

MathBot: A Personalized Conversational Agent for Learning Math

William Cai
Stanford University
Stanford, CA, USA
willcai@stanford.edu

Joshua Grossman
Stanford University
Stanford, CA, USA
jgrossman@stanford.edu

Zhiyuan Lin
Stanford University
Stanford, CA, USA
zylin@cs.stanford.edu

Hao Sheng
Stanford University
Stanford, CA, USA
haosheng@stanford.edu

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

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

Sharad Goel
Stanford University
Stanford, CA, USA
sagoel@stanford.edu

ABSTRACT

Online math education often lacks key features of in-person instruction, such as personalized feedback. To emulate such interactivity, we developed MathBot, a rule-based chatbot that explains math concepts, provides practice questions, and offers tailored feedback. We evaluated MathBot through three Amazon Mechanical Turk studies in which participants learned about arithmetic sequences. In the first study, we found that more than 40% of our participants indicated a preference for learning with MathBot over videos and written tutorials from Khan Academy that covered similar material. Although more participants preferred the Khan Academy materials, our results point to demand for alternative forms of online education. The second study measured learning gains, and found that MathBot produced comparable gains to the Khan Academy videos and tutorials. Combined with the findings of our first study, these results indicate that conversational agents can appeal to a substantial proportion of the population without sacrificing learning. Finally, in the third study, we integrated a contextual bandit algorithm into MathBot to experiment with different personalization strategies. Compared to a randomized A/B experiment, we found that the contextual bandit learned a similarly effective pedagogical policy at a lower cost. Our findings suggest that personalized conversational agents are

promising tools to complement existing online resources for math education.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Chatbot; contextual bandit; online education

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 interactive practice problems and explanations. These resources, however, 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 examines the efficacy of a learning format that more closely mimics some key facets of conversation with a human tutor. We specifically designed and evaluated a prototype chatbot system, which we call MathBot. To achieve conversational flow and mirror the experience of interacting with a human tutor, we paid close attention to the timing of prompts and incorporated informal language and emoji. As with a human tutor, the MathBot system alternates between presenting material and gauging comprehension. MathBot also provides learners with personalized feedback and guidance in the form of explanations, hints, and clarifying sub-problems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 123-4567-24-567/08/06...\$15.00

DOI: http://dx.doi.org/10.475/123_4

We draw inspiration from existing online math platforms with features like tailored feedback and personalized guidance, but which are not themselves presented as a conversation. For example, online math homework tools like ASSISTments [18, 25] give feedback on common wrong answers. And online resources like MathTutor [3] build on example-tracing tutors [5], which model the progression of a lesson with a behavior graph that: (1) outlines potential student actions, such as providing common incorrect responses; and (2) specifies the feedback, explanation, or new problem that should follow those actions. That approach aims to reduce development time while achieving some of the benefits of intelligent tutoring systems for mathematics, like personalized selection of problems [14, 17, 37]. The development process for MathBot is similar, with the key difference being that we combine such an approach with conversational interactions.

One goal of our work is to explore the possibility of creating effective conversational experiences *without* needing to support truly open-ended dialogues. Past work on conversational tutors in education has largely involved the creation of complex conversations around questions like: “What is the direction of acceleration for keys dropped in an elevator? Why?” [19, 21, 23, 24, 36, 41]. These types of conversational tutors have been shown to benefit student learning [10, 14, 33, 51], but creating such systems requires 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 [26, 37]. Our approach, in contrast, facilitates more rapid development of content while still providing a conversational mode of interaction.

When developing an interactive conversation, a content designer must choose an appropriate pedagogical strategy: for example, should the topic be conveyed through conceptual lessons or practice problems, and how much feedback should be provided? To choose an optimal pedagogical strategy for every new piece of content, one could turn to cognitive and educational experts and draw from educational theory, such as aptitude treatment interaction [49]. In practice, however, it can be difficult to operationalize such theories to create effective strategies [57]. We instead carry out personalization via a *contextual bandit* algorithm, a popular technique from the reinforcement learning (RL) literature [31] that aims to learn effective strategies from the data. Further, compared to traditional A/B tests, bandit algorithms can often learn personalized strategies with substantially less experimentation, leading to improved user experiences.

To evaluate MathBot, we carried out three Amazon Mechanical Turk studies. In the first study, 116 participants completed (in a random order) both an abridged lesson about arithmetic sequences with MathBot and a video on Khan Academy covering similar content; these participants then rated their experiences. We found that 42% of users preferred learning with MathBot over the video, with 20% indicating a strong preference. While MathBot was not preferred by the majority of our participants, our results point to demand for conversational agents among a substantial fraction of learners. An additional 110 participants completed the same abridged lesson with

MathBot along with a written tutorial from Khan Academy containing embedded practice problems. In this case, 47% of these users preferred learning with MathBot over the written tutorial, with 18% stating a strong preference.

To assess learning gains, we conducted a second study in which we randomized 369 participants to either complete a full-length conversation with MathBot about arithmetic sequences or complete a set of videos and written tutorials from Khan Academy covering similar content. To test their knowledge, each subject took an identical quiz before and after completing their assigned learning module. Under both conditions, participants exhibited comparable average learning gains: 65% improvement under MathBot and 60% with the Khan Academy material; we note that the difference between the two was not statistically significant.

