MathBot: Transforming Online Resources for Learning Math into Conversational Interactions

Joshua Grossman*
Stanford University
jgrossman@stanford.edu

Zhiyuan Lin*
Stanford University
zylin@cs.stanford.edu

Hao Sheng*
Stanford University
haosheng@stanford.edu

Johnny Tian-Zheng Wei
University of Massachusetts Amherst
jwei@umass.edu

Joseph Jay Williams
University of Toronto
williams@cs.toronto.edu

Sharad Goel
Stanford University
scgoel@stanford.edu

ABSTRACT

Online math education often lacks key features of in-person instruction, such as personalized feedback. To emulate such interactivity while preserving the scalability of online systems, we developed MathBot, an automated text-based tutor that explains math concepts, provides practice questions, and offers students tailored feedback. We evaluated MathBot through two online studies conducted with Amazon Mechanical Turk in which participants learned about arithmetic sequences. In the first study, we examined user preferences, comparing MathBot with videos and written tutorials from Khan Academy. This within-subject study revealed that 42% of individuals in our sample preferred MathBot over videos, while 47% preferred MathBot over written tutorials. In a second, between-subject randomized study, we found that both MathBot and Khan Academy produced sizable learning gains, with MathBot performing slightly better, though the difference was not statistically significant. Our findings indicate that conversational agents are a promising tool for complementing online math education.

KEYWORDS

Chatbot, conversational agent, online education

1 INTRODUCTION

Math learners can now turn to a wide variety of freely available online resources, from Khan Academy to Massive Open Online Courses (MOOCs). For example, popular videos on Khan Academy explain complex mathematical concepts, and accompanying written tutorials provide practice problems and explanations. However, these resources cannot completely reproduce features of in-person tutoring, like giving students the sense that they are engaged in a back-and-forth exchange with a tutor, tailored feedback, and guidance about how to allocate their attention between reading explanations and practicing problems.

Our paper explores how to adapt existing online educational resources into a format that can mimic some facets of conversation with a human tutor: conversational flow, comprehension checks, and personalized feedback and guidance. To explore this approach, we designed and evaluated a prototype system, MathBot. To achieve conversational flow, we built the system as a chatbot, with all material presented via a simple text-based interface. To better mirror the experience of interacting with a human tutor, we paid close attention to the timing of prompts and incorporated informal language, including the use of emoji. As with a human tutor, the MathBot system alternates between presenting material and gauging comprehension. Comprehension checks range from simple queries (e.g. "Let me know when you're ready to hear more!") to asking students to solve multi-step mathematical problems. Finally, MathBot provides learners with personalized feedback and guidance. Common incorrect answers to problems are identified through rule-based logic and followed up with specific explanations and hints on how to proceed. Further, upon detecting incorrect answers, MathBot presents students with sub-problems to clarify the concept at hand, and reviews past explanations with them.

The goal of MathBot is to explore the possibility of creating conversational experiences *without* needing to support true dialogues that ask and answer open-ended questions. Past work on conversational tutors in education has involved the creation of custom-tailored conversations around questions like "What is the direction of acceleration for keys dropped in an elevator? Why?" [16, 18, 20, 21, 29, 34]. These types of conversational tutors have been shown to benefit student learning [10, 12, 27, 41], but creating such systems can require input from teams of computational linguists, cognitive psychologists, and domain experts. Few of these conversational systems exist for mathematics problem solving in topics like algebra (although see Nye et al. [30] for first steps).

MathBot aims for conversational interaction with an agent by integrating approaches from online math platforms that are not presented in a conversation, but have features like

*These authors contributed equally to the paper; ordered by last names alphabetically.

tailored feedback and personalized guidance. Online math homework like ASSISTments [15, 22] gives feedback on common wrong answers and allows repeated practice. Online resources like MathTutor [2] build on example-tracing tutors [5], which aim to reduce the development time in achieving some of the benefits of intelligent tutoring systems for mathematics [12, 14, 30], like personalized selection of problems and feedback. Example-tracing tutors model the progression of a lesson with a behavior graph that (a) outlines potential student actions, such as providing common incorrect responses, and (b) specifies the feedback, explanation or new problem that should follow those actions. The development process for MathBot is similar, but the key difference is that we explore whether it is possible to combine such approaches with conversation-style interactions.

To evaluate MathBot, we carried out two online studies on Amazon Mechanical Turk, one measuring user preferences and the other measuring learning outcomes. The first study had two distinct parts. In the first part, 116 participants completed an abridged lesson on MathBot and watched a video on Khan Academy covering similar content, and then rated their experiences. 42% of users stated at least a weak preference for MathBot, with 20% indicating a strong preference. We conducted the second part identically to the first, though we replaced the video with a written tutorial from Khan Academy. 47% of 110 participants stated at least a weak preference for MathBot, with 18% indicating a strong preference.

In our second study, we randomized 370 participants to either complete a full-length conversation with MathBot or a set of videos and written tutorials from Khan Academy covering similar content. To gauge learning, each subject took a test of knowledge before and after completing the learning module. Participants assigned to MathBot fared somewhat better than those assigned to Khan Academy, though the difference was not statistically significant.

In summary, our contributions are: (1) A prototype system that adds conversational interaction to learning mathematics by solving problems and receiving explanations. (2) Qualitative and quantitative data about users' perceptions and learning outcomes after using MathBot and Khan Academy videos and written tutorials.

2 RELATED WORK

Below we discuss relevant work on conversational tutoring systems, as well as approaches to building example-tracing tutors and other intelligent tutoring systems. Furthermore, we discuss the implementation and user interface design of chatbots.

Conversational Tutors in Education. Conversational tutors in education often build a complex dialogue, such as asking students to write qualitative explanations of concepts (e.g. *A battery is connected to a bulb by two wires. The bulb lights. Why?*) and initiating a discussion based on the responses. AutoTutor and its derivatives [16, 21, 29, 42] arose from Graesser et al. [19]'s investigation of human tutoring behaviors and modeled the common approach of helping students improve their answers by way of a conversation. These systems rely on natural language processing (NLP) techniques, such as regular expressions, templates, semantic composition [42], LSA [21, 32], and other semantic analysis tools [18]. Nye et al. [30] added conversational routines to the online mathematics ITS ALEKS by attaching mini-dialogues to individual problems, but leaving navigation on the website. MathBot aims to have the entire learning experience take place through a text conversation, giving the impression of a single tutor. More broadly, MathBot differs from past work on NLP-based conversational tutors in that it explores the possibility of reproducing part of the conversational experience without handling open-ended dialogue, potentially reducing development time.

