

Designing Simulated Students to Emulate Learner Activity Data in an Open-Ended Learning Environment

Paras Sharma
University of Pittsburgh
Pittsburgh, PA, USA
pas252@pitt.edu

Qichang Li
University of Pittsburgh
Pittsburgh, PA, USA
qil107@pitt.edu

ABSTRACT

Open-design environments, under open-ended learning environments, provide high agency to learners to define their goals and pathways toward those goals. However, such environments could be difficult to navigate through for some learners due to this openness in goals and activities. Our long-term goal is to build intelligent pedagogical agents to support learner activities within these environments using different dialogue strategies. In this work, we propose to build a Simulated Students system to emulate learner activities in an open-design environment. We hope to use this simulated data in future work to distill knowledge from Large Language Models (LLMs) to build adaptive, and context-based reinforcement learning dialogue models for learner support. We present the early results and proposed directions of our ongoing work and seek advice on how the different strategies that we propose to use in this work could be further used to build adaptive and context-based dialogue models for effective learning in open-design environments.

Keywords

Simulated Students, Open-Design environments, Dialogue Systems, Large Language Models

1. INTRODUCTION

Open-design environments comprise open-ended learning tasks that facilitate learner agency in the ways they define the learning goals and pathways toward those goals [11, 7]. Learning in these environments often involves learners transitioning through different self-regulatory learning processes [15] and cognitive states [9, 3]. However, while transitioning through these processes a learner could end up in undesirable states like “Wheel Spinning” [3] associated with long action times ultimately leading to a “stuck” state, hence affecting the whole learning process. Dialogue interactions as a medium have been widely used to support learners in open-ended learning experiences. Feedback dialogues have been used to help learners in problem-solving tasks [6]. Self-

explanation dialogues help promote critical thinking in learners about their actions [1]. Information-delivery dialogues help learners in the knowledge construction processes throughout the learning [8]. An intelligent pedagogical agent actively building learner activity models can support learning through various dialogue interactions with the learners. However, these dialogue interactions should be learner-specific and often require large amounts of diverse learner data to effectively generate responses outside pre-defined rule-based scenarios. In this work, we propose to build a Simulated Students system to generate large amounts of plausible and diverse learner interaction data in an open-design learning system, ultimately building a reinforcement learning model for learner-agent dialogue interactions to support learning.

To generate agent responses based on these learner action sequences, we plan to use Large Language Models (LLMs), e.g. GPT-4¹. Recently LLMs have seen a huge boom in their use in different application domains because of their capabilities to understand and model natural language efficiently. These models have been trained on huge amounts of internet data and have shown remarkable performance in many language tasks like question-answering tasks, information retrieval, and dialogue systems. These models show emergent capabilities and have been used in various downstream tasks across domains (such as game-design [2]) and applications (such as conversational agents [5]). Their ability to act as conversational agents and respond to queries across domains without additional training shows potential for their use in open-ended educational systems.

However, LLMs often possess inherent human biases [10], have a black-box nature of the model outputs, and present over-generalized responses due to training on a wide variety of internet data. Hence, we believe their direct use in educational systems engaging with young children is not suitable. Moreover, there are potential security risks and ethical concerns to feeding learner data directly into LLMs. Whether the private learner data should live on the internet and be used in perpetuity to train LLMs, and how to get meaningful consent for these behaviors, especially when minors are involved, are some important questions that need to be thought of before deploying these models directly in educational systems. We propose to use “*Domain Knowledge Distillation*” using prompt engineering [14] to extract responses

P. Sharma and Q. Li. Designing simulated students to emulate learner activity data in an open-ended learning environment. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 986–989, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

© 2024 Copyright is held by the author(s). This work is distributed under the Creative Commons Attribution NonCommercial NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.
<https://doi.org/10.5281/zenodo.12730023>

¹<https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4>

for different simulated learner action sequences from LLMs, and then locally filter these responses with safety checks before using them to train a reinforcement learning dialogue system to support learners.