The results of the first two studies indicate that conversational agents may effectively complement existing tools for online education. Our prototype system achieves learning outcomes on par with Khan Academy videos and written tutorials with embedded practice problems, and is preferred by a substantial minority of users. To explore the potential of MathBot to support personalized pedagogical strategies, we conducted a third study in which we recruited 405 participants to complete a full-length conversation with MathBot about arithmetic sequences. We randomized participants into one of two ongoing experiments to find the optimal strategy of skipping conceptual lessons and/or providing additional practice questions, with the ultimate goal of reducing learning time without reducing learning gains. We found that the contextual bandit was able to find a pedagogical policy with comparable performance to one learned by a multi-factor A/B experiment, in addition to having lower user dropout and higher learning efficiency during the experimentation.

In summary, our contributions are: (1) MathBot, a prototype system that adds conversational interaction to learning mathematics through 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 containing embedded practice problems; (3) evidence that a contextual bandit can continuously personalize an educational conversational agent at a lower cost than a traditional A/B test.

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 (ITSs). We also discuss the implementation and 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 [19, 24, 36, 52] arose from Graesser et al.’s investigation of human tutoring behaviors [22] 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 [52], LSA [24, 39], and other semantic analysis tools [21]. Nye et al. added conversational routines to the online mathematics ITS ALEKS by attaching mini-dialogues to individual problems, but leaving navigation on the website [37]. 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

A wide range of intelligent tutoring systems in mathematics use precise models of students' mathematical knowledge and misunderstandings [3, 4, 38, 43, 50]. To reduce the time and expertise needed to build ITSs, some researchers have proposed example-tracing tutors [4, 5, 28]. 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]. Using the Cognitive Tutor Authoring Tools (CTAT), Aleven et al. built MathTutor, a suite of example-tracing tutors for teaching 6th, 7th, and 8th grade math [3, 4]. 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 [55], college management [7], and purchase recommendation [27]. One approach to building a chatbot is to construct rule-based input-to-output mappings [2, 56]. 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 [12, 47, 53]. We envisioned MathBot as having an explicit, predefined goal of the conversation along with clear guidance of intermediate steps, so we took the approach of modeling the conversation as a finite-state machine [6, 40, 42], 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. detail the design considerations of conversational agents acting as learning companions [54], Rodriguez et al. investigate conversational patterns in human discussions of computer science problems [44], and Candello et al. explore the UX design of systems involving multiple chatbots [9]. More broadly, researchers have outlined design principles and guidelines for conversational agents [35]. For example, Cramer and Thom consider how chatbots address factors such as domain-specific tone and venturing beyond the intended conversation [15]. MathBot takes inspiration from these broad principles while transforming typical interactions with online educational resources into a conversational experience.

Learning Pedagogical Strategies with Contextual Bandits

To allow MathBot to learn how to personalize elements during live deployment, we incorporate a contextual multi-armed bandit algorithm [29, 31], a tool from reinforcement learning, for discovering which actions are effective in different situations (contexts). Other reinforcement learning approaches have been applied in education, typically for offline learning. Ruan et al. [45] increased student performance by combining adaptive question sequencing with an NLP-based conversational tutor for teaching factual knowledge, but use a combination of random selection and a probabilistic model of learners' knowledge of particular items to order questions. Chi et al. [11] use another popular technique from RL, a Markov Decision Process model, to learn an effective pedagogical strategy for making micro-decisions, such as eliciting the next step of the problem versus revealing it, in an NLP-based ITS teaching college-level physics. Lan and Baraniuk [30] develop a contextual bandit framework to assign students to an education format and optimize performance on an immediate follow-up assessment, but evaluate the performance of the framework offline and do not personalize the actual lessons. However, a key difference between these studies and MathBot is that it is rare to use these strategies online in a live educational deployment. Only a handful of studies have begun to explore live deployments for sequencing problems [13, 46], but not for learning which actions to take in a conversation.

