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.

Introduction

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. Finally, MathBot provides learners with personalized feedback and guidance.

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?" (Graesser et al. 2007; 2001; 2004; Nye, Graesser, and Hu 2014; Ramachandran et al. 2007; Graesser et al. 1999). These types of conversational tutors have been shown to benefit student learning (McLaren, DeLeeuw, and Mayer 2011; VanLehn et al. 2007; Chi et al. 2014; Craig et al. 2013), 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. 2018 for first steps).

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 complete either 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.

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 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.*

*These authors contributed equally to the paper, and are ordered alphabetically by last name.

The bulb lights. Why?) and initiating a discussion based on the responses. AutoTutor and its derivatives (Nye, Graesser, and Hu 2014; VanLehn et al. 2002; Graesser et al. 1999; 2004) arose from Graesser, Person, and Magliano's investigation of human tutoring behaviors (1995) 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 (VanLehn et al. 2002), LSA (Graesser et al. 1999; Person 2003), and other semantic analysis tools (Graesser et al. 2007). Nye et al. added conversational routines to the online mathematics intelligent tutoring system ALEKS by attaching mini-dialogues to individual problems, but leaving navigation on the website (2018). 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 (ITSs) in mathematics use precise models of students' mathematical knowledge and misunderstandings (Ritter et al. 2007; VanLehn 1996; Aleven et al. 2009a; 2009b; Aleven, McLaren, and Sewall 2009; O'Rourke et al. 2015). To reduce the time and expertise needed to build ITSs, some researchers have proposed example-tracing tutors (Koedinger et al. 2004; Aleven et al. 2009a; 2016). 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 (Aleven et al. 2016). With the help of 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 (Aleven et al. 2009a; 2009b; Aleven, McLaren, and Sewall 2009). 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 (Xu et al. 2017), college management (Bala et al. 2017), and purchase recommendation (Horzyk, Magierski, and Miklaszewski 2009). One approach to building a chatbot is to construct rule-based input to output mappings (Al-Rfou et al. 2016; Yan, Song, and Wu 2016). One can also embed chatbot dialogue into a higher-level structure (Bobrow and Winograd 1977) to keep track of the current state of the conversation, move fluidly between topics, and collect context for later use (Walker and Whittaker 1990; Seneff 1992; Chu-Carroll and Brown 1997). 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 (Raux and Eskenazi 2009; Quarteroni and Manandhar 2007; Andrews et al. 2006), where user responses update the conversation state according to a preset transition graph.

MathBot System Design & Development

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 how to progress through concepts and what actions to take based on user responses; (3) elaborate on the design goals of the system; (4) explain the development process and collection of user data that was used to create the rules in MathBot's conversational graph.

