QuickTA : Exploring the Design Space of Using Large Language Models to Provide Support to Students

Harsh Kumar[1,*], Ilya Musabirov[1], Joseph Jay Williams[1] and Michael Liut[2]

[1]University of Toronto
[2]University of Toronto Mississauga

Abstract
Pre-trained large language models (LLMs) show promise in providing support to students through dialogues. However, current research in LLM-based support has highlighted the need to involve different stakeholders (e.g., instructors, researchers, students) in the design and deployment of these interactions. Based on our formative interviews with students and the prior literature, we are designing a system for instructors to: (1) program LLMs according to the task, (2) provide support to students through a chat interface, and (3) collect student feedback and usage statistics to inform future deployments. In this work, we report on our ongoing development of the system, design considerations, possible use cases of the system, and the path to the deployment of the system for a database management course. We hope that other researchers could build on this work to design systems that enable human-AI collaboration when it comes to improving the learning outcomes of students.

Keywords
large language models, intelligent tutoring systems, intelligent teaching assistants, human-AI collaboration

1. Introduction

There has been a decade-long pursuit of using artificial intelligence to provide support to students and teachers in the form of Intelligent Tutoring Systems [1, 2, 3, 4] and Teaching Assistants [5, 6, 7]. Large language models (LLMs) have been gaining attention as a potential support system for students in educational settings [8, 9, 10]. Pre-trained LLMs have shown promise in providing assistance through dialogues, but current research in this area has highlighted the need to involve different stakeholders, such as instructors, researchers, and students, in the design and implementation of these interactions [11].

This paper reports on the ongoing development of a system designed to address this need. The system is intended for instructors to program LLMs according to the task, provide support to students through a chat interface, and collect student feedback and usage statistics to inform

Workshop on Partnerships for Co-Creating Educational Content at the Learning Analytics and Knowledge Conference 2023 (LAK '23), March 13, 2023, Arlington, TX, USA
*Corresponding author.
harsh@cs.toronto.edu (H. Kumar); ilya@musabirov.info (I. Musabirov); williams@cs.toronto.edu (J. J. Williams); michael.liut@utoronto.ca (M. Liut)
https://harsh-kumar.com/ (H. Kumar)
0000-0003-2878-3986 (H. Kumar); 0000-0003-2246-0094 (I. Musabirov); 0000-0002-9122-5242 (J. J. Williams); 0000-0003-2965-5302 (M. Liut)

future deployments. We detail the system design considerations and possible use cases, with a specific focus on a database management course. The goal is to demonstrate how human-AI collaboration can be used to improve student learning outcomes and provide a foundation for future research in this area.

2. Relevant Work

In this section, we provide a brief overview of Intelligent Tutoring Systems, followed by a review of some applications of large language models for human support applications.

2.1. Intelligent Tutoring Systems and Teaching Assistants

Intelligent tutoring systems (ITSs) are defined as computer programs that are designed to incorporate AI techniques to provide tutors who know what they teach, who they teach, and how to teach it [12]. These systems have been studied for decades to replicate learning experiences with human tutors [1, 2, 3, 4]. Although the original objective of ITSs is to replicate the teaching of human teachers, in some cases they have even been shown to outperform human teachers in terms of learning outcomes for students [13, 14]. ITSs such as Duolingo have found tremendous commercial value when it comes to learning languages [15]. Recent advances in pre-trained language models have been applied to specific tasks related to tutoring, such as the generation of questions for personalized feedback [8] to the solving of math equations with models such as GPT-3 [9]. GPT-3 has also been used to simulate user interactions to train automated tutoring systems [10]. In a related field of work, Intelligent Teaching Assistant Systems (ITAs) are designed to help both learners and teachers [5, 6, 7]. Deep learning methods have been used to design ITAs to detect student distraction and notify teachers to improve lesson effectiveness [16].

2.2. Applications of Large Language Models

LLMs have been applied to a wide range of contexts within the HCI community to design assistive technologies that generate human-like logical and intellectual responses to prompts. These models have been extensively applied to support writing tasks [17] such as story writing [18], user-adapted semantic description generation [19] and help curate emails [20]. In other domains, such as in helping users with their mental health, researchers have shown the potential of using LLMs for improving the mood of users [11] and helping computer science undergraduate students manage their mental well-being through interactions with chatbots [21].

Through this work, our aim is to explore the design space of developing ITSs with the use of large language models, while involving the relevant stakeholders (e.g. instructors, students, researchers, etc.) in the design process for a human-in-the-loop approach.