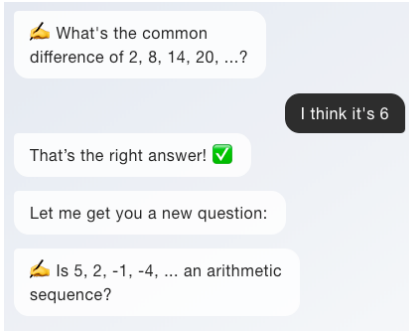
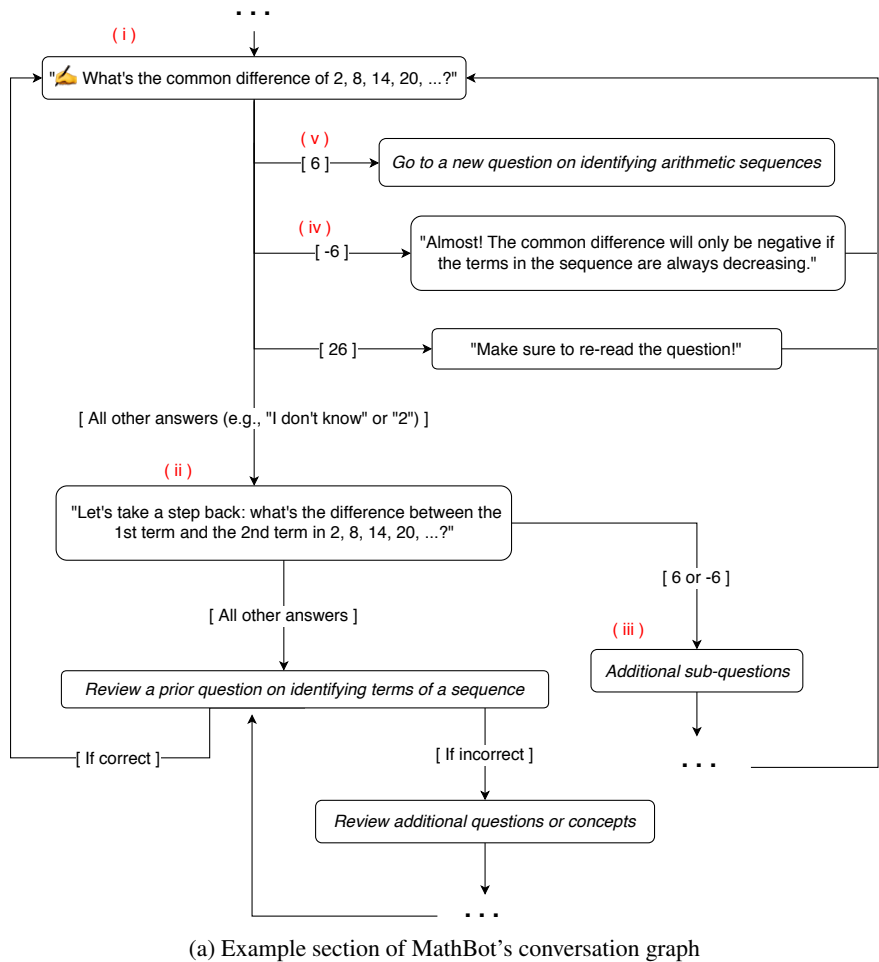
MATHBOT SYSTEM DESIGN & DEVELOPMENT

The high-level goal of developing MathBot is to explore how people can learn math topics 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.

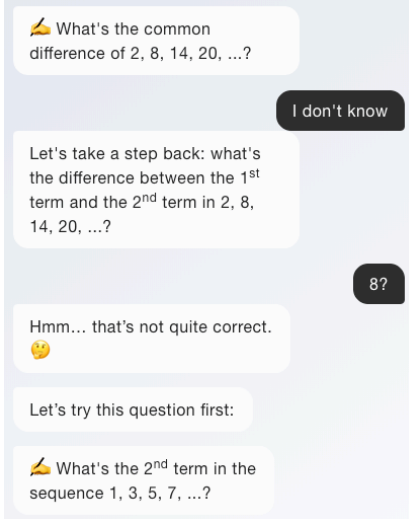
In this section, we: (1) give an illustrative example of a learner interacting with MathBot; (2) describe MathBot's front end of interactive text chat, as well as its back end of a conversation graph that specifies a set of rules, such as how to progress through concepts and what actions to take based on user responses; (3) elaborate on the design goals addressed by the system; and (4) explain the development process 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 begins a conceptual explanation of recognizing patterns in sequences. MathBot asks Alice if she is ready to complete a question to check her understanding, and Alice responds affirmatively. Alice progresses successfully through a series of additional explanations and questions. Following an explanation of common differences, Alice is asked a new question (Figure 1a, i). Figure 1a displays the conversation rules that underlie Alice's current question.



(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. Marked blocks (i) – (v) in (a) denote actions taken by a hypothetical user, Alice, in Section 3.1.

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 new question on identifying decreasing arithmetic sequences (Figure 1a, v).

MathBot Front-End Chat

The front end of MathBot is a text chat window between MathBot and the student (Figures 1b and 1c). Students type replies

to MathBot to give answers to problems and provide responses like “I'm not sure”. Students can freely scroll through the chat history to review explanations or questions.

MathBot Back-End Conversation Graph

The MathBot back end consists of a conversation graph that specifies a set of 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., 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.

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 a user answers 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 a user answers 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 and the number of times a user has attempted a particular question. 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 [26] 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 conceptual explanations and corresponding problems while allowing appropriate review of concepts that learners failed to grasp. 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.

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 until they are ready to move on.

MathBot also employs 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! 🟢”).

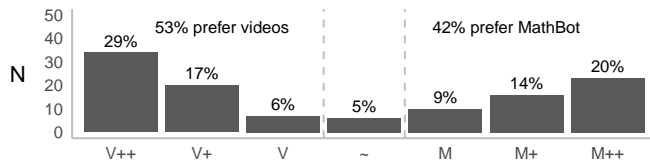
Goal 5: Data-Driven Refinement of Pedagogical Policy

In principle, one can conceive of various pedagogical strategies to achieve the four previous design goals. For example, MathBot could ask students to write out an explanation of their solution to check their understanding (Goal 1) or vary emoji usage to create a supportive atmosphere (Goal 4). It is difficult to know a priori which of these strategies will be most effective upon actual deployment, and whether different strategies should be used for different students. Therefore we set a goal for MathBot to learn to improve continuously by using a data-driven approach to learn pedagogical policies as users begin interacting with it. To consider Goal 3: Guiding Learners’ Review of Concepts, when should MathBot skip the explanation of a concept to a learner or provide an additional practice question? We incorporated a contextual bandit, a reinforcement learning algorithm [31] which conducts adaptive randomized A/B experiments on these design factors (skipping concepts, presenting extra practice). The algorithm automatically identifies which actions are effective for different students and questions in order to provide them more frequently to future users. Study 3 activates this feature of MathBot and evaluates its benefit relative to a traditional randomized A/B experiment to discover which actions are effective.

