Project Proposal: AI Tutee for Data Visualization Skills

Alexis Martin
amartin375@gatech.edu

1 INTRODUCTION

Large Language Models (LLMs) are quickly revolutionizing the way we interact with computers. Unlike traditional tools like search engines or programming languages which required keywords and rigid syntax respectively, LLMs allow users to converse with a computer using natural languages like English. They have been found to be excellent at a diverse array of tasks, and even good enough to pass undergraduate computer science (CS) coursework (Richards et al., 2024).

They are so good in fact, that their impact on education is both exciting, and worrisome. Worrisome due to the risk of plagiarism and the negative impact this type of usage can have on student's learning (B. Chen et al., 2024). Exciting, because with their vast amount of knowledge, it is hopeful that LLMs can provide great benefits to students and educators, especially with regards to providing rapid access to personalized help (Beale, 2025).

Therefore, the objective of this proposal is to offer a tool that leverages LLMs to help students through the evidence-based learning-by-teaching paradigm.

2 RELATED WORK

Despite the relative newness of LLMs, research to leverage LLMs in education to improve learner outcomes is plentiful. A lot of the work is related to the impact of LLMs on plagiarism, as well as building AI tutors and teachable agents.

2.1 LLM-Based Tutors

2.1.1 *Tools in Industry*

Numerous tools exist in industry, however, most are closed-source, making opaque points of interests such as design decisions, objective results, or even how to replicate/build the system.

MathGPT, EduGPT, EduChat, Tutorai.me, Khanmigo.ai, TutorOcean are examples of tools claiming to leverage AI (LLMs) to help students learn better. ChatGPT is a popular base model on which these tools build on top of.

Most of these advertise similar features and functions, most notably:

1. Always-available tutor
2. Personalization
3. Boosting Engagement
4. Creation of flash cards and study material

They also often display testimonials on their websites, claiming to improve learning outcomes, but experiments to support these claims were not found. Khanmigo had an FAQ claiming to guide students towards the answers, rather than giving answers straight up, but tool is a black box, so it is difficult to understand which teaching principles are guiding it and how it is built.

2.1.2 *SocraticLM*

As mentioned previously, many tools are being developed to combine LLMs and Intelligent Tutoring Systems (ITS). The authors of SocraticLM argue that these systems are too simple, limiting the learning process to a series of questions and answers, without actively engaging the student with the material (J. Liu et al.,).

SocraticLM is a tool developed to fix that gap by using Socratic style dialog to guide students towards better self-reflection and engagement with the material. As a first step, a multi-agent process is used to generate a dataset of Socratic dialog. To do so, a Dean supervising a Teacher, who in turns interacts with a Student are used to generate this realistic dataset, which in turn is used to train an open source GPT model to learn Socratic conversation-making.

SocraticLM is found to be superior for Socratic teaching when compared to "plain" GPT4. Some limitations include the narrow field investigated as well as having access to an open-source model to perform the training on.

2.1.3 *MWPTutor*

Autotutor is an ITS which necessitate experts to attempts to predict students' misconceptions, and write a set of rules and states to provide engaging dialog with the student. As a result, it takes an expert a hundred hours of careful crafting to handle one hour of student dialog with Autotutor (Pal Chowdhury et al., 2024).

However, Chowdhury, Zouhar and Sachan (2024) built MWPTutor, a system similar to Autotutor, but focused on Math Word Problems (MWP) that leverages LLMs to author the state space and reduce manual human authoring requirements. They found that the combination of pedagogically-based ITS and LLM worked better than simply using GPT-4. However, important weaknesses were found, such as sounding monotonous and dry, It also can get lost in context and inadvertently give out the answer, output a wrong answer, or ignore a student multi-step answer to focus on their initial one (thus holding the student back).