3. Design of QuickTA Framework

In this section, we cover some details of the design process followed to develop the initial prototype for QuickTA. To help formulate our system's design, we conducted a literature review (covered in Section 2) and preliminary semi-structured interviews with computer science students of a North American public university to understand their needs for getting help in their moment of learning.

3.1. Formative Interviews

We started by asking interviewees to recall any course they took in the past that had a "programming" component to it (e.g., introduction to programming, database systems, etc.). Then open-ended questions were asked to identify different contexts in which they sought or did not seek help from the instructor and teaching assistants. They were then asked to share about the different tools they used for help while solving problems or the different communication channels (e.g., classroom forums, and emails) they used while reaching out to others for help. Special emphasis was placed on identifying the different contexts in which students preferred one of the options over the others. They were also asked about specific prompts that they think will be helpful when they are stuck while solving assignments.

In addition, we ask some targeted questions to help in the design of an automated system to provide them with support. They were asked to think aloud about the characteristics of this ideal student-assisting tool, as well as the different contexts in which they would use this tool over a human teaching assistant (TA). Finally, they were asked to share any "acceptable" or "unacceptable" mistakes they thought this ideal system could make.

3.2. Design Goals

Guided by our formative interviews with students and prior literature, we formulate three design goals to guide our design of the QuickTA system.

D1: Always accessible channel / medium to ask questions. One of the common themes that emerged during the interviews was that students "rarely" or "not often" sought help from existing channels to get help in the course. When it came to asking for help from instructors or TAs, one of the students mentioned *"I'm very busy with other courses or the time usually conflicts with a class (near midterm and exam time professors usually have more students in queue during office hours), or I'm not too sure what to ask."*

D2: Ask questions to understand the context before generating hints. QuickTA should be able to maintain context and be able to ask questions when students are unable to articulate their problems. When it came to asking questions on a classroom forum (called Piazza), one of the students mentioned *"The problem with Piazza is that sometimes when you need to give a long context to the problem you're trying to solve it ends up confusing the person answering and is difficult to follow."*. Some of the students also highlighted that they had problems in articulating their questions and said *"Not sure how to summarize certain questions"*.

D3: Mixed initiative tutoring. The system is based on ITS's core concept of using artificial intelligence to assist learning, which will introduce uncertainties. In order to balance automation and controllability [22], a mixed-initiative approach should be used to include humans (instructors and students) to identify and correct errors.

3.3. QuickTA Framework

For designing a system to fulfill these design goals, we propose a full-stack system with separate student and instructor views. Help is provided to students through a conversational agent that could be accessed at any time by the student (D1). A large language model (e.g., GPT-3), capable of maintaining context and retrieving hints [23], is used to generate the interaction with the student (D2). The instructor can program the LLM, based on usage statistics and feedback from the students (D3).

3.3.1. Interface

Some views of QuickTA are shown in Figure 1, 2 and 3. Now we look at each of these in more detail.