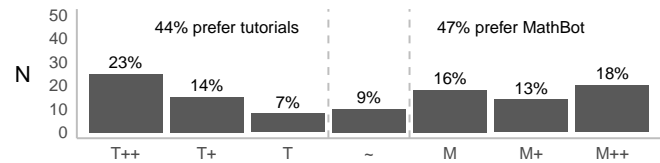
DEVELOPMENT OF RULES IN CONVERSATION GRAPH

In creating MathBot, we iteratively developed a conversational graph covering arithmetic sequences at the level of a high-school Algebra I class. 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 seven Khan Academy videos and four Khan Academy written tutorials on arithmetic sequences.

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 over 10 years—originally defined the graph by adapting explanations, images, and problems from the Khan Academy videos and written tutorials. This author sought to keep MathBot’s content similar to Khan Academy’s content: 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 with MathBot to provide suggestions for improving the conversational graph.



(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. "+" indicates a stronger preference.

User Testing of 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.

STUDY 1: LEARNING PREFERENCES

We begin our evaluation of MathBot by investigating user experiences in a two-part study comparing MathBot to Khan Academy, a high-quality, free, and widely-used online resource for math tutorials and problems that delivers content in a non-conversational format.

Study Design

In the first part of this within-subject study, we ask participants to 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 written tutorials containing embedded practice problems provides an additional layer of insight, as one might conjecture that any result favoring MathBot over video instruction may simply be the result of MathBot providing an interface to work through 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 a 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 quizzes. 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 quiz 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 50% of the questions correctly, reasoning that these individuals already had substantial knowledge of sequences. Users were

paid a bonus proportional to their score on a post-learning quiz. 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. 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 seriously engage with the material. 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.

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 to answer the following question on a 7-point scale ranging from “strongly prefer” MathBot to “strongly prefer” the Khan Academy material: “If you had 60 minutes to learn more about arithmetic sequences and then take a quiz for a large bonus payment, which of these two scenarios would you prefer? 1. Interact with an expanded version of the conversational computer program, then take the quiz. 2. [Watch more videos / Complete more interactive tutorials] about arithmetic sequences, then take the quiz.” We note that the ordering of options 1 and 2 was randomized for each user.

The responses to 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% indicated no preference. The corresponding results for the second part of the study are displayed in Figure 2b. In that case, 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. Overall, more of our participants preferred Khan Academy materials to MathBot—a testament to the popularity of Khan Academy. Our results, however, also illustrate the promise of new forms of instruction to address heterogeneous learning preferences, as a substantial minority of the population appears to prefer the teaching style of MathBot over conventional methods. Indeed, 20% of users in the first part of the study and 18% of users in the second part expressed a “strong preference” for MathBot.

Qualitative Results

After each part of the study, we asked users to respond to the following prompt: “Please compare your experience with the

conversational computer program and the [video / interactive tutorial]. In what scenarios could one learning method be more effective or less effective than the other?" We analyzed the resulting comments to identify themes and understand users' perspectives on MathBot and the Khan Academy videos and written tutorials. One author conducted open coding to identify common themes addressed by each response. Another author verified the coded labels and resolved conflicts with discussion.

Our analysis suggests that MathBot is preferred for learning concepts that require short and simple explanations, and helps keep students focused as they practice and review concepts. On the other hand, videos and written tutorials appear to be preferred when introducing concepts that require more in-depth explanations. A sizable fraction of users also suggested that the ideal learning experience could result from open access to both learning modules. We discuss the coded labels below.

Self-pacing versus guidance. 8 out of 116 users in the first part of the study noted the benefits of freely navigating the video: *"I can rewind them and fast forward if I already know the concept."* Similarly, 22 out of 111 users in the second part of the study indicated value in freely scrolling through the written tutorial: *"I like to learn at my own pace and be able to skip around."* These users frequently indicated frustration with the inability to freely navigate the material in the MathBot conversation: *"With 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, 6 users in the first part of the study 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 15 users in the second part of the study, who preferred that MathBot explicitly guided them through concepts, unlike the written 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."* Furthermore, 8 users in the first part of the study noted value in being able to scroll through the MathBot conversation to review concepts: *"I liked the computer program simply for the ease of being able to scroll back through previous steps."*

Human elements and interactivity. 7 out of 116 users in the first part of the study found MathBot to be more agentic than the video: *"The conversational computer program felt more similar to the experience of interacting with a real teacher."* However, 9 users reported the opposite: *"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."* 12 out of 111 users in the second part of the study indicated that MathBot provided a greater sense of interaction 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, un-

like MathBot, answering these questions correctly was not required. 22 out of 116 users in the first part of the study 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, and 21 out of 111 users in the second part of the study found that being held accountable aided their learning: *"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."* 10 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. 42 users in the first part of the study and 57 users in the second part of the study suggested that both tools could be particularly valuable in specific learning scenarios. For example, 8 users in the first part of the study 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, 16 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."* Relatedly, 25 users in the first part of the study and 15 users in the second part of the study found that videos and written tutorials (respectively) were superior for learning concepts that required complex or detailed explanations: *"The chatbot would be good for learning concepts I am either familiar with or are less complex. The video would be better for more complex math."* We return to the prospect of combining learning modules in the Discussion.