2.1.4 *Personality-Aware Student Simulation for Conversational Intelligent Tutoring Systems*

This framework integrates personality traits in a conversational ITS for language learning (Z. Liu et al., 2024). It includes five different personality traits from the Big Give theory from Costa and McCrae to better simulate student interactions. Psychometric tests were conducted by experts who noted that the LLMs were successful in simulating the traits given to them.

This work suggests that LLMs can be used to provide realistic simulations of student behaviors and personalities for the purpose of tutoring systems, under the Big Five theory for dialogic interaction. It also provides strategies for scaffolding the simulation to different simulated students levels of knowledge. A core limitation of the work is that it does not evaluate biases or hallucinations, which might limit the use in a real tutoring situation where the stakes are higher.

2.2 Teachable Agents

2.2.1 *Learning by Teaching ChatGPT*

ChatGPT can be used out of the box as a teachable agent, with some prompting to guide it to be a more realistic help-seeker (A. Chen et al.,). The study by Chen et al. demonstrated this possibility (with ChatGPT4), which showed improved programming abilities among a small sample of learner. Limitations found that due to ChatGPT's strong default abilities, it would often get to the answer really quick, limiting student learning of critical debugging skills.

2.2.2 *MathVC*

This tool uses LLMs to simulate a group of students and recreate a group study scenario. Prompt engineering on a base LLM model is the method use to for simulation of students. It generates diversity by using configuration schemas to assign different personalities and skill level for each simulated student, as well as configuration to limit the release of answers (Yue et al., 2025). The tool focuses on teaching mathematical modeling skills and hasn't been tested on its target population (middle-schoolers), but has been tested by field experts who assessed the tool as realistic. Authors also haven't evaluated how the tool adapts to different learner characteristics and didn't investigate the flow of conversation quantitatively.

2.2.3 *TeachYou and Algobo*

This solution introduces the concept of a *Teaching Helper* (TeachYou) on top of an AI tutee (AlgoBo), to help human learners. TeachYou is the platform which orchestrate the human learner's learning, starting from teaching a subject (binary search) all the way to helping the human learner reflect on their own teaching of AlgoBo. It also orchestrate how AlgoBo reacts, from being an implementer of the human's teaching, to a questioner, leading to further engagement and metacognitive skill improvements.

The tool focused on programming knowledge, specifically algorithm learning and procedural knowledge, and might not be generalizable to other topics (to be determined).

2.2.4 *MatLabTutee*

This tool introduces the learning by teaching paradigm to MatLab. It is simulating a MatLab student by using Chain-of-Thoughts prompting to simulate a novice learner, which it was found successful at doing (Rogers et al., 2025).

Benefits from the tools for human learners were a better self-assessment and more enjoyment from learning computer science. However, depending on the characteristics of the learners, it was found that the tool might also discourage them if their knowledge level was too low. The lack of guidance was raised as a reason for certain learners abandoning the tool.

2.2.5 *HypoCompass*

HypoCompass is another tool that leverages multi-agent learning by teaching. Its goal is to help human learners with debugging skills, and to achieve this goal simulates a queue of help-seeking AI agents, themselves powered by GTP-3-Turbo and GPT4 (Ma et al., 2024).

2.3 LLM Impact on Plagiarism

Review of the state of LLM in education has shown that even when used with good intentions, over-reliance on out-of-the-box LLMs is impacting learning outcomes negatively (Beale, 2025). This can be because LLMs are too quick to give answers to their users, but also because they make plagiarism easier than ever. According to the same review, LLM-generated answers are rapidly becoming more difficult to identify against real answers as LLMs become more powerful and emergent abilities to emulate different personalities arise. That is, with some tuning, LLMs are able to write answers that are near impossible to identify as AI-generated, when compared to real student answers (Richards et al., 2024).

Additionally, research has demonstrated a shift of plagiarism from traditional cheating hubs to LLM-based cheating (B. Chen et al., 2024, Becker et al., 2023).