Intelligent Tutoring Systems and Example-Tracing Tutors for Math. A wide range of intelligent tutoring systems in mathematics use precise models of student's mathematical knowledge and misunderstandings [2–4, 31, 36, 40]. To reduce the time and expertise needed to build ITSs, some researchers have proposed example-tracing tutors [3, 5, 25]. Specifically, example-tracing tutors allow content designers to specify the feedback that should appear after students provide certain answers and then record those action-feedback pairs in a behavior graph [5]. With the help of the Cognitive Tutor Authoring Tools (CTAT), Aleven et al. [2, 3, 4] built MathTutor, a suite of example-tracing tutors for teaching 6th, 7th, and 8th grade math. Our work draws from insights of example-tracing tutors in that we build a graph encoding rules that determine how MathBot responds to specific student answers, though our approach differs in that we display these responses in a conversational context.

Chatbots. Chatbots have been widely applied to various domains, such as customer service [45], college management [7], and purchase recommendation [24]. One approach of building a chatbot is to construct rule-based input to output mappings [1, 46]. One can also embed chatbot dialogue into a higher-level structure [8] to keep track of the current state of the conversation, move fluidly between topics, and collect context for later use [11, 38, 43]. We envisioned MathBot as having an explicit, predefined goal of the conversation along with clear guidance and control of intermediate steps, so we took the approach of modeling the conversation as a finite-state machine [6, 33, 35], where user responses update the conversation state according to a preset transition graph.

Conversational Agents and Chatbots in HCI. There is a wide range of HCI research on user experience (UX) design for conversational agents and chatbots. For example, Wiggins et al. [44] detail the design considerations of conversational agents acting as learning companions, Rodríguez et al. [37] investigate conversational patterns in human discussions of computer science problems, and Candello and Pinhanez [9] explore the UX design of systems involving multiple chatbots. More broadly, researchers have outlined design principles and guidelines for conversational agents [28]. For example, Cramer and Thom [13] consider how chatbots address factors such as domain-specific tone and venturing beyond the intended conversation. MathBot takes inspiration from these broad principles, while focusing on how to transform typical interactions with online educational resources (reading explanations, solving problems) into a conversational experience.

3 MATHBOT SYSTEM DESIGN & DEVELOPMENT

The high-level goal of developing MathBot is to explore how people can learn math topics (like arithmetic sequences) through conversation-style interaction, rather than simply browsing online resources like videos, written lessons, and problems. The more specific design goals are for MathBot to use a conversational context to check understanding, provide personalized feedback, guide learners' study activities, and give the experience of interacting with a supportive agent.

In this section we: (1) give an illustrative example of a learner interacting with MathBot; (2) describe MathBot's front end of interactive text chat, as well as its back end of a conversation graph that specifies a set of rules, such as how to progress through concepts and what actions to take based on user responses; (3) elaborate how each design goal is addressed by the system; (4) explain the development process and collection of user data that was used to create the rules in MathBot's conversational graph.

Sample Learner Interaction with MathBot

A learner, Alice, wants to learn about arithmetic sequences by interacting with MathBot. To start the interaction, MathBot greets Alice and asks her to extend the basic sequence "2, 4, 6, 8 ...". Alice answers correctly, so MathBot provides positive feedback (e.g. "Good work! 🥇") and starts a brief lesson on recognizing patterns in sequences. After the lesson, MathBot asks Alice if she is ready to complete a new question to check her understanding, and Alice responds affirmatively. Alice progresses successfully through a series of additional lesson and question pairs. Following a lesson on common differences, Alice is asked a new question (Figure 1a, i). Figure 1a displays the conversation rules that underlie Alice's current question.

When asked the new question, Alice confuses the term "common difference" with "greatest common factor", a topic she recently reviewed, so she answers "2". MathBot recognizes that Alice has made a mistake and subsequently checks that she knows how to identify terms in a sequence and subtract them, a prerequisite task for finding the common difference (Figure 1a, ii). Alice answers correctly, so MathBot begins to ask her a series of additional sub-questions to further clarify the concept of common differences (Figure 1a, iii). Alice successfully completes these sub-questions, so MathBot directs her back to the original question. Alice remembers learning that the common difference is the difference between consecutive terms, though she mistakenly subtracts 8 from 2 and answers "I think it's -6". Rather than have Alice finish a redundant series of sub-questions, MathBot recognizes that Alice has made a common mistake, subsequently provides specific feedback to address that mistake, and then allows Alice to retry the original question (Figure 1a, iv). Alice answers the original question correctly and proceeds to a question on identifying decreasing arithmetic sequences (Figure 1a, v).

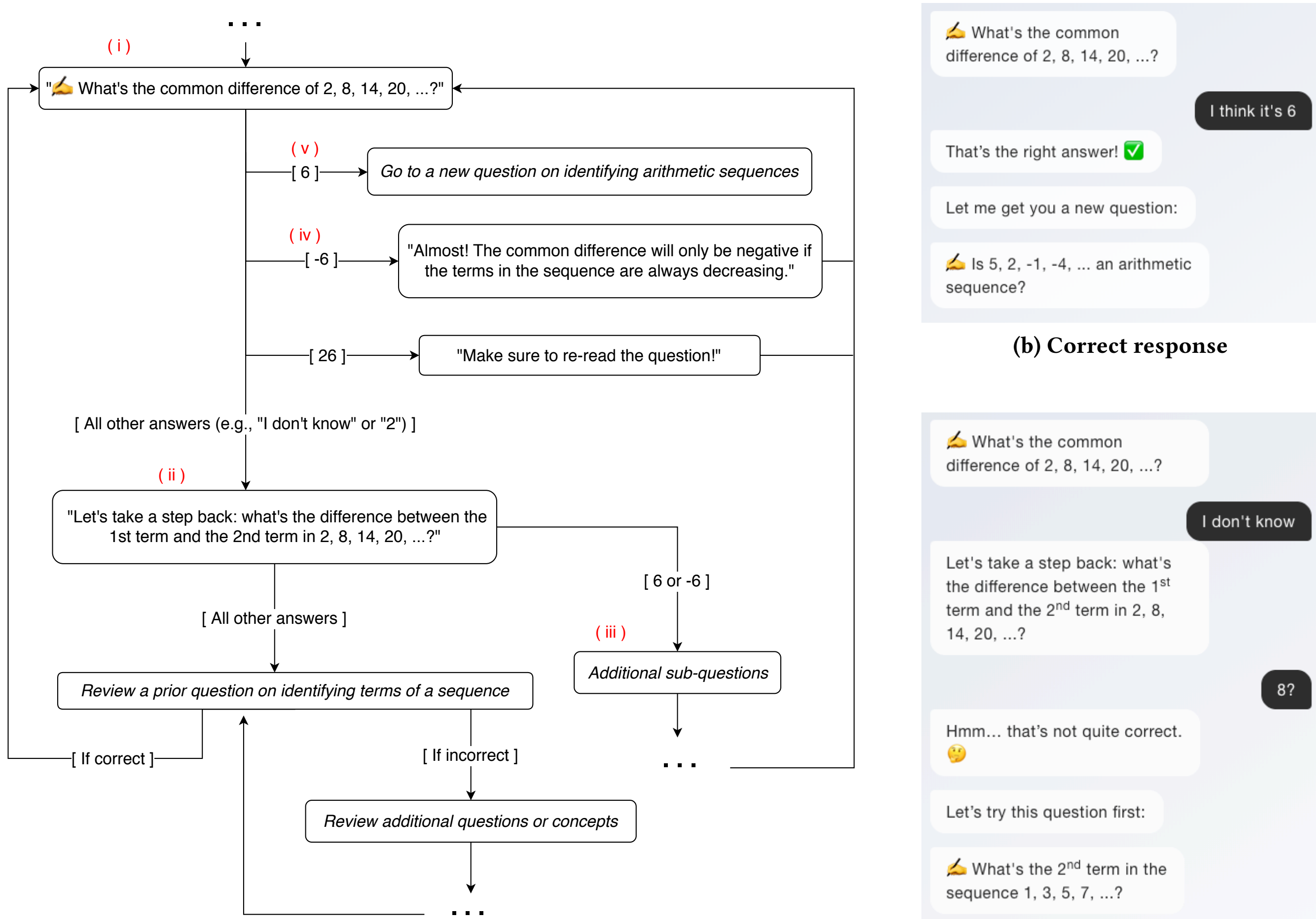
Alice spends approximately 30 minutes completing the interaction with MathBot. At the end of the interaction, MathBot praises Alice for her hard work and informs her that the interaction is finished.

