Ruffle&Riley: Towards the Automated Induction of Conversational Tutoring Systems

Robin Schmucker
Machine Learning Department
Carnegie Mellon University
rschmuck@cs.cmu.edu

Meng Xia
Human-Computer Interaction Institute
Carnegie Mellon University
mengxia@cs.cmu.edu

Amos Azaria
Department of Computer Science
Ariel University
amos.azaria@ariel.ac.il

Tom Mitchell
Machine Learning Department
Carnegie Mellon University
tom.mitchell@cs.cmu.edu

Abstract

Conversational tutoring systems (CTSs) offer learning experiences driven by natural language interaction. They are known to promote high levels of cognitive engagement and benefit learning outcomes, particularly in reasoning tasks. Nonetheless, the time and cost required to author CTS content is a major obstacle to widespread adoption. In this paper, we introduce a novel type of CTS that leverages the recent advances in large language models (LLMs) in two ways: First, the system induces a tutoring script automatically from a lesson text. Second, the system automates the script orchestration via two LLM-based agents (Ruffle&Riley) with the roles of a student and a professor in a learning-by-teaching format. The system allows a free-form conversation that follows the ITS-typical inner and outer loop structure. In an initial between-subject online user study (N = 100) comparing Ruffle&Riley to simpler QA chatbots and reading activity, we found no significant differences in post-test scores. Nonetheless, in the learning experience survey, Ruffle&Riley users expressed higher ratings of understanding and remembering and further perceived the offered support as more helpful and the conversation as coherent. Our study provides insights for a new generation of scalable CTS technologies.

1 Introduction

Intelligent tutoring systems (ITSs) are a type of educational technology that provides millions of learners worldwide with access to learning materials and affordable personalized instruction. ITSs can, in certain cases, be as effective as human tutors [61, 32] and can play an important role in mitigating the educational achievement gap [53, 25]. However, despite their potential, one major obstacle to the widespread adoption of ITS technologies is the large costs associated with content development. Depending on the depth of instructional design and available authoring tools, preparing one hour of ITS content can take instructional designers hundreds of hours [3].

Conversational tutoring systems (CTSs) are a type of ITS that engages with learners in natural language. Various studies have confirmed the benefits of CTSs, across multiple domains, particularly on learning outcomes in reasoning tasks [47]. Still, many existing CTSs struggle to maintain coherent free-form conversations and understand the learners' responses due to limitations imposed by their underlying natural language processing (NLP) techniques [19]. In this paper, we introduce a new type of CTS that draws inspiration from design principles of earlier CTSs [45, 35] and that leverages

NeurIPS'23 Workshop on Generative AI for Education (GAIED).

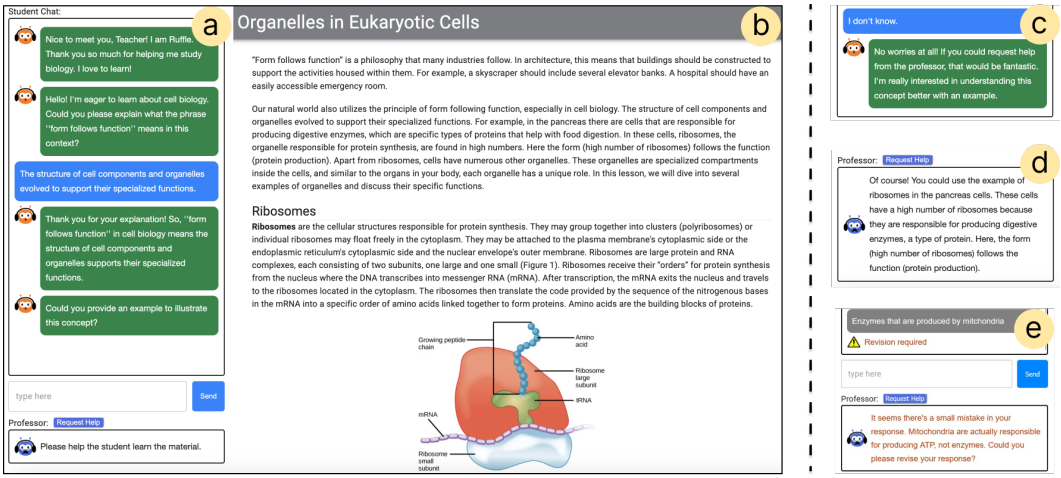

Figure 1: UI of Ruffle&Riley. (a) Learners are asked to teach Ruffle (student agent) in a free-form conversation and request help as needed from Riley (professor agent). Ruffle tries to guide the learner to articulate the expectations in the tutoring script. (b) The learner can navigate the lesson material during the conversation. (c) Ruffle encourages the learner to explain the content. (d) Riley responds to a help request. (e) Riley detected a misconception and prompts the learner to revise their response.