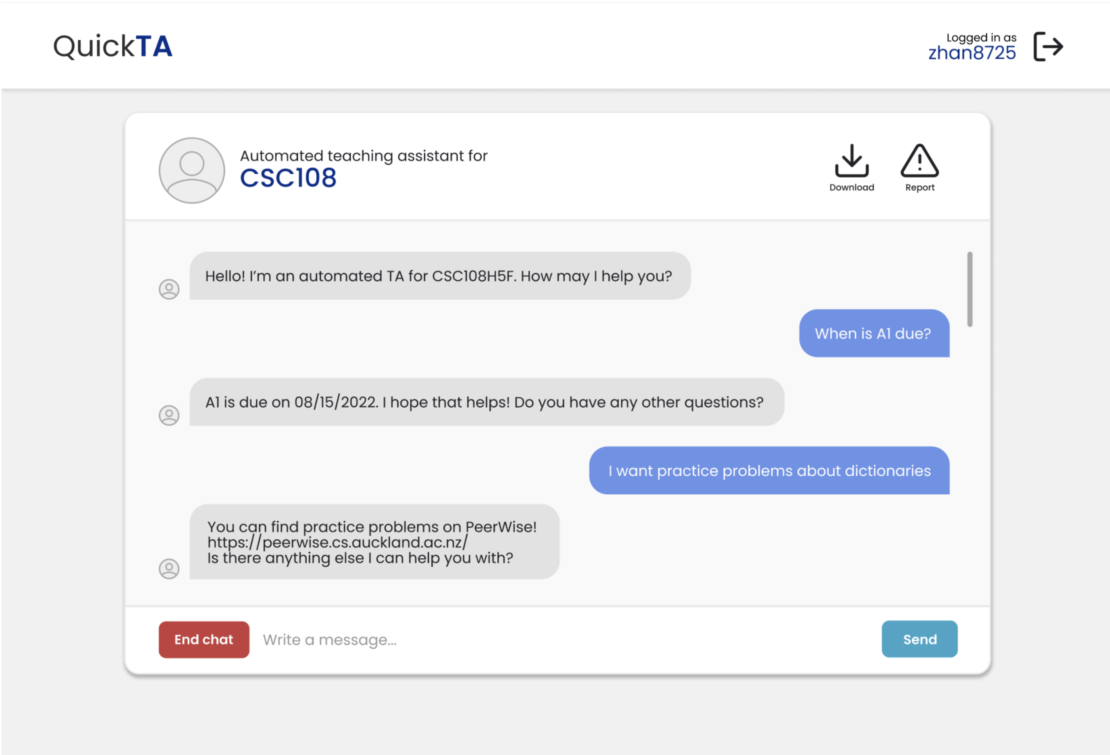

Figure 1: Chat window for students to interact with the conversational agent.

3.3.2. Instructor View

QuickTA provides instructors with the ability to configure different modular components of the interaction according to their requirements. For example, depending on the course, the instructor can modify the "Prompt" parameter (see Figure 2) to provide appropriate help to students. In the context of a database course, the prompt could be *"The following is a* *conversation with a database instructor. The instructor helps human solve assignment problems related to databases. The instructor does NOT explicitly give the solution or write the SQL query, but only helps brainstorm possible solutions.".* The student-QuickTA interaction generated in Section 3.3.3 is based on this prompt (the DDL commands to generate tables were also provided as input).

The data analytics view (see Figure 3) provides a summary of all interactions students have had with a particular LLM. Some metrics include the average rating, the average response rate, and the number of reported conversations. Flagged dialogs are also displayed in the dashboard to make future adjustments to the model.

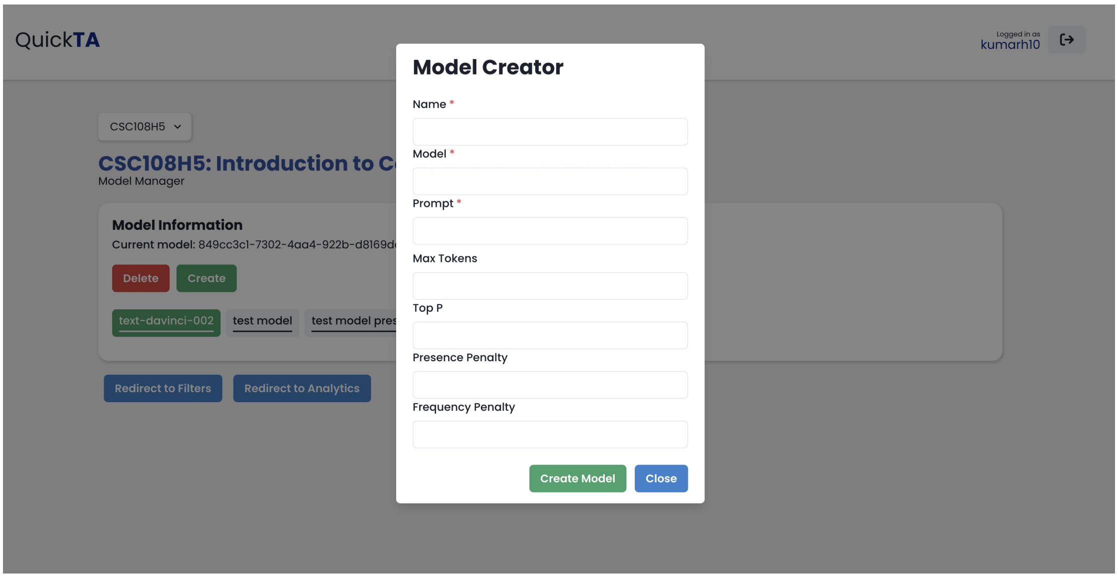

Figure 2: Instructor window for programming/fine-tuning LLMs - input prompts, hyperparameters, etc.

