time used for similar questions, $h$ is decreased by a constant amount; otherwise, $h$ is increased.
- $hl$: current hint level. If $hl$ is less than the previous number of hints used for similar questions, $h$ is decreased by a constant amount. Otherwise, $h$ is increased by a constant amount.

### 2.3 Proposed Experimental Setup

The system will attempt first to construct a help-providing strategy based on (1) an initial evaluation of the student and then (2) fine-tune this strategy with real-time information gathered from the subsequent tutoring interaction while (3) incorporating historical student data pertaining specifically to the particular student being tutored as well as all students holistically. This type of help-providing behavior is meant to simulate the traditional human tutor and student interaction where a tutor autonomously guides the student with hints when the student is stuck, oftentimes without the student explicitly asking for help. Furthermore, this approach will allow for a more personalized tutoring interaction with students, which should help promote academic gains.

For this framework proposal, the Nao humanoid robot will be used to interact verbally and visually with students. In addition, an Android tablet application will be used as a method to display and check answers to problems given in the tutoring sessions. The framework, while meant to be generalizable, is intended to tutor fifth and sixth graders in math fraction problems as part of a study.

Students in the study would be expected to return for a few tutoring sessions each in order to track a student’s learning gains as well as track a student’s help-seeking behavior over time.

Due to various time constraints and lab resource limitations, work regarding the personalized help-seeking study was halted in favor of implementing the infrastructure necessary to conduct an upcoming break study in the summer of 2016. However, much of the infrastructure built to implement the break taking research study that will be subsequently discussed can also be applied to further the development of this help-seeking research study in the future.

### 3. Break Timing Research

#### 3.1 Purpose and Goals

Utilizing off-task breaks in the middle of an individual tutoring session has great potential to improve student-robot interactions, especially with younger students who exhibit trouble maintaining attention over an extended period of time. Breaks could be used as a (1) reward mechanism to encourage good performance or learning behaviors or a (2) frustration alleviation mechanism designed to refocus a disengaged student to the task at hand. The aim of this study is to understand whether these two types of personalized break strategies are more effective than breaks given at fixed time intervals when examining learning outcomes, student perceptions of the robot tutor, and overall student engagement.

#### 3.2 Experimental Setup

Similar to the aforementioned help-seeking study [7], this study will also primarily involve visual and verbal student interaction with the robot Nao in a one-on-one setting. To forgo issues with speech recognition and difficulties with effectively verbally relaying mathematics problems, an Android tablet will be used to facilitate communication between Nao and the student. The bulk of the tutoring sessions will involve Nao giving the student order of operations math problems through the tablet interface and providing feedback based on the student’s responses.

Additionally, four potential breaks will be interspersed throughout the interaction. The timing of these breaks will depend on the particular break strategy being implemented at the time.

#### 3.3 Platform Summary

There are three main platforms being used in concurrence for this experiment: the Nao robot, the Android tablet application, and the Python server backend. The Nao robot executes all social interactions with the student while the Android application serves breaks, displays questions, and allows for the student to respond to Nao’s prompts. The
Python server handles much of the internal data processing including the decision of whether or not to serve a break. The server communicates in real-time with both Nao and the Android application for the duration of the study.

3.3.1 Android Tablet Application
An Android tablet application was constructed to help facilitate communication between the student and the Nao robot. In particular, the tablet serves the text for each question, allows for student input for each answer, and facilitates certain tablet-specific break interactions (such as the tic-tac-toe break). The tablet also gathers timing data from student interactions for later processing on the server to determine break timing. In general, the tablet application is in constant communication with the Python server through a TCP Client connection.

3.3.2 Nao Robot
The Nao robot is an autonomous, programmable humanoid robot that can speak, move, and maintain eye contact throughout a human interaction.

The Nao robot is given instructions from the Python server to perform certain actions including saying question prompts, giving verbal and physical (i.e. celebratory actions) question feedback, and facilitating breaks both physically (i.e. in the stretch break) and verbally. The Nao robot’s actions are kept in sync with the tablet application through communication with the Python server.

3.3.3 Python Server
The python server implements much of the back-end logic including the decision of whether or not to take a break at a given time. Additionally, the server logs all important interaction information from each student interaction and coordinates actions between Nao and the Android tablet (for example, an Android tablet’s buttons will be disabled while Nao is speaking).