the recent advances in large language models (LLMs) [67] to automate content authoring and to orchestrate free-form conversational tutoring. Our main contributions include:

- *Automated Induction of a Conversational Tutoring System from Text:* We introduce a new type of CTS that employs LLMs to generate a tutoring script automatically from a lesson text and that further automates the script orchestration in a free-form conversation. In particular, we orchestrate the conversation in a learning-by-teaching format via two conversational agents taking on the roles of a student (Ruffle) and a professor (Riley). The conversation follows the prototypical ITS structure by exhibiting an outer loop (problem sequencing) and an inner loop (feedback/assistance) [60].

- *Findings from an Online User Study* We report findings from a user study that evaluates the effects of our LLM-induced CTS workflow on learning outcomes and user experience, comparing it to two simpler QA chatbots and reading activity. We discuss the strengths and limitations of our current system, describe our plans for system refinements in response to user feedback, and provide guidance for the design and evaluation of future LLM-based CTSs.

2 Related Work

2.1 Conversational Tutoring Systems

Dialog-based learning activities are known to lead to high levels of cognitive engagement [10], and various studies have confirmed their benefits on learning outcomes (e.g., [12, 9]). This motivated the integration of conversational activities into learning technologies. In their systematic review, Paladines and Ramirez [47] categorized the design principles underlying existing CTSs into three major categories: (i) expectation misconception tailoring (EMT) [15, 45], (ii) model-tracing (MT) [54, 52, 23]) and (iii) constraint-based modeling (CBM) [40, 62]. While all three frameworks can promote learning, they require instructional designers to spend substantial effort configuring the systems for each individual lesson and domain. Further, due to limitations of underlying NLP techniques, many CTSs struggle to maintain coherent free-form conversations, answer learners' questions, and understand learners' responses reliably [45]. In this context, this paper employs recent NLP advances as the foundation for a novel type of LLM-driven CTSs that can orchestrate coherent free-form adaptive dialogues and that can alleviate the burdens associated with content authoring.

2.2 Content Authoring Tools

One major obstacle to the widespread adoption of CTSs and other types of ITSs is the complexity and cost of content authoring [2, 19, 14]. For early ITSs, the development ratio (i.e., the number of hours required to author one hour of instructional content) was estimated to vary between 200:1 and 300:1 [2]. Content authoring tools (CATs) [42] were developed to facilitate ITS creation, often with an emphasis on making the process accessible to educators without programming background (e.g., [29, 65, 3]). While a comprehensive survey of CATs is vastly beyond the scope of this paper–for this, we refer to [14, 57]–here we focus on highlighting prior studies that illustrate the ability of existing CATs to reduce authoring times. ASSISTment Builder [51] was developed to support content authoring in a math ITS and enabled a development ratio of 40:1. For model tracing-based ITSs, example tracing [3] has proven itself as an effective authoring technique that depending on the context enables development ratios between 50:1 and 100:1. Recently, apprentice learner models were evaluated as another authoring technique that in certain cases can be more efficient than example tracing [63, 38]. In the context of CTSs, multiple CATs have been developed for AutoTutor [7], and while we were not able to find concrete development ratio estimates, the authoring of CTS content is still considered to be complex and labor intensive.

Alternative approaches explored the use of learner log data to enhance ITS components such as skill models and hints (e.g., [4, 8, 5]) as well as machine learning-based techniques for automated questions and feedback generation (e.g., [33, 22]). Recent advances in large language models (LLMs) [67] sparked a new wave of research that explores ways in which LLM-based technologies can benefit learners [28]. Settings in which LLMs already have been found to be effective include question generation and quality assessment [55, 43, 41, 26, 1], feedback generation [55, 27, 44, 36, 48, 68, 1], answering students' questions [34, 56], automated grading [24, 6], and helping teachers reflect on their teaching [39, 37, 13]. What sets this paper apart from the aforementioned works is that it does not focus on the generation of *individual* ITS components; instead, we propose a system that can automatically induce a *complete ITS workflow*, exhibiting the prototypical inner and outer loop structure [60], directly from a lesson text. Our work represents a step towards LLM-driven ITS authoring tools that can generate entire workflows automatically from existing learning materials and reduce system development times by an order of magnitude potentially.