MathBot Front-End Chat

The front end of MathBot is a text chat window between MathBot and the student (see Figures 1b and 1c). Students are presented with short lessons, asked to solve math problems, and shown explanations. Students type replies to MathBot into the chat to give answers to problems and provide responses like "I'm ready for the next part" or "I'm not sure". Students can freely scroll through the chat history to review concepts or questions.

MathBot Back-End Conversation Graph

The MathBot back end consists of a conversation graph that specifies a set if-then rules for how learner input (e.g. "I'm ready" or "The answer is 6") leads to MathBot's next action (e.g. continue a lesson, give a new problem, or provide feedback). In this rule-based system, the state of the conversation is represented as a finite state machine (FSM). In this FSM, each state is a response provided by MathBot, and user responses route the user along different paths in the conversation graph. For example, the question asked at the top of Figure 1a is a state, and responses to that question (e.g., "I don't know" or "6") route users to a new state.



(a) Example section of MathBot's conversation graph

(b) Correct response

(c) Incorrect responses

Figure 1: Example section of MathBot's conversation graph and sample conversations. Ellipses (...) in (a) denote excised sections of the full conversation graph. (i) – (v) in (a) denote actions taken by a hypothetical user, Alice, in Section 3.

Goal 1: Checking Understanding

The first goal of MathBot is to use conversational questions to continually check users' understanding while they are learning. This is often not possible while learners are listening to explanations from a video, as they may realize they misunderstand a concept only after they begin to solve problems.

MathBot therefore asks learners to solve problems after providing text and image explanations of concepts. When users answer incorrectly, MathBot's conversation graph breaks the problem into sub-problems that isolate and help remediate the specific concept about which the user is confused. This allows MathBot to embed the benefits of practicing problems within the same conversational context as direct instruction and explanation of concepts. For example, if users answer the question at the top of Figure 1a with "I don't know", MathBot will begin to break down the concept of common differences by asking the user to find the difference between consecutive terms.

Goal 2: Personalized Feedback

MathBot aims to provide specific feedback dependent on the user's answer to a question, such as an explanation of the learner's particular misconception.

For example, consider again the question at the top of Figure 1a. The user may answer "-6" if they don't understand when common differences are negative, or "26" if they simply extend the sequence without carefully reading the question. Each of these two common incorrect answers receives specific feedback. Such answer-specific feedback while solving problems has been shown to be effective for learning [23] and such "tailored feedback" on problems is increasingly used in settings like MOOCs.

Goal 3: Guiding Learners' Review of Concepts

MathBot aims to guide learners' study activities by progressing through concepts and corresponding problems while allowing appropriate review of concepts that learners failed to grasp. Within a particular concept, MathBot also aims to appropriately guide learners between study activities such as reading explanations, seeing examples, and solving problems.

MathBot achieves this by encoding progressions from lesson to lesson, as well as rules that indicate when inaccuracy on certain problems suggests the need for review of certain prerequisite concepts. Based on detection of whether a learner understands a prerequisite concept, MathBot may push the learner back to an earlier line of conversation and problem-solving for reviewing. This enables tailored pathways for each student.

For example, if a user answers the question at the top of Figure 1a with "I don't know" and struggles with the proceeding sub-question on finding the difference between consecutive terms, MathBot asks the user a prerequisite question on identifying terms in a sequence (Figure 1c shows the corresponding chat transcript). Depending on the user's response to this prerequisite question, the user will either (a) continue reviewing past concepts and questions until MathBot confirms the user's understanding of necessary prerequisite material or (b) return to the original question.

This approach is also taken by intelligent tutoring systems; however, MathBot does not use a formal model of users' mathematical knowledge, but instead takes a rule-based approach similar to example-tracing tutors [3]. As discussed in Section 4, we designed MathBot's rules to specify the concepts that should be reviewed after a learner answers a particular question incorrectly.

Goal 4: Interaction with a Supportive Agent

MathBot aims to give students the experience that they are interacting with a supportive agent, versus just solving problems or watching videos alone. The goal is to create a casual conversational experience analogous to communicating with a human tutor via text-chat, even without the benefit of NLP algorithms designed to handle the full range of language a student might use with a tutor.

MathBot therefore turns actions one can take with an online problem (e.g. clicking 'Next' to see more information) into questions that give the impression of conversation. For example, to mimic the effect a 'Next' button, MathBot may say something like "Let me know when you're ready to hear more!" after sending a series of messages. These prompts give users a chance to pause and reflect, though all possible user responses signal MathBot to move on.

To help users understand why MathBot presents particular concepts or questions, MathBot provides transition phrases such as "Let's revisit the concept" or "Let's try this question again". These transition phrases are not hard coded into the conversation graph; rather, MathBot detects the type of transition (e.g. completed question to new concept, or reviewed concept to question previously answered incorrectly), and selects an appropriate transition phrase from a predefined list of phrases.