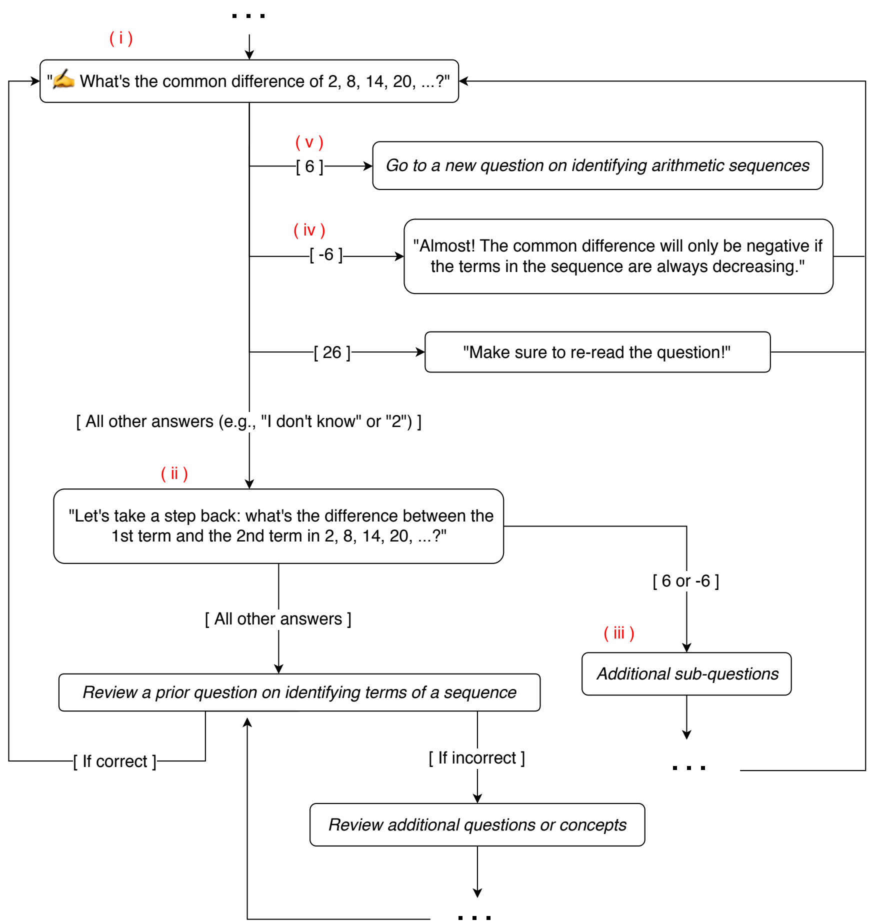
Sample Learner Interaction with MathBot

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.

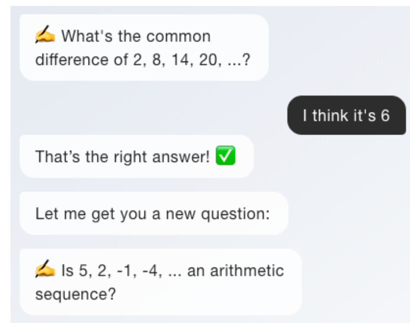
When asked the new question, Alice confuses the term "common difference" with "greatest common factor", a topic she recently reviewed, so she answers "2". MathBot recognizes that Alice has made a mistake and subsequently checks that she knows how to identify terms in a sequence and subtract them, a prerequisite 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).

MathBot Front-End Chat

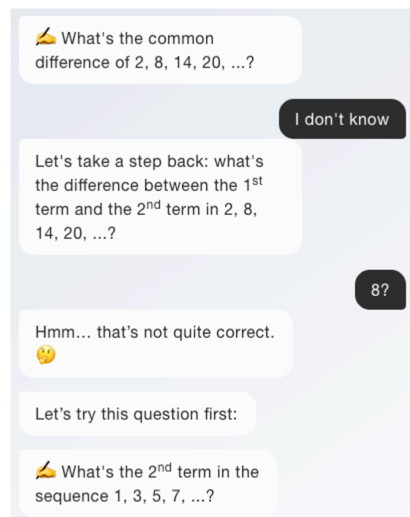
The front end of MathBot is a text chat window between MathBot and the student (see Figures 1b and 1c). Students are presented with short lessons, asked to solve math problems, and shown explanations. Students type replies to MathBot into the chat to give answers to problems and provide responses like "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.



(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.

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.

Goal 1: Checking Understanding

The first goal of MathBot is to use conversational questions to continually check users' understanding while they are learning. 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.

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 (Heffernan and Koedinger 2002) 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.

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 uses a friendly tone, provides supportive cues such as transition phrases and emoji, and exhibits natural typing patterns. Correct and incorrect feedback 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 (✍️).

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. 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 with MathBot to provide suggestions for improving the conversational graph.