3 System Architecture

Design Considerations

We approached the design of Ruffle&Riley with two specific goals in mind: (i) Facilitate an ITS workflow that provides learners with a sequence of questions (outer loop) and meaningful feedback during problem-solving (inner loop); (ii) Streamline the process of configuring the conversational agents for different lesson materials. We reviewed existing CTSs and identified EMT [20] as a design framework suitable for our objectives. EMT mimics teaching strategies employed by human tutors [21] by associating each question with a list of expectations and anticipated misconceptions. After presenting a question and receiving an initial user response, EMT-based CTSs provide inner loop support (goal (i)) by guiding the conversation via a range of dialogue moves to correct misconceptions and to help the learner articulate the expectations before moving to the next question (outer loop). While EMT-based CTSs have been shown to be effective in various domains [45], they need to be configured in a labor-intensive process that requires instructional designers to define a *tutoring script* that specifies questions, expectations, misconceptions and other information for each lesson [7]. For us, tutoring scripts are attractive as a standardized format for CTS configuration (goal (ii)).

An overview of our user interface, together with descriptions of its key elements, is provided by Figure 1. Inspired by the success of learning-by-teaching activities [16, 18, 35], we decided to orchestrate the conversation in a learning-by-teaching format via two conversational agents taking on the roles of a student (Ruffle) and a professor (Riley). While our design is similar to some CTSs in the AutoTutor family [45] that follow a trialogue format, one notable difference is that Riley solely serves as an assistant to the learner by offering assistance and correcting misconceptions. Riley never communicates with Ruffle directly. In the following, we describe the system architecture underlying Ruffle&Riley in more detail (Figure 2).

Tutoring Script Generation Ruffle&Riley is capable of generating a tutoring script fully automatically from a lesson text by leveraging GPT4 [46]. This involves a 4-step process: (i) A list of review

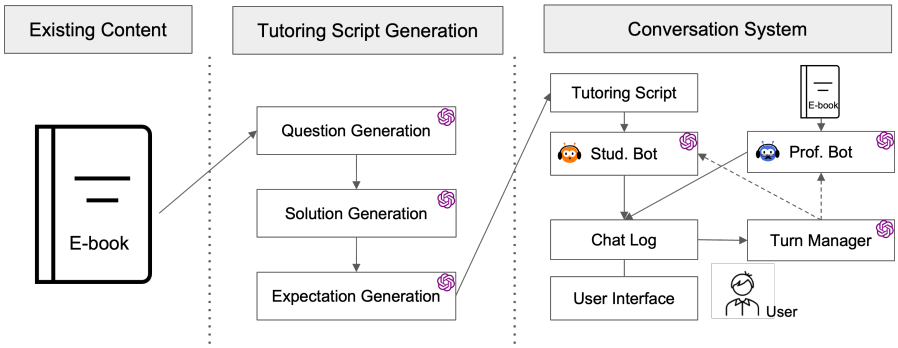

Figure 2: System architecture. Ruffle&Riley generates a *tutoring script* automatically from a lesson text by executing three separate prompts that induce questions, solutions and expectations for the EMT-based dialog. During the learning process, the script is orchestrated via two LLM-based conversational agents in a free-form dialog that follows the ITS-typical outer and inner loop structure.

questions is generated from the lesson text; (ii) For each question, a solution is generated based on question and lesson texts; (iii) For each question, a list of expectations is generated based on question and solution texts; (iv) The final tutoring script is compiled as a list of questions together with related expectations. The first three steps are implemented via three separate prompts written in a way general enough to support a wide range of lesson materials. Unlike traditional EMT-based CTSs, our tutoring scripts do not attempt to anticipate misconceptions learners might exhibit ahead of time (this is a difficult task even for human domain experts). Instead, we rely on GPT4's ability to detect factually incorrect information in the learner's responses during the active teaching process.

Conversation Orchestration EMT-based CTSs require the definition of dialog moves and conversational turn management to facilitate coherent conversations, which in itself is a complex authoring process [7]. Ruffle&Riley automates the tutoring script orchestration by including descriptions of desirable properties of EMT-based conversations into the agents' prompts and captures the user's state solely via the chat log. The student agent receives the tutoring script as part of its prompt and is instructed to let the user explain the individual questions and to ask follow-ups until all expectations are covered. Ruffle reflects on user responses to show understanding, provides encouragement to the user, and keeps the conversation on topic. In parallel, Riley's prompt contains the lesson text and instructions to provide feedback to user responses, to offer relevant information after help requests, and to prompt the user to revise their response after detecting incorrect information. Both agents are instructed to keep the conversation positive and encouraging and to not refer to information outside the tutoring script/lesson text. The turn manager coordinates the system's queries to GPT4.