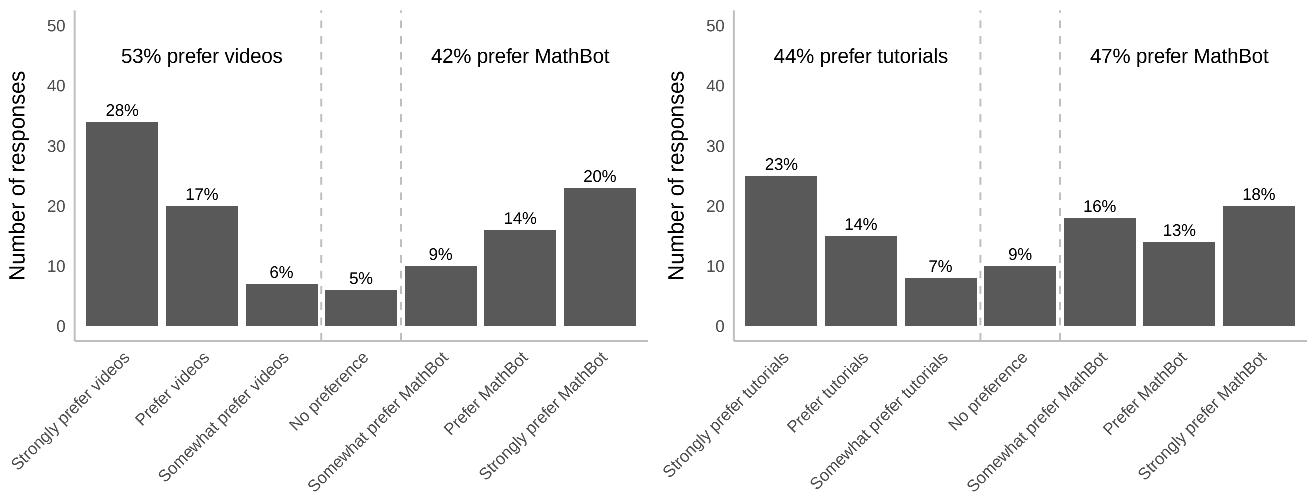
MathBot also uses a friendly tone, provides supportive cues such as transition phrases and emoji, and exhibits natural typing patterns. Correct and incorrect feedback to answers provided by MathBot users often incorporates icons or emoji that might be used in an SMS text or messaging program (e.g. "That's correct! ✅). Emoji are also used to signal key ideas (💡) and problems (✍️).

In order to give the sense of someone sending successive texts while also minimizing interruption to users while they are reading, MathBot introduces delays between messages. The delays between messages are based on estimates of how long the average user will take to read a batch of text, with the waiting time between messages increasing more substantially based on the number of math symbols (e.g. "+") in the first message, as these can take longer to comprehend than typical prose. To give the sense of an agent actively "typing" between messages, MathBot uses animated ellipses similar to those used by popular text-chat interfaces.

4 DEVELOPMENT OF RULES IN CONVERSATION GRAPH

In creating MathBot, we iteratively developed a conversational graph covering introductory arithmetic sequences at the Algebra I level. In order to experimentally compare MathBot against widely used and popular non-conversational resources (see Studies 1 and 2), we designed MathBot to address similar content as 7 Khan Academy videos and 4 Khan Academy written tutorials.

We used a multi-stage process to develop the conversation graph for MathBot. One of the authors (who has tutored high school students frequently for more than 9 years) originally defined the graph by adapting explanations, images, and problems/questions from the Khan Academy videos and interactive tutorials (containing practice problems). This author sought to keep MathBot's content as similar to Khan Academy's content as possible: for most questions and conceptual lessons, the conversation graph presents content in the same order as Khan Academy and uses identical or nearly-identical text and images. Over the course of several weeks, this author iteratively modified the conversation graph and interacted with MathBot to assess the logical flow and clarity of content. Periodically, two of the other authors interacted



(a) MathBot vs. Khan Academy Video **(b) MathBot vs. Khan Academy Written Tutorial**

Figure 2: Distributions of user preferences among the participants of Study 1.

with MathBot to provide suggestions for improving the conversational graph.

The final version of MathBot taught 10 distinct concepts and walked users through 12 key questions representative of those concepts. These concepts and questions were sorted into 6 overarching lessons: "Extending sequences", "Defining arithmetic sequences", "Common difference, "Recursive formulas", "Explicit formulas", and "Equivalent formulas".

User Testing for Enhancing Conversation Graph Rules

To get user input on the structure of the conversational graph, we conducted 6 iterative pilot studies over the course of which approximately 60 users recruited from Amazon Mechanical Turk interacted with different versions of MathBot. We recorded and examined the conversation log for each user to find common patterns of incorrect answers, identify common misconceptions, and discover use cases that required new features to be implemented. For example, we found that some users did not fully understand MathBot's explanation of recursive formulas and therefore could not move past a question involving recursive formulas. To remedy, we added an alternative explanation of recursive formulas and implemented functionality for MathBot to move on from any question after several incorrect attempts.

5 STUDY 1: LEARNING PREFERENCES

We begin our evaluation of MathBot by investigating user experiences in a two-part study.

Study Design

In the first part of this within-subjects study, we ask participants to both interact with MathBot and watch a six-minute Khan Academy video, and then solicit feedback on the two learning methods. We conduct the second part of the study identically, except we recruit new users and replace the video with a written tutorial from Khan Academy containing embedded practice problems. Comparison to interactive tutorials with embedded problems provides an additional layer of insight, as one could argue that any result favoring MathBot over video instruction may simply be the result of MathBot providing an interface to work through practice problems.

To limit the length of the study, we use an abridged version of our developed MathBot content that covers only explicit formulas for arithmetic sequences, and pair that with either a Khan Academy video or written tutorial that covers similar material. To avoid ordering effects—including anchoring bias and fatigue—we randomized the order in which participants saw MathBot and the Khan Academy video or written tutorial.

Participants

Our study was conducted on Amazon Mechanical Turk and was restricted to adults in the United States. To qualify for the study, we required that participants pass two screening tests. The first was a brief, 5-question quiz to ensure participants had sufficient algebra knowledge to understand sequences, but did not already have advanced knowledge of arithmetic sequences. The second screening test consisted of a more in-depth set of 12 questions selected from a Khan Academy quiz on arithmetic sequences. We excluded participants who answered more than 6 of the 12 questions correctly, reasoning that these individuals already had substantial knowledge of sequences. Finally, we excluded participants who spent less than one minute on either MathBot or the Khan Academy learning module, reasoning that these individuals did not intend on taking the study seriously. After these filtering criteria, there remained 116 participants in the first part of the study and 111 participants in the second part. Our analysis is restricted to this set of users.