3.3.3. Student View

QuickTA provides support to students in the form of conversations in a chat window. Students have the option to report any chat they find inappropriate or not satisfactory enough. They also have the option to download the chat log for future reference. A sample conversation based on the prompt in Section 3.3.2 would look like the following:

- I need help solving my assignment.
- Okay, what is your assignment about?
- it's related to database.
- What specifically do you need help with?
- I am trying to write the SQL query to identify total number of students who took ship route #1 on Sept 4th, 2019. But I don't know where to start from.
- Well, one approach would be to start by looking at the table structure and identifying which tables and columns contain the relevant data. In this case, it looks like "Take" table

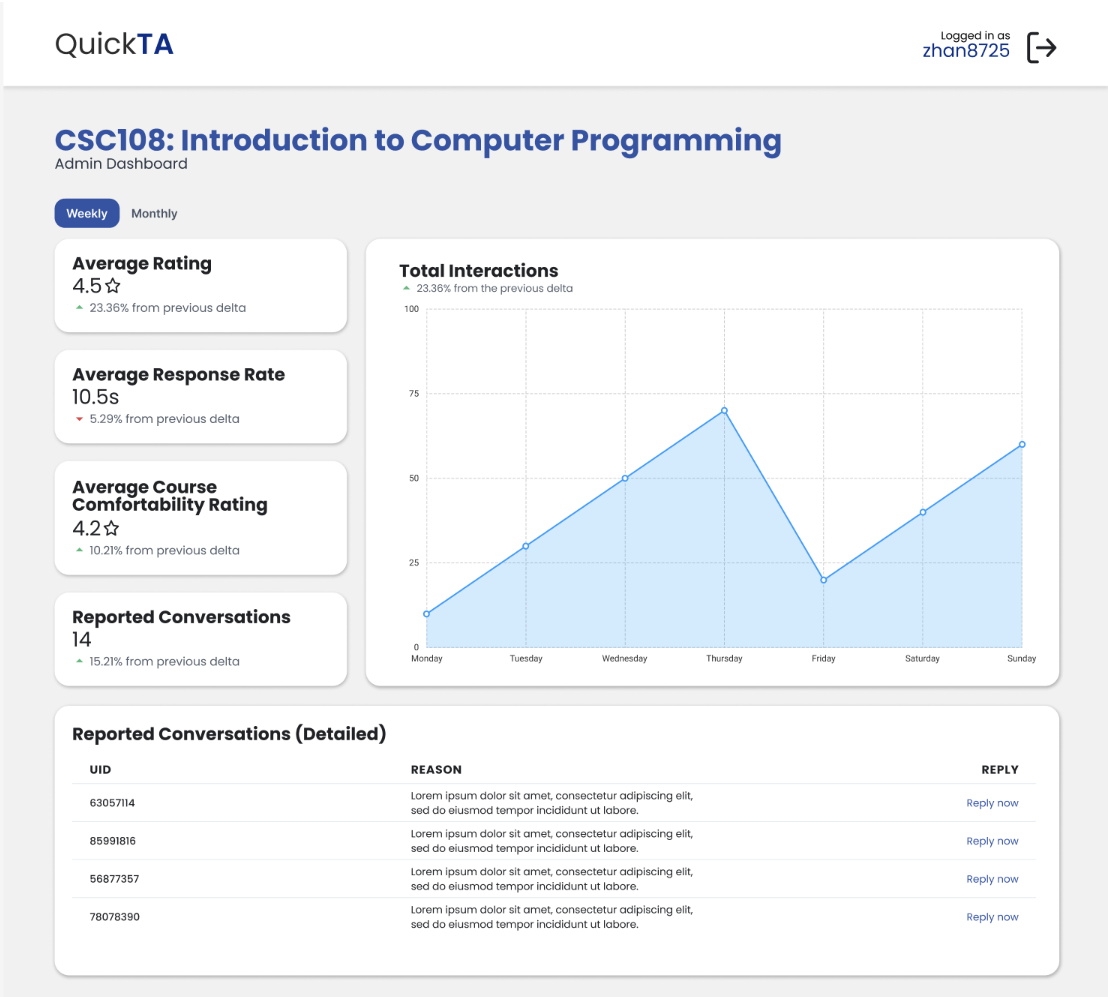

Figure 3: Dashboard for instructors to visualize the interactions students are having with QuickTA.

would be a good place to start, since it contains information about which students took which routes on which dates.

4. Conclusion

In this work, we share a brief overview of our ongoing work to design an automated tool to help students with their learning. We are in the process of conducting studies and interviews to evaluate the effectiveness of the tool. Through this submission, we hope to initiate discussions and collect feedback from other researchers present in the workshop who are working on similar topics.

Our intention is to incorporate QuickTA into computer science courses in the upcoming semesters to improve the learning experience of undergraduate students while they are thinking about the material at hand. Additionally, we intend to collect feedback on the user interface and user experience of QuickTA to better understand student needs.

Acknowledgments

We would like to thank the members of the QuickTA development team: Milind Vishnoi (team lead), Marko Choi, Hammad Sheikh, Joyce Wong, and Brian Zhang, as well as, Professor Lisa Zhang for their support in this research.