4 System Evaluation

4.1 Experimental Design

The recruitment and study process was approved by the Institutional Review Board (IRB).

Content To evaluate the efficacy of the system, we adapted a Biology lesson on organelles of eukaryotic cells from the OpenStax project [11]. We selected this lesson because we expected participants to have low prior familiarity with the material to ensure a learning process. Nevertheless, the lesson text is designed to be accessible to a general audience.

Conditions Similar to prior work [61, 31], we construct conditions to compare the efficacy of our EMT-based CTS to reading alone and to a QA chatbot with limited dialog. To study potential differences, we equip the QA chatbot with content from different sources under two distinct conditions: one using content generated by a biology teacher and the other using content from the LLM.

- Reading: Participants study the material without receiving additional support.
- Teacher QA (TQA): Participants study the material and can answer review questions presented by the chatbot. After submitting an answer, participants receive brief feedback about the correctness of their answer and a sample solution. Questions and answers are designed by a teacher.

Table 1: Learning performance across different conditions.

Conditions	# of participants		Previous Knowledge	Learning Performance
	Before filtering 100	After filtering 58		Post-test Scores (i.e.,Multiple-Choice Questions)
Reading	30	15	2.53 ± 0.41	5.07 ± 0.33
Teacher Q/A	17	7	**3.0 ± 0.58**	4.14 ± 0.83
LLM Q/A	23	15	2.2 ± 0.3	4.67 ± 0.35
Ruffle & Riley	30	21	2.67 ± 0.43	**5.19 ± 0.25**

Table 2: Learning experience across different conditions. The symbol "*" represents $p < 0.05$. The symbol "-" represents that this aspect was not asked in the corresponding condition.

Conditions	Learning Experience (1-strongly disagree, 7-strongly agree)						
	Engagement	Understanding	Remembering	Interruption	Coherence	Support	Enjoyment
Reading	4.33 ± 0.52	-	-	-	-	-	-
Teacher Q/A	5.0 ± 0.53	4.43 ± 0.65 *	4.43 ± 0.65 *	2.71 ± 0.64	5.43 ± 0.53	4.57 ± 0.57 *	3.71 ± 0.52 *
LLM Q/A	4.8 ± 0.47	4.4 ± 0.4 *	4.33 ± 0.42*	2.67 ± 0.45	4.8 ± 0.43*	4.0 ± 0.44 *	4.0 ± 0.44 *
Ruffle & Riley	**5.81 ± 0.3**	**5.81 ± 0.24**	**5.76 ± 0.22**	**2.19 ± 0.34**	**6.1 ± 0.21**	**5.9 ± 0.26**	**5.62 ± 0.31**

- LLM QA (LQA): Same as TQA, but questions and answers are generated by the LLM (Section 3).
- Ruffle&Riley (R&R): Participants study the material while being supported by the two conversational agents (Section 3). The system is equipped with the same questions as LQA.

Surveys/Questionnaires We evaluate system efficacy from two perspectives: *learning performance* and *learning experience*. Performance is captured via a multiple-choice post-test after the learning session, which consists of five questions written by a biology teacher recruited via Upwork [59] and two questions from OpenStax [11]. The learning experience is captured via a 7-point Likert scale questionnaire that queries participants' perception of engagement, intrusiveness, and helpfulness of the agents, based on previous work [49]. To ensure data quality, we employed two attention checks and one question asking participants whether they looked up test answers online. Further, we included a demographic questionnaire to understand participants' age, gender, and educational background.

Participants We recruited participants located in the USA who were fluent in English and had at least a high-school (HS) degree via Prolific [50]. Participants were randomly assigned to the conditions and were free to drop out at any point in the study. Overall, 100 participants completed the task. As shown in Table 1, 30 participants finished the reading condition, 17 finished TQA, 23 finished LQA, and 30 finished R&R. The imbalance is due to random condition assignments and dropouts.

Hypotheses We explore the following hypotheses. H1: *Learning Outcomes*: R&R achieves higher post-test scores than the baseline conditions (H1a); There is no significant difference between TQA and LQA (H1b). H2: *Learning experience*: R&R achieves higher ratings than the baseline conditions in terms of engagement, helpfulness in understanding, remembering, interruption, coherence, support received, and enjoyment (H2a); There are no significant differences between TQA and LQA (H2b).

4.2 Results and Analysis

After filtering participants who failed any of the attention check questions, or who did not rate "strongly disagree" when asked whether they looked up test answers, we were left with 58 (male: 33, female: 21, other: 4) out of the 100 participants (15 in reading, 7 in TQA, 15 in LQA, and 21 in R&R). The age distribution is 18-25 (8), 26-35 (20), 36-45 (18), 46-55 (9), over 55 (3). The degree distribution is: HS or Equiv. (22) Bachelor's/Prof. Degree (25), Master's or Higher (11).