In this work, we propose to first build a Simulated Students system to generate learner action sequences based on learner goals and activity states in an open-design learning environment. We model these goals and activity states using the learner-robot interaction data from a summer camp with 14 middle school girls. In our work to actively model learner actions throughout the learning process [13], we discuss 2 techniques: Sequence Mapping and Hidden Markov Models to extract 7 activity states (self-regulatory learning phases and cognitive states) that a learner transitions through in an open-design environment. In this paper, we further build on the extracted states to emulate learner behaviors by developing a Simulated Students system generating sequences of learner actions. We then propose using the simulated data as prompts to the GPT 3.5-turbo model to generate dialogue responses based on learner states. We will use this generated dialogue dataset in building a reinforcement learning dialogue model to support learning in our open-design environment.

2. PROPOSED CONTRIBUTIONS

Our overall aim is to support a learner, working in an open-ended learning environment, through an adaptive and personalized dialogue system based on the learner’s current cognitive and affective state using knowledge distillation from LLMs. For this purpose, in this paper, we propose to build a Simulated Students system that could emulate different learner behaviors and generate learner action data based on those behaviors in open-design learning environments. The Simulated Students system would help us to generate large amounts of plausible learner action data as they transition through the different cognitive and affective states while progressing toward their goals. We will then evaluate the effectiveness of this generated data to emulate actual learner behaviors. In future work, we will use LLMs to analyze open-ended learning spaces and distill knowledge from LLMs by using various prompts to build a small and localized dataset for responses to different learner states. The localized dataset would be further utilized to develop an RL-based adaptive dialogue system.

The following sections describe the overview of our proposed Simulated Students system.

2.1 Input and Output for Simulated Students System

Our proposed Simulated Students system could support 2 types of input-output behaviors. The main assumption of our Simulated Students system is that the learner’s actions are always guided by their goals and current activity states (SRL or cognitive) in open-ended learning environments and are not random. Hence, we chose to design our system with only these 2 as the parameters to effectively mimic the actual learner behaviors.

1. First, is the input of a set of learner goals and a cognitive or SRL state for the learner, to generate an output

representing a set of learner actions toward the specified goals under the given state. For example, for input with a goal of “Making the robot move” and a “Forethought” activity state, our system would generate plausible action sequences representing learner actions toward this goal under the given activity state.

2. Second, is just the input of a set of learner goals to generate a series of learner actions toward achieving those goals while transitioning through various cognitive and SRL states throughout the process. For example, for an input goal list with “Making the robot dance and then say its name”, our system would generate plausible learner action sequences toward these goals when the learner transitions through different activity states. These transitions between states in the simulated system could either occur randomly or based on specified state and transition probabilities.

The number of actions to be generated can also be specified as an input, default value of 200 actions (the mean number of actions from our collected data in an initial study).

2.2 Correctness Criteria for Simulated Students Output

- Any atomic action generated by the system needs to be valid under the current system state. For example, an action to remove an actuator from the robot before adding it to the robot would be considered invalid.
- For the first I/O type, the generated action sequences should match the specified cognitive or affective state based on codes in our pre-defined codebook [13]. This evaluation could be done either automatically using scripts or by human coders.
- For the second I/O type, the generated action sequences should follow the pattern specified by the input state and transition probabilities for all the learner activity states. We would evaluate the probability of the simulated data to be generated by either the HMM or Sequence Mapping model of the real learner data from our initial study.

3. METHODS & PRELIMINARY RESULTS

In this section, we describe our data collection method, proposed system design, and current progress.

3.1 Data and System

We collected learner interaction data as a part of a 2 week-long summer camp with 14 upper-elementary to middle-school (4th to 7th grades, 8 – 12 years old) girls, recruited through our community partner from a historically African American neighborhood in a mid-sized US city. 12 of the learners identified as Black and 2 with no answer. The learners were involved in a task to “*Create a robot protege to be presented on a robot runway*”. We developed a custom multimodal system that helped learners design, build, and program robots using different sensors, actuators, coding blocks, design materials, and a modified Hummingbird Robotics kit. Our system also had provisions for learner interaction with their designed robot through dialogues and

video interactions. Our system automatically logged all learner interaction events with the system: sensor activities (addition/removal), programming activities (block addition/updates/removal), dialogue, and video interaction activities. We collected 2961 instances of log action data across all the learners (Mean = 199, SD = 141.15). Each of these actions also contains timestamp information associated with them.