All participants received \$0.10 for completing the first screening test, and \$0.25 for the second, regardless of their final eligibility. Eligible users were paid up to \$6 more according to their score on a post-learning test. This performance-based payment scheme was disclosed to participants at the start of the study to incentivize active engagement with MathBot, attentive watching of the Khan Academy video, and dutiful completion of the written tutorial.

Quantitative Results

After study participants completed the MathBot and Khan Academy learning modules, we asked them a series of questions to quantify their experiences. In particular, we asked participants whether they would prefer to continue learning about sequences via MathBot or by watching Khan Academy videos, on a 7-point scale ranging from "strongly prefer videos" to "strongly prefer MathBot". The results of this question for the first part of the study are presented in Figure 2a. We found that 42% of participants stated at least a weak preference for MathBot, 53% stated at least a weak preference for Khan Academy videos, and 5% stated having no preference. Notably, 20% of participants indicated a strong preference for MathBot over videos.

The corresponding results for the second part of the study are displayed in Figure 2b. We found that 47% of the 110 participants who answered the question stated at least a weak preference for MathBot, 44% stated at least a weak preference for Khan Academy interactive tutorials, and 9% stated having no preference. 18% of participants indicated a strong preference for MathBot over written tutorials. These results illustrate the promise of our approach, as a non-negligible fraction of the population has a clear preference for the teaching style of MathBot over traditional video-based instruction, while another has a preference for MathBot over written tutorials with embedded problems.

We also asked study participants to rate their experience with each learning module along the 7-point NASA Task Load Index (NASA-TLX) scales. As shown in Figure 3a, participants generally rated MathBot and the Khan Academy video comparably on the six dimensions measured. The largest difference we found was for effort level, with users reporting that MathBot, on average, required greater effort (average rating of 3.2 on a 0 through 6 scale) than watching the video (2.4). This result is consistent with the fact that watching a video is a relatively passive activity compared to the back-and-forth interactivity of MathBot. We also found that users reported higher frustration levels for MathBot (2.5) than for watching the video (1.9). In part, this may have been due to problems with our implementation of MathBot (e.g. some users reported system-specific issues in the rendering of emoji); further, the greater interactivity of MathBot may inherently create more opportunities for frustration if users fail to correctly answer comprehension questions. Overall, the slightly lower ratings for MathBot are consistent with a greater proportion of the study population indicating a preference for videos.

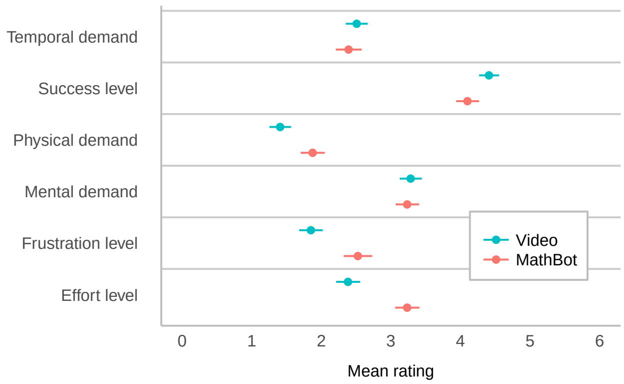
Figure 3b shows the corresponding results for the second part of the study. Participants appear to have rated MathBot and the written tutorial more disparately than MathBot and the video on several dimensions. Temporal demand exhibits by far the largest difference, with users reporting that MathBot, on average, required more time (3.0) than completing the written tutorial (1.7). This result is consistent with the fact that there is no limit to the speed at which a user can read through a written tutorial. A user responding immediately and correctly will still be limited by the speed at which MathBot generates responses. Users also reported that the written tutorial was more mentally demanding (3.9) than MathBot (3.1), and that the tutorial required more effort (3.4) than MathBot (2.9). This may be due to (a) the written tutorial containing more problems than MathBot and (b) MathBot's presentation of text in digestible chunks.

Qualitative Results

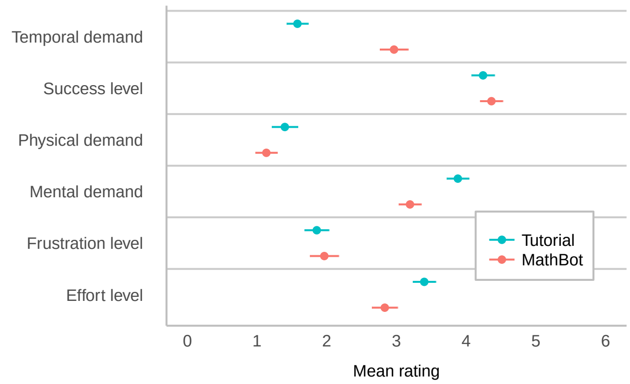
We asked a set of open-ended questions after each part of the study to elicit insights from users about the relative affordances of MathBot and the Khan Academy videos and written tutorials with embedded problems. We analyzed the resulting comments to identify themes and understand users' perspectives on these three ways of learning.

In particular, we asked users to indicate when they thought MathBot could be more effective or less effective than the Khan Academy material, and vice-versa. We also asked users to explain their rating of questions that asked how the tone, emoji, and pace of MathBot enhanced or hindered their learning.

Self-pacing versus guidance Some users appreciated that they could freely navigate the video: *"I can rewind them and fast forward if I already know the concept."* Similarly, certain users preferred to freely scroll through the written tutorial: *"I like to learn at my own pace and be able to skip around."* These users were often frustrated that they could not freely navigate the material in the MathBot conversation: *"With*



(a) MathBot vs. Khan Academy Video



(b) MathBot vs. Khan Academy Written Tutorial

Figure 3: Results from the NASA Task Load Index (NASA-TLX) assessment from participants of Study 1.

the computer [program] I felt like if I got an answer wrong I had to start over and I could see myself getting [frustrated] and ending the computer program."