Learning Performance The post-test consists of seven questions, each worth one point. The mean and standard error in post-test scores for each condition is provided by Table 1. A one-way ANOVA did not detect significant differences in post-test scores among the four conditions. Therefore, we find support for H1b but not for H1a. Even though not significantly different, we observed that participants in R&R achieved somewhat higher scores (5.19 ± 0.25) than in TQA (4.14 ± 0.83).

Learning Experience Table 2 shows participants' learning experience and chatbot interaction ratings. We tested for significance ($p < 0.05$) via one-way ANOVA, followed by Bonferroni post-hoc analysis. We found no significant differences in self-reported engagement levels between the four conditions.

However, among the three chatbot conditions, R&R was rated as significantly more helpful in aiding participants in understanding, remembering the lesson and providing the support needed to learn. Further, R&R participants expressed more enjoyment than TQA and LQA participants. In addition, participants found R&R provided a significantly more coherent conversation than LQA. Interestingly, even though we expected R&R to be rated as more interrupting, we found no significant differences in perceived interruption among the three chatbot conditions. Therefore, H2a is partially supported. In addition, there were no significant differences detected between LQA and TQA among the aspects of the learning experience. Thus, we cannot reject H2b.

Evaluation of Conversations in R&R We analyzed chat and behavior log data in R&R. First, based on logged responses and scrolling events, we found that participants taught Ruffle with different strategies. Some participants learned the lesson driven by the questions asked by Ruffle (10 out of 21) and focused on the conversation immediately after entering the learning session before reading the whole lesson text. Other participants read the text first before starting the conversation. Second, on average, participants submitted 1.33 ± 1.74 help requests. Third, the mean learning time in each condition is: reading (4 mins), TQA (11 mins), LQA (12 mins), and R&R (18 mins).

5 Discussion and Limitations

Here, we discuss lessons learned, future directions, and limitations.

Extending the System Evaluation While our evaluation showed that Ruffle&Riley can improve various learning experience metrics, we were not able to detect significant improvements in post-test scores. Our post-test featured recall-based multiple-choice questions, which represent a shallow form of knowledge assessment. Our observations align with prior research that found that reading and conversational tutoring can lead to similar outcomes in recall-based test formats [20]. As a next step, we want to revise our post-test format and employ fill-in-the-blank and essay questions to assess deeper understanding. We also want to evaluate knowledge retention over time and gauge the system's ability to facilitate learning in other domains (e.g., psychology and business).

Refining the Instructional Design Some users engaged in the conversational workflow before reading the lesson text and focused exclusively on the questions presented by the agents. This can cause users to miss important information that falls outside the scope of the tutoring script. In future evaluations, we want to orchestrate the learning activity in a more structured way, either by requiring the users to read the lesson text first or by interweaving lesson material with conversation [64]. Further, we observed that conversational learning activities require substantially more time than reading. This motivates us to explore more time-efficient CTS workflows [31] in future research.

Human-in-the-loop Capabilities The present study leverages an LLM to automatically generate a tutoring script from a lesson text and automates its orchestration in a free-form dialog via two conversational agents. While our study showcases the impressive capabilities of state-of-the-art NLP technologies, we think that it is crucial to move towards extending the system architecture with human-in-the-loop capabilities [66, 58]. We want to enable general educational practitioners to get involved in the instruction design process. For example, teachers might want to include their own review questions or insert expectations for students' answers into the tutoring script or alternatively choose among different candidates that the LLM can provide to them. Including human domain experts in the design process can further improve trustworthiness and content quality.

Limitations Ruffle&Riley is still in an early stage of development, and the present study is subject to several limitations. First, the system was evaluated in an online user study conducted via Prolific [50] with adult participants exhibiting diverse demographics (i.e., age and education). Current findings focus on a broad population of online users and might not generalize to more specific populations (e.g., K12 or college students). Before evaluating Ruffle&Riley with younger learners in institutional settings, we need to certify safe and trustworthy system behavior [17]. Relatedly, we need to verify the factual correctness of the information that surfaces during the conversations [28]. While we instructed the GPT4-based agents to only refer to information that occurs in the lesson text, and we have not observed incorrect information during system testing and data analysis, a systematic evaluation is fundamental. A related direction is the integration of safeguards and other validations mechanisms into the system to ensure benign outputs. Another limitation is that participants only took part in a single learning session. Users might require some time to get used to the workflow.

