Prompt Engineering for Large Language Models to Support K-8 Computer Science Teachers in Creating Culturally Responsive Projects

Minh Tran
ngminhtran@uchicago.edu
University of Chicago
Chicago, IL, USA

ABSTRACT

The power of large language models has opened up opportunities for educational use. In computing education, recent studies have demonstrated the potential of these models to improve learning and teaching experiences in university-level programming courses. However, research into leveraging them to aid computer science instructors in curriculum development and course material design is relatively sparse, especially at the K-12 level. This work aims to fill this gap by exploring the capability of large language models in ideating and designing culturally responsive projects for elementary and middle school programming classes. Our ultimate goal is to support K-8 teachers in effectively extracting suggestions from large language models by only using natural language modifications. Furthermore, we aim to develop a comprehensive assessment framework for culturally responsive AI-generated project ideas. We also hope to provide valuable insight into teachers' perspectives on large language models and their integration into teaching practices.

CCS CONCEPTS

• **Social and professional topics → Computing education**; *K-12 education*; • **Computing methodologies → Artificial intelligence**; Natural language generation.

KEYWORDS

large language models, culturally responsive pedagogy

ACM Reference Format:
Minh Tran. 2023. Prompt Engineering for Large Language Models to Support K-8 Computer Science Teachers in Creating Culturally Responsive Projects. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.2 (ICER '23 V2), August 7–11, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3568812.3603453

1 MOTIVATION & KEY IDEAS

Culturally responsive pedagogy (CRP) has emerged as an essential and effective approach to enhancing equity and inclusion for students. The primary goal of CRP is to create a learner-centered environment that acknowledges and leverages the cultural backgrounds and experiences of diverse student populations, thus facilitating and supporting their overall learning experiences [6]. In the realm of Computer Science (CS) education, culturally responsive computing (CRC) centers on the resonance of computing instructional materials with the interests, values, and identities of students coming from a broad spectrum of cultures [4, 13].

In response to the growing demand for culturally responsive teaching, leading organizations and research groups have developed comprehensive computing platforms and curricula tailored to elementary and middle school teachers venturing into this field [1, 5, 7, 11]. Moreover, to effectively implement culturally responsive pedagogy at the classroom level, teachers, being intimately familiar with their students, are ideally positioned to lead the way. However, the process of creating culturally responsive teaching materials can be daunting and time-consuming, especially in large-sized class settings.

Over the past few years, the significant advancements of large language models (LLMs) have paved the way for their integration into computing education. Recent research has shown encouraging outcomes of using LLMs to enhance programming learning and teaching experiences [3, 8, 9, 12, 14]. Nevertheless, there remains a gap in examining the potential of LLMs to support educators in curriculum design and exploring how these models can be integrated into pedagogical practices. Furthermore, no research into the possibilities and limitations of using LLMs within the domain of K-12 computing education has been conducted.

The objective of this work is to explore prompt engineering for the ideation and design of culturally responsive computing instructional materials using large language models. In order to simplify the process, we leverage the Scratch Encore curriculum [5], assisting teachers in modifying existing projects rather than creating new projects. We are particularly interested in addressing the following research questions:

RQ1: To what extent do LLMs currently support the ideation and design of culturally responsive projects?

RQ2: How can natural language modifications bias the models towards consistently suggesting project ideas that culturally and technically match our expectations?

RQ3: How should we evaluate project ideas generated by LLMs? Should this task be performed by computing education researchers or teachers, considering their respective expertise and perspectives?

ICER '23 V2, August 7–11, 2023, Chicago, IL, USA

ACM ISBN 978-1-4503-9975-3/23/08.
https://doi.org/10.1145/3568812.3603453

2 RELATED WORKS

2.1 Curriculum

Scratch Encore is a culturally-relevant intermediate curriculum designed for upper-elementary school and middle school students with at least one year of foundational coding experience. It is comprised of 15 learning modules, each introducing a different topic in CS. At the module level, there are three different strands, each covering exactly the same technical materials while showcasing disparate example projects drawn from activities in youths' everyday life [5].

2.2 Large Language Models in Computer Science Education

Large Language Models (LLMs) are advanced AI systems capable of processing and generating human-like text based on vast amounts of training data. These models employ deep learning techniques to understand and produce coherent and contextually relevant responses when prompted with natural language inputs, making them a promising tool for educational support. In CS classroom, LLMs have shown potential in helping students write code [14], generating code explanations [9], parsing non-compiling code and generating programming error messages [8]. While primarily employed to facilitate students' learning experiences, LLMs are beginning to be used to assist instructors in creating programming assignments [12]. However, little is known about their capabilities in designing curriculum and enhancing pedagogical practices. This knowledge gap is especially pronounced in K-12 CS education, where no research has been conducted, emphasizing the need for further investigation in this field.

2.3 Prompting Large Language Models

An approach to effectively engage with LLMs is through prompting (i.e., natural language to steer the models' responses). While prompting may seem as simple as conversing with a human, devising prompt strategies that are both effective and generalizable is challenging, particularly for non-AI-experts. This process is very time-consuming as it involves extensive trial and error and iterative experimentation to assess the effectiveness of different prompts on individual input/output pairs [15]. Within CS education, researchers have investigated how students can learn to modify natural language descriptions of introductory programming problems to guide models into generating solutions [3]. Our work extends prior research on supporting non-experts, specifically teachers, in prompting engineering for LLMs.

3 RESEARCH APPROACH

3.1 Prompt Design

A large design challenge in this work is determining how to interact with LLMs in a way that can consistently generate project ideas culturally aligning with teachers' expectations while adhering to the technical requirements of each Scratch Encore module. To direct the models to provide responses that are technically suitable for a particular module, the prompt structure incorporates three key elements: (1) general project requirements, (2) a detailed description of an example project within that module, and (3) a question for the model(s) to suggest a similar project related to a specific topic. To streamline the process for teachers and minimize their time investment, we pre-define the general project requirements and example projects for each module. This allows teachers to simply choose a topic that resonates with their own classrooms and include it in the prompt.