On the other hand, some users preferred that MathBot adapted its speed to their progression through concepts and questions, unlike the video: *"The video is paced for you, whereas you determine the speed of the computer simulation by interacting with it."* Similar sentiments were echoed by users who preferred that MathBot explicitly guided them through concepts, unlike the interactive tutorial: *"I liked the conversational program a lot with how it walked things step by step through the process instead of the interactive tutorial just basically giving a list of steps to do."*

Human elements and interactivity Some users found MathBot to be more human/agentic than the video: *"The conversational computer program felt more similar to the experience of interacting with a real teacher."* Others reported the exact reverse: *"Even though it was a video, it felt like a more personal experience because it was a human voice talking versus just reading on the screen."* Users generally indicated that MathBot provided a great sense of interactivity than the written tutorial: *"The conversational program was more like a real person and was more engaging. The tutorial made me feel like I was teaching myself."*

Requiring users to evaluate their knowledge The video asked users to pause and think about problems; however, unlike MathBot, answering these questions correctly was not required. Several users noted the value of MathBot holding them accountable for understanding concepts before progressing: *"When watching the video, I wasn't sure if I was actually understanding the concepts correctly."* Similarly, although the tutorial embedded problems between text, users could easily skip them: *"The program helped me pace myself while slowly giving me the appropriate information over time, while the tutorial I could just breeze through with no consequences."*

Learners also valued that MathBot provided more specific feedback on their answers than the tutorial: *"I need what I did wrong to be explained to me. The interactive [tutorial] wasn't bad but you had to pay attention and really read and teach yourself what you think you did wrong and try another approach."*

Combining learning modules Several users noted that each learning module had particular strengths and weaknesses, and thus suggested that combining them would be most effective. For example, some users thought the video was superior for learning concepts, whereas MathBot was better for learning how to apply those concepts: *"The best option for me would be to watch the video first, and then take part in the conversational computer program so that I could verify my understanding."* Similarly, other users indicated that, like the video, the written tutorial introduced concepts more effectively: *"I think the interactive tutorial was better at presenting the information but the chat bot was far better when inputting answers and getting feedback. I think a mixture of the two would be the right blend."* We further address the prospect of combining learning modules in Section 7.

6 STUDY 2: LEARNING EFFECTIVENESS

Our first study indicated that a substantial fraction of the study population preferred MathBot over Khan Academy videos or interactive tutorials. One might worry, however, that MathBot does not provide the same level of educational benefit as a video or written tutorial. We thus directly investigate learning effectiveness in our second study.

Study Design

To assess educational gains, we randomly assigned participants to learn about sequences via MathBot or via Khan Academy videos and written tutorials. In contrast to Study 1, the learning modules covered a more expansive set of topics on arithmetic sequences, including recursive formulas. The Khan Academy video instruction ran for approximately 45 minutes, spread over 7 separate videos. Users assigned to the Khan Academy condition also had access to 4 written tutorials with embedded practice problems, and were free to learn the material through either method—videos or written tutorials—or a combination of the two.

We assessed learning outcomes with a 12-question test, with the same test administered both before and after each participant completed the learning module. The difference in pre-module and post-module test scores is our measure of learning gain. To ensure that the Khan Academy materials sufficiently prepared participants for this test, we selected relevant questions directly from an arithmetic sequences quiz on Khan Academy. We note that this test was the same as the second eligibility screen used in Study 1.

Participants

As in Study 1, we filter users according to their performance on the same two screening mechanisms, restricting to users who both know enough math to understand the presented material but not so much that they have nothing left to learn. In this case, the second eligibility test does double duty: filtering the population, and assessing base knowledge to measure learning gains. As before, we also restricted our analysis to those individuals who spent at least 2 minutes on their assigned learning module, and compensated participants according to their post-module test score.[1] These filtering criteria resulted in our analyzing 182 subjects assigned to MathBot, and 188 assigned to Khan Academy videos and written tutorials.

Results

We start by computing the average difference between pre- and post-module test scores for users of MathBot and Khan Academy videos and written tutorials, where scores can range from 0 to 12, with one point per question. We find the average learning gain for MathBot users is 6.1 points (from a score of 2.6 to 8.6), with a 95% confidence interval of [5.6, 6.6]; the corresponding average gain for Khan Academy users is 5.7 points (from a score of 2.3 to 8.0), with a 95% confidence interval of [5.2, 6.2]. This result suggests

Figure 4: The distributions of the differences in pre- and post-module test scores across users of MathBot and Khan Academy in Study 2. It appears that both learning methods are similarly effective at conveying information.

that MathBot and Khan Academy are comparably effective tools for learning. We note that the gains from MathBot are slightly higher than those from Khan Academy, but the difference is not statistically significant (Welch's t-test, p = 0.2, 95% CI: [-0.269, 1.128]). Figure 4 shows the full distribution of learning gains for both teaching tools, and again illustrates the similar effectiveness of the two methods.

We corroborate these results via a linear regression that estimates the difference in pre- and post-module test scores after controlling for age, gender, education, and pre-module test score. The fitted model coefficients are shown in Table 1. As above, we see that MathBot users achieve slightly higher—though not statistically significant—learning gains (p = 0.16). We further see that pre-module test score has a negative coefficient (-0.40, $p < 0.01$), indicating that those with higher base knowledge experience diminished returns, possibly because of a ceiling effect.

Finally, we note that MathBot and Khan Academy users spent comparable time completing the learning modules—28.4 minutes on average for MathBot (SD = 20.3) and 28.9 minutes for the Khan Academy videos and interactive tutorials (SD = 21.5). Both tools thus appear to be similarly effective and efficient at conveying the presented information.

7 DISCUSSION, LIMITATIONS, & FUTURE WORK

Although the content and problems in MathBot were closely matched to the Khan Academy written tutorials and videos, we found that 42% of users preferred learning with MathBot over videos, and 47% of users preferred learning with MathBot over written tutorials. Some users appreciated that MathBot *"felt more similar to the experience of interacting with a real teacher"*, saying that *"I need what I did wrong to be explained to me"*, and they appreciated being guided through explanations and problems. MathBot wasn't any

[1] In Study 1, we required users spend at least one minute on each of MathBot and the Khan Academy material. Here, though, participants were assigned to view material from only one platform, and so for consistency we required they spend at least two minutes on the lesson.

Table 1: Linear regression analysis of the difference in pre- and post-module test scores in Study 2, with standard errors in parentheses.

	Difference in test scores
MathBot user	0.49 (0.35)
Pre-learning test score	-0.40^{***} (0.10)
Male	-0.11 (0.38)
Age (years)	-0.01 (0.02)
Has college degree	0.42 (0.46)
Intercept	6.60^{***} (0.78)
Observations	368
Residual Std. Error	3.35 (df = 362)

*p<0.1; **p<0.05; ***p<0.01

less effective for learning than the Khan Academy resources, resulting in an average learning gain slightly higher than that of Khan Academy videos and written tutorials.