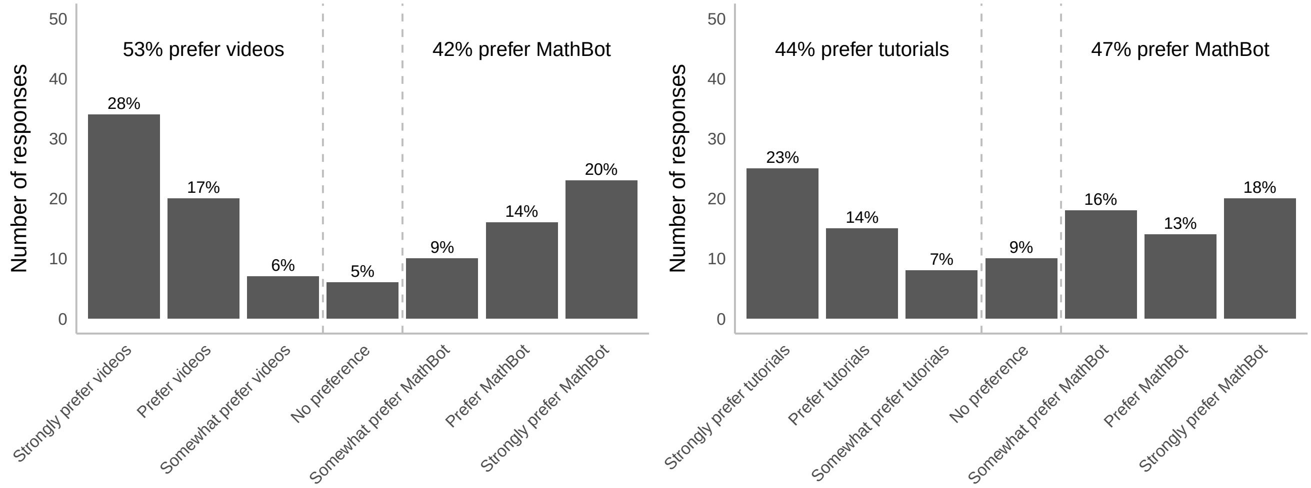
Study 1: Learning preferences

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.

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.



(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.

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.

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.

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.

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 content.

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 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]).

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.

Discussion & Limitations

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. MathBot wasn't any 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 (VanLehn et al. 2002; Graesser et al. 1999; Person 2003; Graesser et al. 2007; Nye et al. 2018). 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 (McLaren, DeLeeuw, and Mayer 2011; Chi et al. 2014; VanLehn et al. 2007). A valuable future direction could be to integrate user interface insights from MathBot with existing conversational tutors (Graesser et al. 2004; 2001; Nye et al. 2018), which have only recently begun to be applied to math (Graesser et al. 1999; 2007; Graesser, McNamara, and VanLehn 2005).

A key limitation of our study is that we evaluated MathBot using a convenience sample of adults from 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 may 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.

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.

Acknowledgements

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.

[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.

References

Al-Rfou, R.; Pickett, M.; Snaider, J.; Sung, Y.-h.; Strope, B.; and Kurzweil, R. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*.

Aleven, V.; Mclaren, B. M.; Sewall, J.; and Koedinger, K. R. 2009a. A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. Technical report.

Aleven, V.; Mclaren, B. M.; Sewall, J.; and Koedinger, K. R. 2009b. A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. Technical report.

Aleven, V.; McLaren, B. M.; Sewall, J.; van Velsen, M.; Popescu, O.; Demi, S.; Ringenberg, M.; and Koedinger, K. R. 2016. Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers. *International Journal of Artificial Intelligence in Education* 26(1):224–269.

Aleven, V.; McLaren, B. M.; and Sewall, J. 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):64–78.

Andrews, P.; De Boni, M.; Manandhar, S.; and De, M. 2006. Persuasive Argumentation in Human Computer Dialogue. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*, 8–13.

Bala, K.; Kumar, M.; Hulawale, S.; and Pandita, S. 2017. Chat-bot for college management system using ai. *International Research Journal of Engineering and Technology*.

Bobrow, D. G., and Winograd, T. 1977. An overview of krl, a knowledge representation language. *Cognitive science* 1(1):3–46.

Chi, M.; Jordan, P. W.; Vanlehn, K.; and Jordan, P. 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?

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

Craig, S. D.; Hu, X.; Graesser, A. C.; Bargagliotti, A. E.; Sterbinsky, A.; Cheney, K. R.; and Okwumabua, T. 2013. The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors. *Computers & Education* 68:495–504.