3 PROPOSED WORK

The purpose of this project is to build a functional prototype for a tool leveraging LLMs as teachable agents, guided by learning-by-teaching principles. The scope of the prototype will be limited to building the AI student and about three different skills related to data visualization best practices, targeting career changing adults. These adults would be novice to the field of data visualization. As well, a small pilot evaluation will be conducted.

3.1 Prototype AI Tutee

The prototype AI tutee will be constrained using prompting configuration files to adopt the role of learners with different skill levels (beginner, intermediate or advanced), for each of the teaching scenarios.

It will be prompted to ask clarifying questions and make conceptual mistakes to engage and challenge the human learner's knowledge. It will also be prompted to avoid solving tasks immediately, rather to solve them only after it receives instructions from the human learner.

The AI learner will also encourage the human learner with positive reinforcement, without being negative.

3.2 Teaching Scenarios

The prototype will focus on the following skills:

1. Identification of data types: The ability to recognize between categorical and numerical data, as well as their sub-types (ordinal vs nominal, discrete vs continuous, time data, etc.)
2. Connecting data types to appropriate charts: When given data to analyze, understanding when to use different type of charts such a bar charts, scatter plots, pie charts, etc.
3. Matching charts and analytical task together: When given a specific task, such as observing data over time, understand what type of chart to use.
4. Data Preparation: Based on a task and data, understanding different data preparation techniques and when to use them to build the desired chart. For example, identifying missing data, handling outliers, duplicates, standardizing or modifying data formats.

For each scenario, settings will be available to set the human learner's level of knowledge to trigger a different configuration or personality for the AI student. A "beginner" level would trigger more simplistic scenarios and more straightforward questions. With "intermediate" and "advanced" levels, the scenarios will be more challenging, nuanced, and require better command of the material.

For the human learner to get feedback on their teaching (and understanding of the material), a list of level-appropriate questions will be generated for the AI students to answer, as well as an explanation for why it chose each answer. This little test will be answered before and after the lesson, to compare how the AI student performed.

3.3 Tooling and UI

The tool will be built using the Python programming language, and interface as a web app with streamlit. The AI tutee will be built using the OpenAI API.

The UI for the prototype will be a simple menu to select which skill to work on and which level to set the AI tutee. This will lead to a chatbot interface, where the chatbot will greet the human learner with an introduction and a list of misconceptions to work on. Upon completion of the session, a button to trigger the ending test will be available, which the human learner can push to get feedback on how their AI tutee learned during the lesson. A score showing how much it improved will be displayed, terminating the session.

3.4 Pilot Evaluation

The author and a few volunteers (one to five) will conduct a small evaluation of the tool.

A qualitative analysis will be provided on the following criterion:

- The AI Student has adequate misconceptions
- The knowledge level of the AI student scales appropriately based on the selected setting
- The AI student engages with the human learner by asking questions, being positive and provides explanation for its answers
- The AI tutee learns from the teaching and shows signs of progress on a small test before and after the session

The conversations will be logged to assess the quality of the interactions.

4 DELIVERABLES

4.1 Intermediate Milestone 1

For this milestone, the video will focus on:

- The definition of the teaching scenarios, including the exhaustive list of sub-skills to be handled by the tool.
- Early prompt-engineering experiments to demonstrate how the model act as a student. This might be done on a very simple script that triggers a chat session with the AI student within the context of a teaching scenario.
- Demonstrate the strengths and weaknesses of the approach at this point.

4.2 Intermediate Milestone 2

For this milestone, the scenarios and prompt engineering should be near-completion or fully completed, and the tool have a basic UI.

The video will focus on:

- Prototype demonstration with at least two different scenarios.
- The demonstration should showcase the AI tutee engaging the human learner through questions and making common errors.