STUDY 2: LEARNING EFFECTIVENESS

Given that a sizable minority of users in Study 1 preferred learning via MathBot and could feasibly benefit from access to it, we next sought to evaluate whether MathBot produced comparable learning gains to Khan Academy material.

Study Design

To assess educational gains, we randomly assigned participants to learn about arithmetic sequences via (1) a full-length MathBot conversation or (2) a combination of Khan Academy videos and written tutorials covering the same content as the MathBot conversation. We assessed learning outcomes with a 12-question quiz, giving the same quiz before and after each participant completed the learning module. Users assigned to Khan Academy had access to seven videos and four written tutorials with embedded practice problems, and they were informed that completing either the videos or the tutorials would sufficiently prepare them for the post-learning quiz. Users were incentivized to complete the learning module to the best

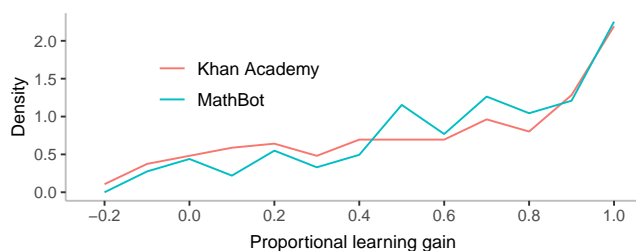


Figure 3: Distributions of proportional learning gain (PLG) for users of MathBot and Khan Academy in Study 2. It appears that both learning methods are similarly effective at conveying information.

of their ability with a bonus payment proportional to their performance on the post-learning quiz. Similar filtering criteria to Study 1 resulted in our analyzing 182 subjects assigned to MathBot and 187 assigned to Khan Academy videos and written tutorials.

Results

We start by computing the proportional learning gain (PLG) for each subject. To calculate PLG, we first determine the raw learning gain by subtracting the pre-learning quiz score from the post-learning quiz score. We divide this result by the maximum possible score increase, defined as the difference between the maximum possible post-learning score (12) and the user’s pre-learning score.

We find the average PLG for MathBot users is 65%, with a 95% confidence interval of [58%, 72%]; the corresponding average PLG for Khan Academy users is 60%, with a 95% confidence interval of [53%, 67%]. This result suggests 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 (two-proportion z-test, $p = 0.3$, 95% CI: [-5%, 15%]).

Figure 3 shows that the distribution of the PLG. Given that MathBot and Khan Academy users spent comparable time completing the learning modules—28 minutes on average for MathBot (SD = 20) and 29 minutes for the Khan Academy videos and written tutorials (SD = 22)—both tools appear to be similarly efficient and effective at conveying information.

STUDY 3: LEARNING A PEDAGOGICAL POLICY

In Study 2, we saw that different users expressed different sentiments about the pacing of the lessons. For example, one participant noted, “*as it gets more complicated, the lesson should slow down a bit,*” while another indicated, “*I felt like the teaching went too slow for me.*”

We sought to address this feedback via personalization, slowing down or speeding up the conversation for each learner as appropriate. Given that the MathBot conversation is structured as a series of lessons, each consisting of a conceptual explanation followed by an assessment question, we could potentially adjust pacing of a lesson in one of four ways: (1) show the conceptual explanation and show an isomorphic practice question before the assessment question (slowest); (2) show the conceptual explanation but skip the isomorphic practice question; (3) skip the conceptual explanation but show the isomorphic

practice question; and (4) skip the conceptual explanation and skip the isomorphic practice question (fastest).

We took a data-driven approach to learning a personalized pedagogical strategy that selects between these four actions for each user and question. We specifically chose to use a *contextual bandit*, a tool from the reinforcement learning literature which balances exploring actions whose payoffs are unclear with exploiting actions whose payoffs are believed to be high [31]. For each user and question, the bandit selects one of the four above actions based on the user’s pre-learning quiz score (the *context*). For example, the algorithm might learn to speed up the conversation for users with high pre-learning quiz scores and slow it down for those with low scores.

To train a contextual bandit, we must not only specify the actions but also the objective function (the *reward*) over which the algorithm will optimize.¹ As a proxy for learning efficiency, we define our reward to be a linear combination of the total time spent on a lesson and an indicator of whether the user gets the assessment question correct on their first try:

$$150 \cdot \mathbf{1}_{\text{correct}} - \text{seconds spent on lesson.}$$

In other words, we assume it is worth 150 seconds of extra time spent on a lesson to turn a student who would have answered the assessment question incorrectly into a student who answered the question correctly. It bears emphasis that the precise form of the reward function should be set by domain experts, but the function we selected seems reasonable for the purposes of our study.

Study Design

Our goal is to assess the value of using a contextual bandit to learn a personalized pedagogical strategy for students. We benchmark the bandit to a common alternative: a regression fit on data from users who were randomly assigned to one of the four possible actions before each assessment question. That is, in the benchmark approach, we first conduct an *exploration* phase, in which we assign users to the four actions uniformly at random; then, we fit a regression on the collected data to learn a personalized policy. The bandit, in contrast, aims to better manage exploration by down-weighting actions that are learned to be ineffective.

To carry out this comparison, we first recruited 30 participants from Amazon Mechanical Turk and assigned them to each of the four actions at random, independently for each question. Data from this pilot study were used to provide the bandit a warm start. We then randomly assigned the remaining participants to either: (1) the contextual bandit condition; or (2) the uniform random condition.