The successful conversational experience is especially noteworthy, because MathBot achieves this *without* the capacity to handle open-ended dialogue. The kind of conversation realized in MathBot can therefore be complementary to past work on conversational tutors, which use a range of NLP techniques [18, 21, 30, 32, 42]. Of course, MathBot is necessarily limited without the use of dialogue, and MathBot focuses more on developing the acquisition of procedural knowledge through solving problems than the development of conceptual understanding at which conversational tutors typically excel [10, 27, 41]. A valuable future direction could be to integrate user interface insights from MathBot with existing conversational tutors [16, 20, 30], which have only recently begun to be applied to math [17, 18, 21].

The approach realized in our MathBot prototype may also yield insights for bringing conversational interactions to online math homework and example-tracing tutors. MathBot uses a conversation graph to integrate text chat with ubiquitous elements of online math homework (e.g. problems and explanations) and the guidance and feedback of example-tracing tutors.

MathBot could be limited in its broader applicability because extensive time is needed to develop and test the rules in the conversation graph. On the other hand, since it does not require researchers to develop NLP algorithms and models for conversation, it has one of the strengths of example-tracing tutors, in that teachers might be able to participate in development. Just as teachers put extensive time into creating curricula, future work could explore whether the broader approach instantiated in MathBot's conversation graph could help teachers create such conversational programs. Since MathBot's applicability to a classroom setting is yet to be explored, future work can explore how this approach would be received and used by teachers. For example, would MathBot be most useful as homework, an optional supplementary resource, or as in-class practice that is not as disruptive as playing videos? Might interaces be provided that integrate teachers into the loop, such that they could strategically choose when to type text responses, or choose from a library of existing prompts? What age and level of student do teachers think MathBot is most helpful for?

A key limitation of our study is that we evaluated MathBot using a convenience sample of adults from Amazon Mechanical Turk. In the future, it would be valuable to test our system with a population actively exposed to algebra instruction, such as high school students or learners on Khan Academy.

Additionally, our system taught a single algebra topic, arithmetic sequences, with a conversation intended to last approximately 30 minutes (Study 2) and could be as short as 5 minutes (Study 1). Some of our insights will generalize to longer interaction periods and different mathematics topics, while others may not. Further work is necessary to understand the exact scope of our insights. Our study also does not address the implications of using MathBot as a major component of a full-length course. For example, we did not investigate knowledge retention, and we do not know whether students would enjoy using MathBot less or more if they used it to learn over the course of several weeks or months.

We note several exciting directions for future work on MathBot. Several users in Study 1 noted the benefit of interacting with multiple learning modules, and past work has demonstrated that prompting users with relevant questions periodically during a video may improve learning outcomes [39]. Perhaps brief conversations with MathBot could take place periodically during an educational video, or video elements could be used in the MathBot conversation.

The wide variation in user opinions on MathBot's pacing and tone suggests value in methods for discovering how to personalize beyond the expert conversational graph. For example, some of MathBot's actions could be randomized, and algorithms like this for contextual bandits [26] could be used to determine which pathways are best for learners whose profiles differ in response times, past performance, and other variables.

Lastly, the chatbot model could be particularly promising for locales lacking internet connections with sufficient bandwidth for video streaming, as MathBot could be delivered via SMS to basic mobile phones.

8 CONCLUSION

We presented a prototype system, MathBot, which restructured existing online math tutorials and problems so that students could learn through a conversational text chat, even without algorithms to process natural language dialogue. We obtained qualitative evidence that many users found the conversational experience was engaging and gave a sense of the back-and-forth interaction with a tutor, even though we used no open-ended dialog. A sizable minority of learners preferred to learn using MathBot over videos and tutorials with embedded problems. They appreciated the friendly tone, and the appropriate interleaving of multiple components: Asking questions (giving problems) to check comprehension, tailoring feedback to incorrect answers, and guiding learners to review concepts or answer easier or harder questions (problems). This approach to learning math through conversation was at least as effective for learning as existing online resources, and may thus be an attractive and effective complement to online math education.

ACKNOWLEDGMENTS

We thank Keith Shubeck, Carol Forsyth, Ben Nye, Weiwen Leung, Ro Replan, and Sam Maldonado for helpful comments and discussions. This work was supported by the Office of Naval Research.

REFERENCES

[1] Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372* (2016).

[2] Vincent Aleven, Bruce M McLaren, and Jonathan Sewall. 2009. Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies* 2, 2 (2009), 64–78.

[3] Vincent Aleven, Bruce M Mclaren, Jonathan Sewall, and Kenneth R Koedinger. 2009. *A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors.* Technical Report. 105–154 pages. https://www.learnlab.org/opportunities/summer/readings/AlevenMcLarenSewallKoedinger2009.pdf

[4] Vincent Aleven, Bruce M Mclaren, Jonathan Sewall, and Kenneth R Koedinger. 2009. *A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors.* Technical Report. 105–154 pages. https://www.learnlab.org/opportunities/summer/readings/AlevenMcLarenSewallKoedinger2009.pdf

[5] Vincent Aleven, Bruce M. McLaren, Jonathan Sewall, Martin van Velsen, Octav Popescu, Sandra Demi, Michael Ringenberg, and Kenneth R. Koedinger. 2016. Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers. *International Journal of Artificial Intelligence in Education* 26, 1 (mar 2016), 224–269. https://doi.org/10.1007/s40593-015-0088-2

[6] Pierre Andrews, Marco De Boni, Suresh Manandhar, and Marco De. 2006. Persuasive Argumentation in Human Computer Dialogue.. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare.* 8–13.

[7] K Bala, Mukesh Kumar, Sayali Hulawale, and Sahil Pandita. 2017. ChatBot For College Management System Using AI. *International Research Journal of Engineering and Technology* (2017).

[8] Daniel G Bobrow and Terry Winograd. 1977. An overview of KRL, a knowledge representation language. *Cognitive science* 1, 1 (1977), 3–46.

[9] Heloisa Candello and Claudio Pinhanez. 2017. Designing the user experience of a multi-bot conversational system. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17).* ACM.

[10] Min Chi, Pamela W Jordan, Kurt Vanlehn, and Pamela Jordan. 2014. When Is Tutorial Dialogue More Effective Than Step-Based Tutoring? Affective Meta Tutor View project Formative Assessment with Computational Technologies (FACT) View project When Is Tutorial Dialogue More Effective Than Step-Based Tutoring? (2014). https://doi.org/10.1007/978-3-319-07221-0_25

[11] Jennifer Chu-Carroll and Michael K Brown. 1997. Tracking initiative in collaborative dialogue interactions. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 262–270.