6 Conclusion

In 2019, the team around AutoTutor [7] reflected on the labor-intensive process in which conversational tutoring systems are created and hinted towards the possibility that, sometime in the future, one might be able to generate tutoring scripts fully automatically. Now, in 2023, we have reached a point where the capabilities of available NLP technologies enable us to make this vision become a reality. In the coming years, generative-AI-based content authoring tools are likely to allow researchers and educators to focus their limited time more on questions of effective instructional design and ITS architecture and less on system implementation. This might accelerate the evaluation of instructional design principles [30], leading to improvements in ITSs and to general insights for learning science.

Acknowledgments and Disclosure of Funding

We would like to thank Art Graesser for helpful suggestions and critiques during system development. We thank Xiaoyang Lei for creating illustrations for our two agents. We further thank Microsoft for support in the form of Azure computing and access to the OpenAI API through a grant from their Accelerate Foundation Model Academic Research Program. This research was supported by the AFOSR under award FA95501710218.

References

[1] Faruk Ahmed, Keith Shubeck, and Xiangen Hu. Chatgpt in the generalized intelligent framework for tutoring. In *Proceedings of the 11th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym11)*, page 109. US Army Combat Capabilities Development Command–Soldier Center, 2023.

[2] Vincent Aleven, Bruce M McLaren, Jonathan Sewall, and Kenneth R Koedinger. The cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. In *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pages 61–70. Springer, 2006.

[3] Vincent Aleven, Bruce M McLaren, Jonathan Sewall, Martin Van Velsen, Octav Popescu, Sandra Demi, Michael Ringenberg, and Kenneth R Koedinger. Example-tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education*, 26:224–269, 2016.

[4] Tiffany Barnes. The q-matrix method: Mining student response data for knowledge. In *American association for artificial intelligence 2005 educational data mining workshop*, pages 1–8. AAAI Press, Pittsburgh, PA, USA, 2005.

[5] Tiffany Barnes and John Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *International conference on intelligent tutoring systems*, pages 373–382. Springer, 2008.

[6] Anthony Botelho, Sami Baral, John A Erickson, Priyanka Benachamardi, and Neil T Heffernan. Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning*, 2023.

[7] Zhiqiang Cai, Xiangen Hu, and Arthur C Graesser. Authoring conversational intelligent tutoring systems. In *Adaptive Instructional Systems: First International Conference, AIS 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings 21*, pages 593–603. Springer, 2019.

[8] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International conference on intelligent tutoring systems*, pages 164–175. Springer, 2006.

[9] Michelene TH Chi, Stephanie A Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G Hausmann. Learning from human tutoring. *Cognitive science*, 25(4):471–533, 2001.

[10] Michelene TH Chi and Ruth Wylie. The icap framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist*, 49(4):219–243, 2014.

[11] Mary A Clark, Matthew Douglas, and Jung Choi. *Biology 2e*. OpenStax, 2018.

[12] Peter A Cohen, James A Kulik, and Chen-Lin C Kulik. Educational outcomes of tutoring: A meta-analysis of findings. *American educational research journal*, 19(2):237–248, 1982.

[13] Dorottya Demszky, Jing Liu, Heather C Hill, Dan Jurafsky, and Chris Piech. Can automated feedback improve teachers' uptake of student ideas? evidence from a randomized controlled trial in a large-scale online course. *Educational Evaluation and Policy Analysis*, page 01623737231169270, 2023.

[14] Diego Dermeval, Ranilson Paiva, Ig Ibert Bittencourt, Julita Vassileva, and Daniel Borges. Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *International Journal of Artificial Intelligence in Education*, 28:336–384, 2018.

[15] Sidney D'Mello, Andrew Olney, Claire Williams, and Patrick Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5):377–398, 2012.

[16] David Duran. Learning-by-teaching. evidence and implications as a pedagogical mechanism. *Innovations in Education and Teaching International*, 54(5):476–484, 2017.

[17] Marina Escobar-Planas, Emilia Gómez, and Carlos-D Martínez-Hinarejos. Guidelines to develop trustworthy conversational agents for children. *arXiv preprint arXiv:2209.02403*, 2022.

[18] Logan Fiorella and Richard E Mayer. The relative benefits of learning by teaching and teaching expectancy. *Contemporary Educational Psychology*, 38(4):281–288, 2013.

[19] Arthur C Graesser, Sidney D'Mello, Xiangen Hu, Zhiqiang Cai, Andrew Olney, and Brent Morgan. Autotutor. In *Applied natural language processing: Identification, investigation and resolution*, pages 169–187. IGI Global, 2012.