¹To train the bandit, we use linear model with Thompson Sampling, a technique known to have strong empirical performance and theoretical guarantees [1]. We model the reward using ordinary least regression (OLS) where the covariates are the contextual variables, the actions, and the two-way terms between the contextual variables and the actions. Then, we simply choose action a with a probability proportional to its posterior likelihood of being the best action.

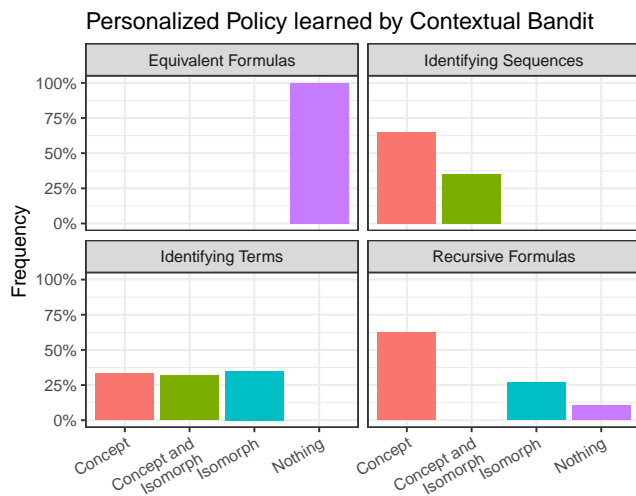


Figure 4: For four representative assessment questions out of the eleven total, the proportion of users for which the final policy learned by the bandit would use each action. The policy chooses different actions based on each user’s pre-learning quiz score.

Participants

We use the same criteria as in Study 2 to filter participants before they interact with MathBot. These filtering criteria resulted in our analyzing 228 subjects assigned to MathBot with a uniform random policy, and 239 assigned to MathBot with a contextual bandit policy. We note that both groups include the 30 participants from the pilot study.

Results

We examine the behavior of the contextual bandit algorithm along three dimensions: (1) its degree of personalization; (2) the quality of the final learned pedagogical policy; and (3) the cost of exploration. We found that the bandit learned a personalized policy that was comparable in quality to the one learned on the uniform random data, but, importantly, the bandit did so while incurring less cost on users.

Personalization. We begin by examining the pedagogical policy ultimately learned by the contextual bandit (i.e., the policy the bandit believed to be the best at the end of the experiment, after seeing 239 participants). Averaged over all questions, the final, learned policy assigns approximately 30% of users to the concept-only, isomorph-only, and no-concept-no-isomorph conditions; the remaining 10% are assigned to the concept-plus-isomorph condition.² In Figure 4, we disaggregate the action distribution by question, showing the result for 4 representative questions out of the 11. The plot shows that the bandit is indeed learning a policy that differs substantially across users and questions. For only 3 of the 11 questions does the bandit determine that it is best to use the same action for every user—though even in these cases, the selected actions differ across questions.

Quality of learned solution. Next, we compare the expected reward of the learned policy from the bandit to that of the

²These numbers do not represent the distribution of actions actually assigned during the experiment, but rather the distribution of actions under the policy the bandit ultimately learned.

learned policy from the uniform random condition.³ For the uniform random condition, we consider three different regression models: (1) a model with two-way interactions between actions and questions (effectively learning a constant policy per question); (2) a model with the same specification as the bandit, which is able to personalize based on pre-learning quiz score; and (3) a lasso regression that includes eight contextual covariates, including pre-learning quiz score and accuracy on the previous question. Of these three models, the first performs the best.

We find that the bandit learned a policy which is comparable to the most successful policy learned from the uniform random condition. In particular, we find no statistically significant difference between the average reward obtained by the final bandit policy and the policy learned from the uniform random data. A 95% confidence interval for this difference in average rewards is [-11, 28], slightly in favor of the policy learned in the uniform random condition.

The cost of exploration. The above results indicate that one can indeed learn a personalized pedagogical policy using a contextual bandit that is on par with one learned from uniform random data. The primary value of a bandit, however, is that it incurs lower costs of exploration, by quickly learning which actions are unlikely to be beneficial. We thus now directly compare the average rewards obtained under the bandit and uniform random conditions *during* the model-learning period. Higher average reward during model-learning suggests users are having a better experience, as they receive sub-optimal actions less often.

We first compute the average reward for each lesson in each condition, and then average that quantity over all the lessons for each condition. This gives us the the average reward per lesson per user for each condition. As shown in Figure 5 (left panel), the average reward in the contextual bandit condition is substantially higher than in the uniform random condition. A 95% confidence interval on the difference is [9.6, 29.1].⁴

As another way to assess the cost of exploration, we compute the average value of a *global* reward function across users in our two conditions—bandit and uniform random. Analogous

³We never observe the actual outcomes of implementing these policies, as that would require running another costly experiment. We instead make use of standard offline policy evaluation techniques to compare the pedagogical strategies learned by the bandit and the uniform random experiment [32]. The specific quantity we are interested in is the expected average reward for a random user drawn from the population distribution. To compute the expected average reward for this policy on question i , we evaluate the average reward on question i in our uniform random condition where the user was randomly assigned into the action our bandit policy would have chosen. We then average these rewards over all the questions and compute standard errors by bootstrapping the uniform random data. We perform a similar method for the policies trained on the uniform random data, except we choose an action for person p using a model trained on all the uniform random data except those of person p to avoid overfitting.