[12] Scotty D. Craig, Xiangen Hu, Arthur C. Graesser, Anna E. Bargagliotti, Allan Sterbinsky, Kyle R. Cheney, and Theresa Okwumabua. 2013. The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors. *Computers & Education* 68 (oct 2013), 495–504. https://doi.org/10.1016/J.COMPEDU.2013.06.010

[13] Henriette Cramer and Jennifer Thom. 2017. Moving Parts surrounding Conversational UX. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17).* ACM.

[14] Jean-Claude Falmagne, Dietrich Albert, Christopher Doble, David Eppstein, and Xiangen Hu. 2013. *Knowledge spaces: Applications in education.* Springer Science & Business Media.

[15] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction* 19, 3 (2009), 243–266.

[16] Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers* 36, 2 (may 2004), 180–192. https://doi.org/10.3758/BF03195563

[17] Arthur C. Graesser, Danielle S. McNamara, and Kurt VanLehn. 2005. Scaffolding Deep Comprehension Strategies Through Point&Query, AutoTutor, and iSTART. *Educational Psychologist* 40, 4 (dec 2005), 225–234. https://doi.org/10.1207/s15326985ep4004_4

[18] Arthur C Graesser, Phanni Penumatsa, Matthew Ventura, Zhiqiang Cai, and Xiangen Hu. [n. d.]. Using LSA in AutoTutor: Learning through mixed initiative dialogue in natural language. *Handbook of latent semantic analysis* ([n. d.]), 243–262.

[19] Arthur C. Graesser, Natalie K. Person, and Joseph P. Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology* 9, 6 (dec 1995), 495–522. https://doi.org/10.1002/acp.2350090604

[20] Arthur C. Graesser, Kurt VanLehn, Carolyn P. Rose, Pamela W. Jordan, and Derek Harter. 2001. Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine* 22, 4 (dec 2001), 39–39. https://doi.org/10.1609/AIMAG.V22I4.1591

[21] Arthur C Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, and Roger Kreuz. 1999. AutoTutor: A simulation of a human tutor. *Cognitive Systems Research* 1, 1 (dec 1999), 35–51. https://doi.org/10.1016/S1389-0417(99)00005-4

[22] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.

[23] Neil T Heffernan and Kenneth R Koedinger. 2002. An intelligent tutoring system incorporating a model of an experienced human tutor. In *International Conference on Intelligent Tutoring Systems*. Springer, 596–608.

[24] Adrian Horzyk, S Magierski, and Grzegorz Miklaszewski. 2009. An Intelligent Internet Shop-Assistant Recognizing a Customer Personality for Improving Man-Machine Interactions. *Recent Advances in intelligent information systems* (2009), 13–26.

[25] Kenneth R Koedinger, Vincent Aleven, Neil Heffernan, Bruce Mclaren, and Matthew Hockenberry. [n. d.]. *Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration*. Technical Report. www.carnegielearning.com

[26] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[27] Bruce M. McLaren, Krista E. DeLeeuw, and Richard E. Mayer. 2011. Polite web-based intelligent tutors: Can they improve learning in classrooms? *Computers & Education* 56, 3 (apr 2011), 574–584. https://doi.org/10.1016/J.COMPEDU.2010.09.019

[28] Robert J. Moore, Raphael Arar, Guang-Jie Ren, and Margaret H. Szymanski. 2017. Conversational UX Design. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 492–497. https://doi.org/10.1145/3027063.3027077

[29] Benjamin D. Nye, Arthur C. Graesser, and Xiangen Hu. 2014. AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education* 24, 4 (dec 2014), 427–469. https://doi.org/10.1007/s40593-014-0029-5

[30] Benjamin D. Nye, Philip I. Pavlik, Alistair Windsor, Andrew M. Olney, Mustafa Hajeer, and Xiangen Hu. 2018. SKOPE-IT (Shareable Knowledge Objects as Portable Intelligent Tutors): overlaying natural language tutoring on an adaptive learning system for mathematics. *International Journal of STEM Education* 5, 1 (dec 2018), 12. https://doi.org/10.1186/s40594-018-0109-4

[31] Eleanor O'rourke, Erik Andersen, Sumit Gulwani, and Zoran PopoviÂťc PopoviÂťc. 2015. A Framework for Automatically Generating Interactive Instructional Scaffolding. (2015). https://doi.org/10.1145/2702123.2702580

[32] Natalie K Person. 2003. AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head. *Artificial intelligence in education: Shaping the future of learning through intelligent technologies* 97 (2003), 47.

[33] Silvia Quarteroni and Suresh Manandhar. 2007. A chatbot-based interactive question answering system. *Decalog 2007* (2007), 83.

[34] Rahul Ramachandran, Sunil Movva, Xiang Li, Prashanth Anantharam, and Sara Graves. 2007. Wxguru: An ontology driven chatbot prototype for atmospheric science outreach and education. *Geoinformatics* (2007), 17–18.

[35] Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 629–637.

[36] Steven Ritter, John R. Anderson, Kenneth R. Koedinger, and Albert Corbett. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review* 14, 2 (apr 2007), 249–255. https://doi.org/10.3758/BF03194060

[37] Fernando J. Rodríguez, Kimberly Michelle Price, Mickey Vellukunnel, and Kristy Elizabeth Boyer. 2017. Toward Conversational Agents that Support Learning: A Look at Human Collaborations in Computer Science Problem Solving. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM.

[38] Stephanie Seneff. 1992. TINA: A natural language system for spoken language applications. *Computational linguistics* 18, 1 (1992), 61–86.

[39] Hyungyu Shin, Eun-Young Ko, Joseph Jay Williams, and Juho Kim. 2018. Understanding the Effect of In-Video Prompting on Learners and Instructors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 319.

[40] Kurt VanLehn. 1996. Conceptual and meta learning during coached problem solving. (1996), 29–47. https://doi.org/10.1007/3-540-61327-7_99

[41] Kurt VanLehn, Arthur C. Graesser, G. Tanner Jackson, Pamela Jordan, Andrew Olney, and Carolyn P. Rosé. 2007. When Are Tutorial Dialogues More Effective Than Reading? *Cognitive Science* 31, 1 (feb 2007), 3–62. https://doi.org/10.1080/03640210709336984

[42] Kurt VanLehn, Pamela W Jordan, Carolyn P Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, et al. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *International Conference on Intelligent Tutoring Systems*. Springer, 158–167.

[43] Marilyn Walker and Steve Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 70–78.

[44] Joseph B. Wiggins, Lydia G. Pezzullo, Kristy Elizabeth Boyer, Bradford W. Mott, Eric N. Wiebe, and James C. Lester. 2017. Conversational UX Design for Kids: Toward Learning Companions. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM.

[45] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3506–3510.

[46] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 55–64.