Graesser, A. C.; Wiemer-Hastings, K.; Wiemer-Hastings, P.; and Kreuz, R. 1999. AutoTutor: A simulation of a human tutor. *Cognitive Systems Research* 1(1):35–51.

Graesser, A. C.; VanLehn, K.; Rose, C. P.; Jordan, P. W.; and Harter, D. 2001. Intelligent Tutoring Systems with Conversational Dialogue. *AI Magazine* 22(4):39–39.

Graesser, A. C.; Lu, S.; Jackson, G. T.; Mitchell, H. H.; Ventura, M.; Olney, A.; and Louwerse, M. M. 2004. AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers* 36(2):180–192.

Graesser, A. C.; Penumatsa, P.; Ventura, M.; Cai, Z.; and Hu, X. 2007. Using lsa in autotutor: Learning through mixed initiative dialogue in natural language. *Handbook of latent semantic analysis* 243–262.

Graesser, A. C.; McNamara, D. S.; and VanLehn, K. 2005. Scaffolding Deep Comprehension Strategies Through Point&Query, AutoTutor, and iSTART. *Educational Psychologist* 40(4):225–234.

Graesser, A. C.; Person, N. K.; and Magliano, J. P. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology* 9(6):495–522.

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

Horzyk, A.; Magierski, S.; and Miklaszewski, G. 2009. An intelligent internet shop-assistant recognizing a customer personality for improving man-machine interactions. *Recent Advances in intelligent information systems* 13–26.

Koedinger, K. R.; Aleven, V.; Heffernan, N.; Mclaren, B.; and Hockenberry, M. 2004. Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. Technical report.

McLaren, B. M.; DeLeeuw, K. E.; and Mayer, R. E. 2011. Polite web-based intelligent tutors: Can they improve learning in classrooms? *Computers & Education* 56(3):574–584.

Nye, B. D.; Pavlik, P. I.; Windsor, A.; Olney, A. M.; Hajeer, M.; and Hu, X. 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):12.

Nye, B. D.; Graesser, A. C.; and Hu, X. 2014. AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education* 24(4):427–469.

O'Rourke, E.; Andersen, E.; Gulwani, S.; and Popovic, Z. P. 2015. A Framework for Automatically Generating Interactive Instructional Scaffolding.

Person, N. K. 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* 97:47.

Quarteroni, S., and Manandhar, S. 2007. A chatbot-based interactive question answering system. *Decalog 2007* 83.

Ramachandran, R.; Movva, S.; Li, X.; Anantharam, P.; and Graves, S. 2007. Wxguru: An ontology driven chatbot prototype for atmospheric science outreach and education. *Geoinformatics* 17–18.

Raux, A., and Eskenazi, M. 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*, 629–637. Association for Computational Linguistics.

Ritter, S.; Anderson, J. R.; Koedinger, K. R.; and Corbett, A. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review* 14(2):249–255.

Seneff, S. 1992. Tina: A natural language system for spoken language applications. *Computational linguistics* 18(1):61–86.

VanLehn, K.; Jordan, P. W.; Rosé, C. P.; Bhembe, D.; Böttner, M.; Gaydos, A.; Makatchev, M.; Pappuswamy, U.; Ringenberg, M.; Roque, A.; et al. 2002. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *International Conference on Intelligent Tutoring Systems*, 158–167. Springer.

VanLehn, K.; Graesser, A. C.; Jackson, G. T.; Jordan, P.; Olney, A.; and Rosé, C. P. 2007. When Are Tutorial Dialogues More Effective Than Reading? *Cognitive Science* 31(1):3–62.

VanLehn, K. 1996. Conceptual and meta learning during coached problem solving. 29–47.

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

Xu, A.; Liu, Z.; Guo, Y.; Sinha, V.; and Akkiraju, R. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3506–3510. ACM.

Yan, R.; Song, Y.; and Wu, H. 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*, 55–64. ACM.