Generation and Evaluation of a Culturally-Relevant CS1 Textbook for Latines using Large Language Models

Ismael Villegas Molina
University of California, San Diego
La Jolla, California, USA
isvilleg@ucsd.edu

Audria Montalvo
University of California, San Diego
La Jolla, California, USA
ansaravi@ucsd.edu

Shera Zhong
University of California, San Diego
La Jolla, California, USA
s3zhong@ucsd.edu

Mollie Jordan
North Carolina State University
Raleigh, North Carolina, USA
mcjordan@ncsu.edu

Adalbert Gerald Soosai Raj
University of California, San Diego
La Jolla, California, USA
asoosairaj@ucsd.edu

ABSTRACT

In the United States, culturally relevant computing (CRC) is one of the most popular pedagogical implementations for Latin American (Latine) students. Culturally-relevant learning resources are a valuable tool for implementing CRC. However, the traditional method of creation and maintenance of textbooks takes a significant amount of time and effort. Given the duration required for textbook production, the development of culturally-relevant learning resources may become lengthened, as it requires close attention both on the material and the incorporation of cultural referents. In order to accelerate the process, we used the advancement of large language models (LLMs) to our advantage. Through prompt engineering, we created a series of prompts to produce a textbook for an introductory computer science course (CS1) that incorporates Latine culture. This textbook was evaluated on metrics regarding sensibility, correctness, readability, linguistic approachability, appropriateness of examples, and cultural relevance. Overall, the generated textbook was mainly sensible, correct, readable, and linguistically approachable. Code examples were not always appropriate due to the usage of libraries that are not typically used in a CS1 course. The cultural relevance was apparent, but it often included surface-level cultural referents. The main incorporation of culture was through geographical locations and people's names. This suggests that the use of LLMs to generate textbooks may serve as a valuable first step for writing culturally-relevant learning resources. Though this study focuses on Latines, our results and prompts may be applicable for generating culturally-relevant CS1 textbooks for other cultures.

CCS CONCEPTS

• **Social and professional topics** → **Race and ethnicity**; **Computer science education**.

KEYWORDS

Latine, Latinx, Latina, Latino, large language models, culturally relevant resources, computer science textbook, resource generation

ITiCSE 2024, July 8–10, 2024, Milan, Italy

ACM ISBN 979-8-4007-0600-4/24/07.
https://doi.org/10.1145/3649217.3653600

ACM Reference Format:
Ismael Villegas Molina, Audria Montalvo, Shera Zhong, Mollie Jordan, and Adalbert Gerald Soosai Raj. 2024. Generation and Evaluation of a Culturally-Relevant CS1 Textbook for Latines using Large Language Models. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649217.3653600

1 INTRODUCTION

Computer science remains a consistently lucrative career choice, as indicated by employer salary data [10]. The field is experiencing steady growth in enrollment, with projections for 2029 suggesting a faster expansion compared to other domains [31, 39, 47]. However, a well-known and significant underrepresentation issue persists, particularly affecting women and the Black, Latine, Native American, and Pacific Islander (BLNPI) communities [35, 36, 46]. Despite Latin Americans (Latines) being the second-largest racial or ethnic group in the United States, recent longitudinal studies reveal a persistently low participation rate in computer science among this demographic [4, 28]. This limitation hinders their access to the abundant career opportunities, competitive pay rates, and social advantages offered by the field. A study conducted by Lewis et al. sheds light on the challenges faced by Latine students in computer science, including a low sense of belonging, negative perceptions regarding identity, and the importance of community in the field [27]. Addressing these challenges is crucial for fostering a more inclusive and supportive environment, aligning with our continuous efforts to cultivate diversity in the computing field.

In a literature review conducted by Villegas Molina et al. which researched all the papers that specifically studied Latines in computing, one of the results was that many popular pedagogical practices (e.g., Pair Programming, Peer Instruction) were not very well researched [40]. The most popular pedagogy that was conducted with U.S. Latines in computing was Culturally Relevant Computing (CRC). CRC for Latines has shown to have positive effects for their computing experience and learning (see Section 2.2).