References

[1] J. R. Carbonell, Ai in cai: An artificial-intelligence approach to computer-assisted instruction, IEEE transactions on man-machine systems 11 (1970) 190–202.

[2] R. A. Sottilare, A. Graesser, X. Hu, H. Holden, Design recommendations for intelligent tutoring systems: Volume 1-learner modeling, volume 1, US Army Research Laboratory, 2013.

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

[4] T. Carmichael, M. J. Blink, J. Stamper, Tutorgen scale®-student centered adaptive learning engine (????).

[5] K. Yacef, Intelligent teaching assistant systems, in: International Conference on Computers in Education, 2002. Proceedings., IEEE, 2002, pp. 136–140.

[6] S. Yu, Y. Lu, Intelligent teacher assistant, in: An Introduction to Artificial Intelligence in Education, Springer, 2021, pp. 101–123.

[7] G. Siemens, R. S. J. d. Baker, Learning analytics and educational data mining: Towards communication and collaboration, in: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, LAK '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 252–254. URL: https://doi.org/10.1145/2330601.2330661. doi:10.1145/2330601.2330661.

[8] D. Kulshreshtha, M. Shayan, R. Belfer, S. Reddy, I. V. Serban, E. Kochmar, Few-shot question generation for personalized feedback in intelligent tutoring systems, arXiv preprint arXiv:2206.04187 (2022).

[9] M. Zong, B. Krishnamachari, Solving math word problems concerning systems of equations with gpt-3, in: Proceedings of the Thirteenth AAAI Symposium on Educational Advances in Artificial Intelligence, 2022.

[10] P. BRICMAN, M. SABATELLI, How can this be split into parts? training intelligent tutors on user simulators using reinforcement learning (2022).

[11] H. Kumar, I. Musabirov, J. Shi, A. Lauzon, K. K. Choy, O. Gross, D. Kulzhabayeva, J. J. Williams, Exploring the design of prompts for applying gpt-3 based chatbots: A mental wellbeing case study on mechanical turk, 2022. URL: https://arxiv.org/abs/2209.11344. doi:10.48550/ARXIV.2209.11344.

[12] H. S. Nwana, Intelligent tutoring systems: an overview, Artificial Intelligence Review 4 (1990) 251–277.

[13] K. R. Koedinger, J. R. Anderson, W. H. Hadley, M. A. Mark, et al., Intelligent tutoring goes to school in the big city, International Journal of Artificial Intelligence in Education 8 (1997) 30–43.

[14] G. S. Plano, The effects of the Cognitive Tutor Algebra on student attitudes and achievement in a 9 th-grade algebra course, Seton Hall University, College of Education and Human Services, 2004.

[15] R. Vesselinov, J. Grego, Duolingo effectiveness study, City University of New York, USA 28 (2012).

[16] Z. Zhou, An intelligent teaching assistant system using deep learning technologies, in: Proceedings of the 2020 5th International Conference on Machine Learning Technologies, ICMLT 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 18–22. URL: https://doi.org/10.1145/3409073.3409079. doi:10.1145/3409073.3409079.

[17] B. Robinson, Speculative propositions for digital writing under the new autonomous model of literacy, Postdigital Science and Education (2022) 1–19.

[18] A. Yuan, A. Coenen, E. Reif, D. Ippolito, Wordcraft: Story writing with large language models, in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 841–852.

[19] J. Sevilla Salcedo, L. Martín Galván, J. C. Castillo, Á. Castro-González, M. A. Salichs, User-adapted semantic description generation using natural language models, in: International Symposium on Ambient Intelligence, Springer, 2023, pp. 134–144.

[20] E. BUEHLER, P. CLARY, A. COENEN, A. DONSBACH, T. N. HORNE, M. LAHAV, R. MACDONALD, R. B. MICHAELS, A. NARAYANAN, J. R. PUSHKARNA, et al., Lampost: Design and evaluation of an ai-assisted email writing prototype for adults with dyslexia (2022).

[21] H. Kumar, K. Yu, A. Chung, J. Shi, J. J. Williams, Exploring the potential of chatbots to provide mental well-being support for computer science students, in: Proceedings of the 51st ACM technical symposium on computer science education, 2023, p. (poster) to appear. URL: https://doi.org/10.1145/3545947.3576285.

[22] Q. Roy, F. Zhang, D. Vogel, Automation accuracy is good, but high controllability may be better, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–8.

[23] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, L. Wang, An empirical study of gpt-3 for few-shot knowledge-based vqa, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, 2022, pp. 3081–3089.