[20] Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36:180–192, 2004.

[21] Arthur C Graesser, Natalie K Person, and Joseph P Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522, 1995.

[22] Marcelo Guerra Hahn, Silvia Margarita Baldiris Navarro, Luis De La Fuente Valentín, and Daniel Burgos. A systematic review of the effects of automatic scoring and automatic feedback in educational settings. *IEEE Access*, 9:108190–108198, 2021.

[23] Neil T Heffernan, Kenneth R Koedinger, and Leena Razzaq. Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18(2):153–178, 2008.

[24] Dollaya Hirunyasiri, Danielle R Thomas, Jionghao Lin, Kenneth R Koedinger, and Vincent Aleven. Comparative analysis of gpt-4 and human graders in evaluating praise given to students in synthetic dialogues. *arXiv preprint arXiv:2307.02018*, 2023.

[25] Xudong Huang, Scotty D. Craig, Jun Xie, Arthur Graesser, and Xiangen Hu. Intelligent tutoring systems work as a math gap reducer in 6th grade after-school program. *Learning and Individual Differences*, 47:258–265, 2016.

[26] Ying Jiao, Kumar Shridhar, Peng Cui, Wangchunshu Zhou, and Mrinmaya Sachan. Automatic educational question generation with difficulty level controls. In *International Conference on Artificial Intelligence in Education*, pages 476–488. Springer, 2023.

[27] Nayoung Jin and Hana Lee. Stubot: Learning by teaching a conversational agent through machine reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3008–3020, 2022.

[28] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[29] Kenneth R Koedinger, Vincent Aleven, Neil Heffernan, Bruce McLaren, and Matthew Hockenberry. Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In *Intelligent Tutoring Systems: 7th International Conference, ITS 2004, Maceió, Alagoas, Brazil, August 30-September 3, 2004. Proceedings 7*, pages 162–174. Springer, 2004.

[30] Kenneth R. Koedinger, Julie L. Booth, and David Klahr. Instructional complexity and the science to constrain it. *Science*, 342(6161):935–937, 2013.

[31] Kristopher J Kopp, M Anne Britt, Keith Millis, and Arthur C Graesser. Improving the efficiency of dialogue in tutoring. *Learning and Instruction*, 22(5):320–330, 2012.

[32] James A. Kulik and J. D. Fletcher. Effectiveness of intelligent tutoring systems: A meta-analytic review. *Review of Educational Research*, 86(1):42–78, 2016.

[33] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121–204, 2020.

[34] Yoonjoo Lee, Tae Soo Kim, Sungdong Kim, Yohan Yun, and Juho Kim. Dapie: Interactive step-by-step explanatory dialogues to answer children's why and how questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2023.

[35] Krittaya Leelawong and Gautam Biswas. Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[36] Mark Liffiton, Brad Sheese, Jaromir Savelka, and Paul Denny. Codehelp: Using large language models with guardrails for scalable support in programming classes. *arXiv preprint arXiv:2308.06921*, 2023.

[37] Jionghao Lin, Danielle R Thomas, Feifei Han, Shivang Gupta, Wei Tan, Ngoc Dang Nguyen, and Kenneth R Koedinger. Using large language models to provide explanatory feedback to human tutors. *arXiv preprint arXiv:2306.15498*, 2023.

[38] Christopher J MacLellan and Kenneth R Koedinger. Domain-general tutor authoring with apprentice learner models. *International Journal of Artificial Intelligence in Education*, pages 1–42, 2022.

[39] Julia M Markel, Steven G Opferman, James A Landay, and Chris Piech. Gpteach: Interactive ta training with gpt based students. *EdArXiv*, 2023.

[40] Antonija Mitrovic. The effect of explaining on learning: a case study with a data normalization tutor. In *AIED*, pages 499–506, 2005.

[41] Steven Moore, Huy A Nguyen, Tianying Chen, and John Stamper. Assessing the quality of multiple-choice questions using gpt-4 and rule-based methods. *arXiv preprint arXiv:2307.08161*, 2023.

[42] Tom Murray. An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, pages 491–544, 2003.

[43] Huy A. Nguyen, Shravya Bhat, Steven Moore, Norman Bier, and John Stamper. Towards generalized methods for automatic question generation in educational domains. In Isabel Hilliger, Pedro J. Muñoz-Merino, Tinne De Laet, Alejandro Ortega-Arranz, and Tracie Farrell, editors, *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption*, pages 272–284, Cham, 2022. Springer International Publishing.