In creating CRC resources, one can turn to the use of large language models (LLMs). The use of artificial intelligence to enhance educational resources and practices has been on the rise in computing and beyond [21, 22, 26, 34, 43] with a recent study suggesting LLM-generated resources may serve as viable materials in certain contexts [7]. The study by Denny et al. points towards the potential

for LLMs to improve the accessibility and scalability of high-quality educational content, while reducing the burden on educators [7]. Work by Jordan et al. using LLMs to generate programming exercises in different natural languages has shown that Spanish (a prominent language in Latin America) outperformed exercises in English with respect to the exercises being mostly sensible, readable, and containing the proper programming concepts [20].

Therefore, we propose the generation of CRC resources for U.S. Latines using LLMs to aid the resource creation process. We believe that our approach will make the process of creating culturally-relevant resources more accessible to educators, and in turn increase the availability of culturally relevant computing education for many Latine students in the United States.

2 RELATED WORK

2.1 Textbook Writing

Writing a textbook (excluding the publishing phase) can span from one to over five years [32, 37]. It requires emphasis on readability, sensibility, assuring that it is written in such a way the content is appropriate to the intended audience, in addition to the pedagogical framework that can benefit your audience [37]. Overall, generating textbooks is complex and the introduction of cultural relevance might lengthen the process as it requires all the traditional requirements of textbook writing with the added complexity of incorporating meaningful cultural engagement. Our study leverages the utilization of LLMs to generate a culturally-relevant introductory computer science textbook – reducing the textbook process from the magnitude of years to a magnitude of minutes.

2.2 Culturally Relevant Computing

Culturally Relevant Computing (CRC) stems from Culturally Relevant Pedagogy – a theoretical framework developed by Ladson-Billings [24]. The framework has three criteria: students should 1) be able to achieve academic success regardless of their differences of category (e.g., race, gender, language, community), 2) develop and maintain cultural competence by recognizing and honoring their own and others' cultural beliefs and practices, and 3) develop a broader sociopolitical consciousness to critique societal norms and institutions that produce social inequality [24].

A literature review was conducted by Villegas Molina et al. in order to understand the current body of work regarding U.S. Latines in computing [40]. Among the analysis, they studied the pedagogical interventions that were conducted specifically to a Latine classroom. They found that CRC is the predominant pedagogy studied with Latines [40]. CRC implementation spanned various themes such as bilingual education [14, 33, 41, 42, 45], culturally-relevant topics [1, 12, 33, 38], social justice [1, 11, 33, 38], family-based approaches [9, 11, 45], and community involvement [11]. CRC was found to increase computing interest [12, 14, 45], engagement [9, 33, 41], confidence [14, 45], social value towards computing [1, 14], computing comprehension [33], motivation [1], and computing awareness [9]. Due to these benefits, we aimed to leverage the use of LLMs to generate culturally-relevant learning resources for U.S. Latines in an effort to serve this underrepresented community in computing.

2.3 LLMs and Cultural Relevance

A study by Cao et al. highlighted that ChatGPT shows a strong cultural alignment to the United States and adapts less effectively to other cultural contexts – specifically Chinese, German, Japanese, and Spanish cultures [5]. This finding is not unique to ChatGPT; similar findings have been found in other LLMs [2]. Efforts have been made to increase cultural relevance within LLM by fine-tuning the system with parallel data, leading to an increase in Chinese cultural relevance [44]. Our work will extend the analysis of cultural relevance of LLMs via the Latin American culture in an introductory computing textbook context.

2.4 LLMs for Resource Generation

LLMs has seen an increased adoption to assist in the learning process through resource generation [8, 13, 25], including in Latin America [19], with students reporting planned use of ChatGPT for educational purposes [18]. Within the usage of LLMs for computing education, there has been a focus on assessments of individual code snippets or explanations [7, 20, 29, 34]. These assessments have found that LLM-generated learning resources are mainly sensible and readable [20, 34] and are not significantly different from human-generated resources [7, 18]. To the best of our knowledge, our research is the first to investigate the efficacy of LLMs in generating a CS1 textbook with Latine culture incorporated throughout.

3 METHOD

3.1 Research Questions

The aim of this study is to evaluate 1) the efficacy of LLMs in creating a culturally-relevant introductory computing textbook and 2) the cultural relevance used by LLMs in such textbooks. To understand this, we asked the following research questions:

(1) How effective are LLMs at creating a culturally-relevant introductory computing textbooks for U.S. Latines?
(2) How do LLMs incorporate cultural relevance for U.S. Latines in such a textbook?

The "effectiveness" of the LLM was evaluated on metrics of sensibility, readability, cultural relevance, correctness, linguistic approachability, and appropriateness level of code examples. These metrics are further explained in Section 3.3.