4.3 Final Project Deliverable

As part of the final project deliverable, the following will be submitted:

- Full codebase including python code and prompts
- Written scenarios of a human student interaction
- Final report
- Final presentation slides and video

5 TASK LIST

Table 1—Project Task List

Week	Task #	Task Description	Est. Time (hrs)	Deliverable / Check-in
8	1	Detail skills to teach; Begin Draft prompting to simulate student tutee	8	Weekly status check
9	2	Continue draft prompting; Start designing interaction flows	8	Weekly status check
10	3	Build low-fidelity prototype (Streamlit/Jupyter setup, API integration) ; Record **Milestone 1 video** (design doc + prototype sketches + prompt experiments)	6	**Milestone 1 Oct 26**
11	4	Implement first 2 scenarios in prototype; test interactions for realism and scaffolding	10	Weekly status check
12	5	Refine prototype (adjust prompts, error simulation); prep demo content	8	Weekly status check
13	6	Expand prototype to all skills; Record **Milestone 2 video** (prototype demo, 2 scenarios working)	6	**Milestone 2 Nov 16**
14	7	Run pilot test with self and/or volunteers	12	Weekly status check
15	8	Analyze pilot data; Polish prototype/code; Start writing report	12	Weekly status check
16	9	Write and finalize report; prepare final presentation slides/video	16	**Final project due**

6 REFERENCES

. Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 500–506. https://doi.org/10.1145/3545945.3569759

. Chen, B., Lewis, C. M., West, M., & Zilles, C. (2024). Plagiarism in the Age of Generative AI: Cheating Method Change and Learning Loss in an Intro to CS Course. *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, 75–85. https://doi.org/10.1145/3657604.3662046

. Liu, Z., Yin, S. X., Lin, G., & Chen, N. F. (2024, November). Personality-aware Student Simulation for Conversational Intelligent Tutoring Systems. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 626–642). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.37

. Ma, Q., Shen, H., Koedinger, K., & Wu, S. T. (2024). How to Teach Programming in the AI Era? Using LLMs as a Teachable Agent for Debugging. In A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, & I. I. Bittencourt (Eds.), *Artificial Intelligence in Education* (pp. 265–279). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-64302-6_19

. Pal Chowdhury, S., Zouhar, V., & Sachan, M. (2024). AutoTutor meets Large Language Models: A Language Model Tutor with Rich Pedagogy and Guardrails. *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, 5–15. https://doi.org/10.1145/3657604.3662041

. Richards, M., Waugh, K., Slaymaker, M., Petre, M., Woodthorpe, J., & Gooch, D. (2024). Bob or Bot: Exploring ChatGPT's Answers to University Computer Science Assessment. *ACM Trans. Comput. Educ.*, *24*(1), 5:1–5:32. https://doi.org/10.1145/3633287

. Beale, R. (2025, July). The Revolution Has Arrived: What the Current State of Large Language Models in Education Implies for the Future [arXiv:2507.02180 [cs] version: 1]. https://doi.org/10.48550/arXiv.2507.02180

. Rogers, K., Davis, M., Maharana, M., Etheredge, P., & Chernova, S. (2025). Playing Dumb to Get Smart: Creating and Evaluating an LLM-based Teachable Agent within University Computer Science Classes. *Proceedings of the 2025 CHI Confer-*

ence on Human Factors in Computing Systems, 1–22. https://doi.org/10.1145/3706598.3713644

. Yue, M., Lyu, W., Mifdal, W., Suh, J., Zhang, Y., & Yao, Z. (2025, January). MathVC: An LLM-Simulated Multi-Character Virtual Classroom for Mathematics Education [arXiv:2404.06711 [cs]]. https://doi.org/10.48550/arXiv.2404.06711

. Chen, A., Wei, Y., Le, H., & Zhang, Y. (n.d.). Learning by teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjet.70001]. *British Journal of Educational Technology*, *n/a*(n/a). https://doi.org/10.1111/bjet.70001

. Liu, J., Huang, Z., Xiao, T., Sha, J., Wu, J., Liu, Q., Wang, S., & Chen, E. (n.d.). SocraticLM: Exploring Socratic Personalized Teaching with Large Language Models.