⁴We compute the standard errors for our uniform random condition through the bootstrap. The standard errors for the bandit condition are obtained through creating a response surface model fit on the uniform random data and running simulations.

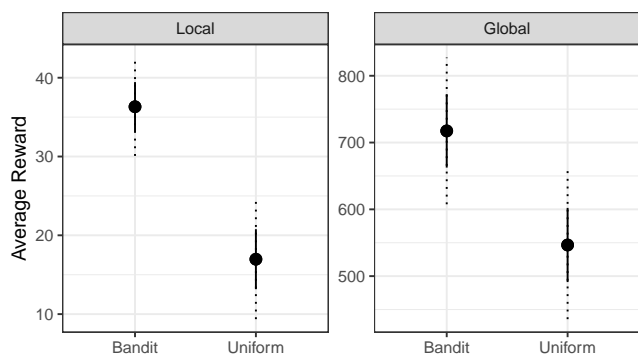


Figure 5: Average local (left) and global (right) rewards during the experiment for the bandit and uniform random conditions. In both cases, the contextual bandit obtains higher rewards suggesting that it provides improved user experience while learning an optimal pedagogical policy.

to the local reward function, the global reward is defined as:

$150 \cdot \text{Post-learning Quiz Score}$ – seconds spent on MathBot.

In contrast to the local reward function, the global reward considers the total post-learning quiz score and total time spent on the entire MathBot conversation, rather than correctness and time spent during individual lessons.

Figure 5 (right panel) shows the average global rewards of participants between the two conditions. We find that the bandit obtains considerably higher average global rewards than the uniform random condition. a 95% confidence interval on the difference is [17.7, 323.9]. We note that the bandit was only designed to optimize for local rewards, so this result offers further proof that the bandit is learning a generally effective policy.

As a final way to assess user satisfaction during exploration, we examine the difference in dropout rates between the two conditions. A user is said to “drop out” if they complete the first MathBot lesson but not the final lesson, either skipping to the post-learning quiz or leaving the experiment. Out of the participants in the bandit condition, 9% dropped out, compared to 15% of participants in the uniform random condition—a statistically significant gap of 6 percentage points. This result again suggests the bandit provides an improved user experience while learning a pedagogical policy.

DISCUSSION, LIMITATIONS, & FUTURE WORK

Although the content of MathBot closely matched that of the Khan Academy written tutorials and videos, we found evidence of heterogeneous learning preferences. About one-fifth of our study participants expressed a strong preference for MathBot, and about one-quarter expressed a strong preference for the Khan Academy materials. Both modes of instruction appear to be similarly effective; MathBot produced learning gains that were somewhat higher, though the gap was not statistically significant. Finally, we found that a contextual bandit was able to efficiently learn a personalized pedagogical policy for showing extra practice problems and skipping explanations to appropriately alter the pace of conversation.

Particularly noteworthy is that MathBot provided a relatively successful conversational experience *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 [21, 24, 37, 39, 52]. 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, 33, 51]. A valuable future direction could be to integrate user interface insights from MathBot with existing conversational tutors [19, 23, 37], which have only recently begun to be applied to math [20, 21, 24].

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 can participate in development. Just as teachers put extensive time into creating curricula, future work could explore whether the general approach instantiated in MathBot’s conversation graph could help teachers create such conversational programs. In addition, Study 3 demonstrated the successful use of contextual bandit algorithms to learn how to personalize elements of the conversation graph — when to skip conceptual explanations and give additional practice problems. This provides one demonstration of how such components could be learned after deployment, further minimizing the development time.

A key limitation of our study is that we evaluated MathBot using a convenience sample of adults from Amazon Mechanical Turk. While Mechanical Turk workers exhibit similar video-watching behavior and quiz performance as MOOC learners [16], 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. Since MathBot’s applicability to a classroom setting is yet to be explored, future work can consider 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?

Additionally, our system taught a single algebra topic, arithmetic sequences, with a conversation intended to last approximately 30 minutes (Studies 2 and 3) and could be less than 10 minutes (Study 1). Some of our insights will likely 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.

Finally, we note two 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 [48]. Accordingly, one could explore integrating brief conversations with MathBot into educational videos or, conversely, video elements

could be used in the MathBot conversation. Second, though MathBot interactively guides learners through explanations and relevant questions, it does not provide a platform for extensive rote practice after finishing the conversation. An adaptive question sequencing model such as DASH [34] could be used to guide students through an optimized sequence of practice problems by accounting for student performance during the MathBot conversation.