[44] Huy A Nguyen, Hayden Stec, Xinying Hou, Sarah Di, and Bruce M McLaren. Evaluating chatgpt's decimal skills and feedback generation in a digital learning game. *arXiv preprint arXiv:2306.16639*, 2023.

[45] Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. Autotutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24:427–469, 2014.

[46] OpenAI. Gpt-4 technical report, 2023.

[47] José Paladines and Jaime Ramirez. A systematic literature review of intelligent tutoring systems with dialogue in natural language. *IEEE Access*, 8:164246–164267, 2020.

[48] Zachary A Pardos and Shreya Bhandari. Learning gain differences between chatgpt and human tutor generated algebra hints. *arXiv preprint arXiv:2302.06871*, 2023.

[49] Zhenhui Peng, Yuzhi Liu, Hanqi Zhou, Zuyu Xu, and Xiaojuan Ma. Crebot: Exploring interactive question prompts for critical paper reading. *International Journal of Human-Computer Studies*, 167:102898, 2022.

[50] Prolific. Prolific, 2023. First release: 2014, Version: August 2023.

[51] Leena Razzaq, Jozsef Patvarczki, Shane F Almeida, Manasi Vartak, Mingyu Feng, Neil T Heffernan, and Kenneth R Koedinger. The assistment builder: Supporting the life cycle of tutoring system content creation. *IEEE Transactions on Learning Technologies*, 2(2):157–166, 2009.

[52] Jeff Rickel, Neal Lesh, Charles Rich, Candace L Sidner, and Abigail Gertner. Collaborative discourse theory as a foundation for tutorial dialogue. In *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2–7, 2002 Proceedings 6*, pages 542–551. Springer, 2002.

[53] Jeremy Roschelle, Mingyu Feng, Robert F. Murphy, and Craig A. Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4):1–12, 2016.

[54] Carolyn P Rosé. Interactive conceptual tutoring in atlas-andes. *Artif. Intell. in Education: AI-Ed in the Wired and Wireless Future*, pages 256–266, 2001.

[55] Sherry Ruan, Angelica Willis, Qianyao Xu, Glenn M Davis, Liwei Jiang, Emma Brunskill, and James A Landay. Bookbuddy: Turning digital materials into interactive foreign language lessons through a voice chatbot. In *Proceedings of the sixth (2019) ACM conference on learning@ scale*, pages 1–4, 2019.

[56] Shashank Sonkar, Lucy Liu, Debshila Basu Mallick, and Richard G Baraniuk. Class meet spock: An education tutoring chatbot based on learning science principles. *arXiv preprint arXiv:2305.13272*, 2023.

[57] Robert Sottilare, Arthur Graesser, Xiangen Hu, and Keith Brawner. *Design recommendations for intelligent tutoring systems: Authoring tools and expert modeling techniques*. Robert Sottilare, 2015.

[58] Jiarui Richard Tong and Timothy Xueqian Lee. Trustworthy ai that engages humans as partners in teaching and learning. *Computer*, 56(5):62–73, 2023.

[59] Upwork. Upwork, 2023. First release: 2013, Version: July 2023.

[60] Kurt VanLehn. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265, 2006.

[61] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist*, 46(4):197–221, 2011.

[62] Amali Weerasinghe and Antonija Mitrovic. Facilitating deep learning through self-explanation in an open-ended domain. *International journal of Knowledge-based and Intelligent Engineering systems*, 10(1):3–19, 2006.

[63] Daniel Weitekamp, Erik Harpstead, and Ken R Koedinger. An interaction design for machine teaching to develop ai tutors. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–11, 2020.

[64] Rainer Winkler, Sebastian Hobert, Antti Salovaara, Matthias Söllner, and Jan Marco Leimeister. Sara, the lecturer: Improving learning in online education with a scaffolding-based conversational agent. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

[65] Christopher R Wolfe, Colin L Widmer, Valerie F Reyna, Xiangen Hu, Elizabeth M Cedillos, Christopher R Fisher, Priscilla G Brust-Renck, Triana C Williams, Isabella Damas Vannucchi, and Audrey M Weil. The development and analysis of tutorial dialogues in autotutor lite. *Behavior research methods*, 45:623–636, 2013.

[66] Meng Xia, Mingfei Sun, Huan Wei, Qing Chen, Yong Wang, Lei Shi, Huamin Qu, and Xiaojuan Ma. Peerlens: Peer-inspired interactive learning path planning in online question pool. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.

[67] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[68] George Zografos and Lefteris Moussiades. A gpt-based vocabulary tutor. In *International Conference on Intelligent Tutoring Systems*, pages 270–280. Springer, 2023.