3.4 Tutoring Interaction Flow
For a more concrete understanding of the student-robot tutoring interaction flow, a big-picture algorithm is detailed.

Algorithm 1 Tutoring Interaction Flow
Student-Nao interaction begins. Nao provides the student an introduction to order of operations rules.

for every q in the question bank do
Nao serves q through tablet interface and awaits student feedback.

Nao receives one of three possible kinds of student feedback: correct answer, incorrect answer, or question timeout.

Nao evaluates the feedback in congruence with student behavior history and break strategy.

if break triggered then
Nao serves a break before the next question
end if
end for

Tutoring interaction ends.

3.5 Break Strategies
To effectively test the efficacy of different break delivering strategies, three different break strategies will be tested in this study. They will be described in the following sections.

3.5.1 Fixed Condition
The fixed break strategy is the control condition and is fairly straightforward- breaks will be given at fixed question-intervals throughout the tutoring interaction. In total, four breaks will be interspersed after 20%-40%-60%-80% of the questions are completed to ensure that a constant number of questions will be given in between each served break.

3.5.2 Positive/Reward Condition
The reward break strategy seeks to reward students who are either consistently performing well or are improving in their performance. The basis of this strategy hinges on two measured quantities: the accuracy of the student in answering questions correctly and the time it takes for the student to answer each question. A window of history for a student’s most recent responses, the length of which is defined as a constant wi across all experiments, is used to represent historical student data.

First, the reward break strategy will consider any changes in accuracy. Subsequently, the strategy will consider any changes in speed. Based on the results of these conditionals, in concurrence with a set of super rules that will be discussed later, a decision as to whether or not to serve a break will be reached by the server and communicated to the the tablet/Nao.

Algorithm 2 Positive/Reward Condition

if accuracy (Δa) increases

if completion time (Δs) faster or no change then
reward break: overcomes struggles, improves
else
reward break: improves while taking their time
end if
else if accuracy (Δa) has no change
if overall accuracy > 80% then
if completion time (Δs) faster then
reward break: becoming faster and more confident
else if completion time (Δs) has no change
if consistency condition (t) met then
reward break: doing consistently well
end if
end if
end if

3.5.3 Negative/Frustration Condition
The frustration break strategy seeks to refocus students who are performing worse as time goes on. Similarly to the reward condition, the basis of this strategy depends mainly on
the accuracy of the student’s answers to the question and the time it takes for the student to answer each question. A window of history $wi$ of a student’s most recent responses is used to represent past data.

First, the frustration break strategy will consider any changes in question accuracy. Then, the strategy will consider any changes in question answering time. Based on the results of these conditionals along with consideration of a set of super rules to be discussed below, a decision as to whether or not to serve a break will be reached.

Algorithm 3 Negative/Frustration Condition

```
if accuracy ($\Delta a$) decreases then
    if completion time ($\Delta s$) faster or no change then
        frustration break: guessing, making mistakes
    else
        frustration break: performance drop
    end if
else if accuracy ($\Delta a$) has no change then
    if overall accuracy > 80% then
        if completion time ($\Delta s$) slower then
            frustration break: bored, distracted, disengaged
        end if
    else if overall accuracy < 80% then
        if completion time ($\Delta s$) slower then
            frustration break: disengaged
        else if completion time ($\Delta s$) faster then
            frustration break: bored, distracted
        else
            if consistency condition ($t$) met then
                frustration break: doing consistently poorly, frustrated
            end if
        end if
    end if
end if
```

Accompanying each frustration break is a short explanation of why the break was given as a refocusing mechanism. Depending on the kind of explanation for a reward, the Nao robot will introduce the break to the student in a different manner, in accordance with their projected affective state.

3.5.4 Super Rules

There are a few super rules that apply to both the reward and frustration conditions. These rules override any rules that they conflict with. They are constructed in such a manner that they will not conflict with one another.

- **Consistency Rule**: A predetermined value $t$ is determined to trigger a consistency break. In both conditions, if a student achieves $t$ times in a row the consistency condition detailed in the algorithms, then they are rewarded with a break.

- **Consecutive Rule**: No break will be served if less than $c$ questions answered before the last break. In other words, there needs to be at least $c$ consecutive questions answered before an additional break will be served.