3.2 Study

Our study will focus on five Scratch Encore modules. We have selected GPT-3 [2], a publicly-available large language model, and planned to explore it on OpenAI Playground, a widely used prompt design platform among non-experts. For each module, we will create an initial prompt and iteratively apply it to 30 different topics. Through this process, we will pick five topics for which the model produces project ideas of poor quality to further examine the resulting input/output pairs. We will identify characteristics of successful/unsuccessful responses and refine the prompt iteratively until it gives consistently good results. This leads to the challenge of evaluating the responses and assessing the effectiveness of the prompts. We will qualitatively analyze the GPT-generated ideas using the following criteria:

(1) *Implementation feasibility:* can the idea be implemented in Scratch?
(2) *Technical complexity:* does implementing the idea require CS knowledge beyond the given module's coverage?
(3) *Cultural specificity:* is the idea specific about a particular culture?
(4) *Age-appropriateness:* is the idea appropriate for K-8 students?

We will invite teachers who have experience teaching Scratch Encore in their classroom, to participate in co-design sessions and a professional development workshop. Before scaffolding teachers with our pre-defined prompts, we will conduct a survey to assess their familiarity with ChatGPT [10], a chatbot powered by GPT-3. Additionally, we will engage them in a prompt design activity to gain insight into their perspectives on ChatGPT and its potential implementation in their teaching practices, as well as the intuitions they bring to prompt design. Subsequently, we will facilitate teachers in utilizing our prompts to conceptualize culturally-relevant Scratch Encore projects, followed by evaluating the project ideas suggested by ChatGPT. Our objective is to compare how teachers assess those ideas with our previous assessment from researcher perspectives. By doing so, we aim to identify any discernible differences between the two assessments and propose an effective evaluation process for GPT-generated culturally responsive project ideas.

4 CURRENT PROGRESS

We created an initial set of prompts for Module 2-6 of the Scratch Encore curriculum and tested it with ten topics, using `text-davinci-003` model of GPT-3. We selected `text-davinci-003` because it was the newest and the most capable amongst the GPT-3 model family at the time of the study. Our experiments yielded promising project ideas that satisfy the technical requirements of the modules and are feasible to be deployed in Scratch. However, a number of the model's responses are not culturally specific. For example, when asked to generate a project idea related to *Vietnamese cuisine*, it suggests *"a Vietnamese traditional dish"* or *"a Vietnamese chef cooking in a traditional kitchen"*. In some cases, GPT-generated ideas deviated significantly from the given topic. For example, it proposes a project about *Chinese music* and *Japanese music* while the topic requested is *Asian cuisine*.

In 2022-23 school year, we launched co-design sessions with four experienced Scratch Encore teachers from U.S. public school districts in Illinois, Maryland, Minnesota, and Rhode Island. They had been involved in prior participatory design sessions about customizing Scratch Encore lessons using static scaffolds. In one session, we surveyed the teachers about ChatGPT and carried out a prompt design activity. Initially, the teachers exhibited unfamiliarity with the chatbot and expressed skepticism about integrating AI into their teaching. However, as they actively participated in the prompt design activity, their enthusiasm grew, and they gained excitement about using ChatGPT for project idea generation. The teachers were able to draw out detailed information relevant to their selected topic, but failed to direct the chatbot to generate specific project ideas. Additionally, they overlooked the technical requirements of the module they were customizing, seemingly assuming that the model would inherently possess a comprehensive understanding of the intricacies of the Scratch Encore curriculum.

Our next step is to conduct further experiments with GPT-3 to enhance our prompt strategies and refine our assessment framework. Additionally, we are preparing to launch a professional development workshop in Summer 2023, for which we have successfully recruited a larger cohort of teachers. Finally, we plan on submitting a paper to SIGCSE 2024, detailing our experimental findings and sharing the valuable insight gained through the professional development initiative.

REFERENCES

[1] Kara Beason, James B Fenwick Jr, and Cindy Norris. 2020. Introducing middle school students to computational thinking with the CS first curriculum. In *Proceedings of the 2020 ACM Southeast Conference*. 10–17.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[3] Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with Copilot: Exploring prompt engineering for solving CS1 problems using natural language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 1136–1142.

[4] Ron Eglash, Juan E Gilbert, and Ellen Foster. 2013. Toward culturally responsive computing education. *Commun. ACM* 56, 7 (2013), 33–36.

[5] Diana Franklin, David Weintrop, Jennifer Palmer, Merijke Coenraad, Melissa Cobian, Kristan Beck, Andrew Rasmussen, Sue Krause, Max White, Marco Anaya, et al. 2020. Scratch Encore: The design and pilot of a culturally-relevant intermediate Scratch curriculum. In *Proceedings of the 51st ACM technical symposium on computer science education*. 794–800.

[6] Geneva Gay. 2018. *Culturally responsive teaching: Theory, research, and practice.* teachers college press.

[7] Filiz Kalelioğlu. 2015. A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior* 52 (2015), 200–210.

[8] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 563–569.

[9] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from using code explanations generated by large language models in a web software development e-book. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 931–937.

[10] OpenAI. 2023. *Introducing ChatGPT.* https://openai.com/blog/chatgpt

[11] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[12] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*. 27–43.

[13] Kimberly A Scott, Kimberly M Sheridan, and Kevin Clark. 2015. Culturally responsive computing: A theory revisited. *Learning, Media and Technology* 40, 4 (2015), 412–436.

[14] Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*. 1–7.

[15] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–21.