3.2 Simulated Students System Design

Goal Understanding. A learner could have multiple goals and each goal could be either *concrete* or *abstract*. Concrete goals can be directly accomplished by using one or more of the fixed possible atomic actions in the system in no particular order. For example, "Turn the red LED on for 5 seconds" requires 2 fixed atomic actions - attaching the LED to the robot and adding the code block for turning the red LED on. On the other hand, abstract goals depend on individual learner definitions and could have various possible atomic actions. For example, a goal "I want to make my robot dance" could have various atomic actions associated with it like repeating a forward and backward motion, turning the robot's head to different positions, and making the robot speak while moving its hands.

One way to map abstract goals to atomic actions is to use existing learner data to create a base set of abstract goals and then ask human annotators to add to this set. However, this process is costly, time-extensive, unscalable, and might not produce diverse enough goals. We propose to distill domain knowledge from LLMs to generate a large set of abstract goals with their associations with individual atomic actions in our system. We will use different prompt engineering techniques to generate this data and then use different clustering algorithms like HDBSCAN [4] to assess the diversity of the data. Additionally, we plan to evaluate the safety and usability of the data using human experts.

Goal Matching & Atomic Action Categorization. We use BERT-based sentence transformers [12] to calculate embeddings for the input goal and then calculate cosine similarity (threshold 70%) between the GPT-generated goal data and the input embeddings to select the best goal. In case of no match, we would add this goal to the dataset and use it as a seed to generate more related goals. We will also retrieve atomic actions and their execution order from the dataset based on the initial goal matching. Each Atomic action has an associated set of sensors and code blocks, which will be used as input for further stages in the simulation pipeline.

Simulation Based on Goal and Learner State Inputs. This part of the system handles the first I/O type and generates the learner action sequence based on the input learner state and goals. We have developed pseudocode for each learner activity state based on the rules defined in our previous work [13]. We propose to randomly select a plausible action from all the actions at a particular time and then append it to the generated action sequence based on the defined pseudocode. We would also experiment with selecting

actions based on the probability distribution of the actual learner actions in an activity state based on our collected system data from the summer camp.

Simulation Based on Learner Goal. This part of the system handles the second I/O type and generates the learner action sequences based on the input set of learner goals. We will experiment with the following 3 ways of transitioning between different activity states and selecting actions inside the activity state:

- **Random.** In this method, we will randomly (with equal probabilities) select a learner activity state, the number of actions to generate in that state (maximum 20), and the plausible action in that state. We hope this will give us a more general view of all plausible learner actions and the transitions between them. However, this could also lead to impossible or unobserved state transitions in the real learner activity data.
- **Using Initialization from HMM.** In this method, we would initialize the start and transition probabilities for every activity state based on the HMM model of the real learner data as described in our previous work [13]. The number of actions and the plausible action for a state will still be selected randomly to get more diverse action sequences. We believe this will generate data that will mimic the actual learner behaviors while adding diversity to the generated learner sequence of actions.
- **Using Initialization from Sequence Mapping.** We would initialize the start and transition probabilities for every activity state based on the Sequence Mapping model of the real learner data as described in [13]. We would randomly select the number of actions and the plausible action for a state.

Termination. The termination condition for the action sequence generation is the number of actions to be generated in the input (default as 200) or the completion of all the atomic actions relevant to the input goals.

Evaluation. (1) We will first perform automated evaluations to check the plausibility of each action in the environment, the plausibility of an action after a sequence of actions that happened before this action, and the validity of the system state after each action. (2) We will then add more checks to verify if the generated data represents the real learner activity state distributions and fits the identified HMMs. (3) We will further use human-expert evaluation from our team to verify if the input state matches the identified state and if the action sequence generated is relevant to the input goals.

3.3 Current Work and Limitations

We have generated the initial goal dataset containing a list of 58 plausible learner goals with associated atomic actions and the execution order using the GPT 3.5-turbo model.

We have tried with 2 different prompt structures (explaining our system components, roles, and set of actions) to generate this dataset. We have tried zero-shot and one-shot approaches for prompting. Based on our initial human evaluations, the generated data has many diverse concrete goals. However, there is still a lack of enough abstract goals. We are tuning our prompting style to generate more diverse and abstract goals.