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, without using algorithms to process open-ended natural language dialogue. We obtained qualitative evidence that many users found the conversational experience to be engaging and gave a sense of the back-and-forth interaction with a tutor. A sizable minority of users preferred to learn using MathBot over videos and tutorials with embedded problems. They appreciated the appropriate interleaving of multiple components: asking questions to check comprehension, tailoring feedback to incorrect answers, and guiding learners to review concepts or answer easier or harder questions. Finally, a contextual bandit was able to efficiently learn a personalized pedagogical strategy for adjusting the pace of the interactive tutor. Learning math through conversation appears to be at least as effective 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. Shipra Agrawal and Navin Goyal. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28.
2. 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).
3. Vincent Aleven, Bruce M McLaren, and Jonathan Sewall. 2009a. 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.
4. Vincent Aleven, Bruce M McLaren, Jonathan Sewall, and Kenneth R Koedinger. 2009b. *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. DOI : <http://dx.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. Chat-Bot 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? (2014). DOI : http://dx.doi.org/10.1007/978-3-319-07221-0_25
11. Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. In *User Modeling and User-Adapted Interaction*. 137–180.
12. 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.
13. Benjamin Clement, Pierre-Yves Oudeyer, Didier Roy, and Manuel Lopes. 2015. Multi-Armed Bandits for Intelligent Tutoring Systems. 7, 2 (2015), 20–48.
14. 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. DOI : <http://dx.doi.org/10.1016/j.compedu.2013.06.010>
15. 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.
16. Dan Davis, Claudia Hauff, and Geert-Jan Houben. 2018. Evaluating Crowdworkers as a Proxy for Online Learners in Video-Based Learning Contexts. In *Proceedings of the ACM on Human-Computer Interaction*. ACM, 42:1–42:16.

17. Jean-Claude Falmagne, Dietrich Albert, Christopher Doble, David Eppstein, and Xiangen Hu. 2013. *Knowledge spaces: Applications in education*. Springer Science & Business Media.
18. 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.
19. 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. DOI : <http://dx.doi.org/10.3758/BF03195563>
20. 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. DOI : http://dx.doi.org/10.1207/s15326985ep4004_4
21. Arthur C. Graesser, Phanni Penumatsa, Matthew Ventura, Zhiqiang Cai, and Xiangen Hu. 2007. Using LSA in AutoTutor: Learning through mixed initiative dialogue in natural language. *Handbook of latent semantic analysis* (2007), 243–262.
22. 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. DOI : <http://dx.doi.org/10.1002/acp.2350090604>
23. 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. DOI : <http://dx.doi.org/10.1609/AIMAG.V22I4.1591>
24. 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. DOI : [http://dx.doi.org/10.1016/S1389-0417\(99\)00005-4](http://dx.doi.org/10.1016/S1389-0417(99)00005-4)
25. 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.
26. 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.
27. 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.
28. Kenneth R Koedinger, Vincent Alevan, Neil Heffernan, Bruce McLaren, and Matthew Hockenberry. 2004. *Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration*. Technical Report. www.carnegielearning.com
29. Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.
30. Andrew S Lan and Richard G Baraniuk. 2016. A Contextual Bandits Framework for Personalized Learning Action Selection. In *Proceedings of the 9th International Conference on Educational Data Mining*. 424–429.
31. Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *WWW 2010*. 661–670.
32. Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2012. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *JMLR*. 19–36.
33. 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. DOI : <http://dx.doi.org/10.1016/J.COMPEDU.2010.09.019>
34. Robert V. Lindsey Michael C. Mozer. 2016. Predicting and Improving Memory Retention: Psychological Theory Matters in the Big Data Era. *Big Data in Cognitive Science* (2016), 43–73.
35. 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. DOI : <http://dx.doi.org/10.1145/3027063.3027077>
36. 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. DOI : <http://dx.doi.org/10.1007/s40593-014-0029-5>
37. 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. DOI : <http://dx.doi.org/10.1186/s40594-018-0109-4>
38. Eleanor O'Rourke, Erik Andersen, Sumit Gulwani, and Zoran Popović. 2015. A Framework for Automatically Generating Interactive Instructional Scaffolding. (2015). DOI : <http://dx.doi.org/10.1145/2702123.2702580>

39. 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.
40. Silvia Quarteroni and Suresh Manandhar. 2007. A chatbot-based interactive question answering system. *Decalog 2007* (2007), 83.
41. 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.
42. 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.
43. 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. DOI: <http://dx.doi.org/10.3758/BF03194060>
44. 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.
45. Sherry Ruan, Liwei Jiang, Justin Xu, Bryce Joe-Kun Tham, Zhengneng Qiu, Yeshuang Zhu, Elizabeth L Murnane, Emma Brunskill, and James A Landay. 2019. QuizBot: A Dialogue-based Adaptive Learning System for Factual Knowledge. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.
46. Avi Segal, Yossi Ben David, Joseph Jay Williams, Kobi Gal, and Yaar Shalom. 2018. Combining Difficulty Ranking with Multi-Armed Bandits to Sequence Educational Content. In *International Conference on Artificial Intelligence in Education*. 317–321.
47. Stephanie Seneff. 1992. TINA: A natural language system for spoken language applications. *Computational linguistics* 18, 1 (1992), 61–86.
48. 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.
49. Richard E Snow. 1989. Aptitude-treatment interaction as a framework for research on individual differences in learning. In *A series of books in psychology. Learning and individual differences: Advances in theory and research*. 13–59.
50. Kurt VanLehn. 1996. Conceptual and meta learning during coached problem solving. (1996), 29–47. DOI: http://dx.doi.org/10.1007/3-540-61327-7_99
51. 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. DOI: <http://dx.doi.org/10.1080/03640210709336984>
52. Kurt VanLehn, Pamela W Jordan, Carolyn P Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, and others. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *International Conference on Intelligent Tutoring Systems*. Springer, 158–167.
53. 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.
54. 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.
55. 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.
56. 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.
57. Guojing Zhou, Jianxun Wang, Collin Lynch, and Min Chi. 2017. Towards Closing the Loop: Bridging Machine-induced Pedagogical Policies to Learning Theories. In *Proceedings of the 10th International Conference on Educational Data Mining*. 112–119.