- **Minimum Breaks Rule**: Require a minimum of two breaks per interaction. Thus, if no breaks have been triggered, serve an automatic break after 33% of the questions have been served. If only one break has been triggered, then additionally serve an automatic break after 66% of the questions have been served.

3.5.5 Break Parameters

The specific parameters given in the Reward and Frustration conditions will ultimately affect the overall behavior of the tutoring system. Values are chosen such that Nao is neither too aggressively giving breaks for minor positive/negative changes nor too conservatively withholding breaks for major positive/negative changes. The following values are a preliminary attempt at tuning this system and may be modified upon further testing before deployment in the actual study.

Table 1: Preliminary Break Condition Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$: consistency parameter</td>
<td>4 questions</td>
</tr>
<tr>
<td>$c$: consecutive parameter</td>
<td>4 questions</td>
</tr>
<tr>
<td>$wi$: window size</td>
<td>5 questions</td>
</tr>
<tr>
<td>$\Delta a$: accuracy change threshold</td>
<td>20%</td>
</tr>
<tr>
<td>$\Delta s$: speed change threshold</td>
<td>10 seconds</td>
</tr>
</tbody>
</table>

3.5.6 Break Content Information

Arsalan Sufi worked extensively to develop and implement different kinds of breaks for the robot-student interaction. The two breaks that are currently implemented involve (1) a tic-tac-toe game where the student plays Nao as an adversary and (2) a stretch break where the student follows Nao in an active exercise routine.

Two additional breaks will be developed in the near future to round out the maximum of four types of breaks that can be triggered in one full student interaction. The hope is that these breaks will provide a cognitive context-switch from the mathematics learning task while simultaneously improving the student’s social relationship with the Nao robot tutor.

3.6 Data Models

To keep track of historical data for each student as well as allow for subsequent post-study data analysis, a set of object models were created to keep pertinent session information. These objects are pickled and stored for future retrieval after every question—a this also allows for effective app crash management as the state of the tutoring interaction is essentially saved after every question. The different models are briefly detailed below.

- A question class stores and manipulates features related to each question served, such as whether or not the question was answered correctly, the total time needed to answer the question, and whether or not the question timed out.

- A break class stores and manipulates features related to each break decision made after a question is served. These features include what type of break was triggered, whether or not a break was actually given, and the start time of the break.
A session class stores and manipulates features related to each student’s tutoring session. This class maintains a list of question served, a list of break decisions considered, and timing data.

These data models can easily be extended for future use in related student-robot interactions. In particular, the preliminary help-seeking study outlined previously could leverage these models during implementation. In the specific case of the help-seeking study, additional information would be stored in the question class including in-depth hint request information and question attempt information.

4. CONCLUSION AND FUTURE WORK
This paper discussed investigative research conducted in regards to shaping help-seeking behavior and providing breaks in tutoring interactions. Additionally, a full platform involving an Android application, a Nao robot, and a Python server was implemented to provide a personalized manner to serve breaks in a tutoring setting. This platform will be used in an upcoming study in the summer of 2016 to explore the efficacy of breaks provided as a reward for good performance versus breaks provided as a refresher in light of poor performance and frustration.

There are several other areas of research that may be pursued in the future to further the goal of creating an adaptive, personalized robot tutoring experience. Affect detection could be enhanced by the use of vital sign data (especially pulse rate and body temperature) through the use of heart rate monitors or IR cameras. This vital sign data could be used to complement the interaction information already incorporated in the robot tutoring platform described in this paper. Additionally, further investigation into ways that a robot can effectively emulate the nuanced pedagogical approaches of expert tutors is warranted. For example, expert tutors use techniques that implicitly reinforce a growth mindset (i.e. problem solving ability can be improved through effort) as opposed to a fixed mindset (i.e. problem solving ability is innate). Continued research in improving the pedagogical efficacy of robot tutors could soon produce an effective and scalable commercial robot platform that would drastically improve the current state of education.

5. ACKNOWLEDGMENTS
A special thanks to Aditi Ramachandran and Professor Brian Scassellati of the Yale University Computer Science department for advising me on this project. An additional thanks to Arsalan Sufi for collaborating with me on implementing the Nao-tablet system.

6. REFERENCES