Based on this initial data, we also experimented with matching the input goal to a relevant goal from the dataset using BERT sentence embeddings and cosine similarity. Our manual testing verified the efficiency of our method in finding a similar goal from the dataset to an input goal. However, this initial evaluation indicated that the order of the events in the input goal is not very well considered in our matching technique. For certain goal inputs, for example “Turn right after speaking Hello” the matching algorithm could return the goal that addresses these events in no specific order like “Say hello and turn right”. We believe this might be because of the smaller size of examples representing ordering in our dataset. We plan to generate more data with explicit ordering of actions along with including action ordering in the goal matching.

4. ADVICE SOUGHT

For this doctoral consortium, we would like advice regarding the following topics:

1. How can we tune prompts to generate more diverse goals from LLMs?
2. How can we enforce ordering in goal matching? Do you have recommendations for understanding novel learner goals based on the data generated from LLMs?
3. How can we efficiently evaluate the generated data by our Simulated Students system? Are there any other evaluation techniques you would like us to look at?
4. What general suggestions do you have for our design and evaluation methods to build a Simulated Students system to emulate learner activities?

5. ACKNOWLEDGMENTS

We would like to express our sincere thanks to Dr. Erin Walker for her guidance and valuable comments. We thank our team for their support with the system development. We also thank Dr. Kimberly Scott, Dr. Angela E.B. Stewart, Dr. Amy Ogan, Dr. Tara Nkrumah, and the rest of the social programmable robot team. This work is funded by NSF DRL-1811086 and DRL-1935801.

6. REFERENCES

- [1] V. Alevan, A. Ogan, O. Popescu, C. Torrey, and K. Koedinger. Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In J. C. Lester, R. M. Vicari, and F. Paraguaçu, editors, *Intelligent Tutoring Systems*, pages 443–454, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [2] T. Ashby, B. K. Webb, G. Knapp, J. Searle, and N. Fulda. Personalized quest and dialogue generation in role-playing games: A knowledge graph- and language model-based approach. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.
- [3] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International Conference on Artificial Intelligence in Education*, 2013.
- [4] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [5] S. Demetriadis and Y. Dimitriadis. Conversational agents and language models that learn from human dialogues to support design thinking. In C. Frasson, P. Mylonas, and C. Troussas, editors, *Augmented Intelligence and Intelligent Tutoring Systems*, pages 691–700, Cham, 2023. Springer Nature Switzerland.
- [6] M. Evens, R.-C. Chang, Y. Lee, L. Shim, C.-W. Woo, and Y. Zhang. Circsim-tutor: An intelligent tutoring system using natural language dialogue. pages 13–14, 01 1997.
- [7] P. Fournier-Viger, R. Nkambou, and E. M. Nguifo. *Building Intelligent Tutoring Systems for Ill-Defined Domains*, pages 81–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [8] A. C. Graesser, K. VanLehn, C. P. Rose, P. W. Jordan, and D. Harter. Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4):39, Dec. 2001.
- [9] S. Hutt, J. Hardey, R. E. Bixler, A. E. B. Stewart, E. F. Risko, and S. K. D’Mello. Gaze-based detection of mind wandering during lecture viewing. In *Educational Data Mining*, 2017.
- [10] H. Kotek, R. Dockum, and D. Sun. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, CI '23, page 12–24, New York, NY, USA, 2023. Association for Computing Machinery.
- [11] S. Land, M. Hannafin, and K. Oliver. *Student-Centered Learning Environments: Foundations, Assumptions, and Design*, pages 3–25. 01 2012.
- [12] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [13] P. Sharma, A. E. Stewart, Q. Li, K. Ravichander, and E. Walker. Building learner activity models from log data using sequence mapping and hidden markov models. 2024.
- [14] Y. Tang, A. A. B. da Costa, J. Zhang, I. Patrick, S. Khastgir, and P. Jennings. Domain knowledge distillation from large language model: An empirical study in the autonomous driving domain, 2023.
- [15] B. J. Zimmerman. Attaining self-regulation: a social cognitive perspective. In *Handbook of Self-Regulation*, pages 13–39. Elsevier, 2000.