3.2 Prompt Engineering

We initially developed our prompts using OpenAI's ChatGPT web interface. In an effort to automate the process, we switched to the GPT-3.5-Turbo API and wrote a program that prompts the system chapter by chapter. We implemented OpenAI's suggestions regarding prompt engineering – specifically writing clear instructions and splitting complex tasks into simpler subtasks [16]. In order to write clear instructions, we implemented the tactic of "including details to our query for more relevant answers" throughout our prompting process [16].

3.2.1 System Context. One of the tactics listed for writing clear instructions was to ask the LLM to adopt a persona [16]. We accomplished this by using the system message to specify that the LLM educates introductory computer science to undergraduate students.

In an effort to incorporate cultural relevance for a Latine audience, we explicitly added Latin American culture to the description of the persona. We chose to generate a textbook using the Python programming language, as it is syntactically simple and is used at our institution's introductory computing course. The context is as follows: *"You are an expert educator to undergraduate students in introductory computer science using Python. All responses should expertly explain Python topics using accessible, simple language. You write textbooks and each chapter should include detailed explanations of topics, relevant examples, and exercises. You are Latin American. All your responses should be from a Latin American perspective and incorporate Latin American culture."*

Listing 1: Chapter Prompts

```
for chapter in chapters:
  my_prompt = "Write a full and detailed chapter in Python
       on " + chapter + " for an introductory computer
       science textbook. This will be chapter number " +
       str(chapter_number) + ". The chapter will cover " +
       ", ".join(chapters[chapter]) + ". Explain each
       Python topic with great detail and accessible,
       simple language. Provide programming examples
       whenever possible. Provide practice exercises for
       each topic. Make the content and exercises tailored
       to a Latin American context. Some examples and
       exercises should incorporate Latin American culture.
       Do not include descriptions of how the Latin
       American culture made the content more engaging. Do
       not reference the fact that you incorporated Latin
       American culture. Do not ask the reader to
       incorporate Latin American culture, rather the
       textbook should incorporate Latin American culture
       organically without calling attention to it."
```

3.2.2 Chapter Generation. In order to implement the best practice of splitting of complex tasks to simpler subtasks [16], we leveraged how textbooks are split into chapters by nature. This lead us to split our tasks into chapter-specific outputs, rather than attempting to produce the entire textbook all at once. The chapter prompts provided to the LLM can be found in Listing 1. We generated ten chapters spanning the following topics: 1) Expressions and Functions, 2) Conditionals, 3) Loops, 4) References, Objects, and Methods, 5) Coding Best Practices and Debugging, 6) Nested For Loops and 2D Lists, 7) Image Manipulation, 8) Dictionaries, 9) Data Processing, and 10) Final Review. We chose these topics as they reflected the introductory course topics at our institution.

We placed all chapters into a Python dictionary, where each key was the chapter name and the value was a list of topics within the chapter to be written. We iterated through each chapter and wrote a prompt to produce the culturally-relevant chapter with all of its subtopics represented. The instruction to "not include how incorporating Latin American culture would make the content more engaging" was explicitly added as these sorts of statements were present in initial generations of the textbook and produced results that sounded forced. It was also specified that the response should "not ask the reader to incorporate Latin American culture" as the LLM would typically write exercises asking readers to incorporate Latine culture (e.g., write a list with your five favorite Latine soccer players). By having readers incorporate Latine culture, it might not make the textbook accessible to a general audience.

3.3 Textbook Evaluation

We evaluated the culturally-relevant textbook through a similar process ran by Jordan et al.. The evaluation team consisted of the first two authors of this study. The first and second authors had previously served as a teaching assistant and a tutor, respectively, to the CS1 Python course over several academic terms. Both evaluators are Latine and were tasked to evaluate the culturally-relevant textbook as they had the cultural background to evaluate the text with respect to cultural relevance. The evaluators were given the textbook alongside an evaluation table (see Table 1) adapted from Jordan et al.'s original.

The criteria of **sensibility**, **readability**, **cultural relevance** from Jordan et al.'s original rubric was used. Three aspects to our evaluation were added. We added **correctness** to see if the material and code examples were correct. We felt this was a necessary aspect to assess as textbooks are used as learning material which should be correct to avoid confusing learners. We added **linguistic approachability** to see how accessible the language is in the textbook. This is distinct from readability, as a text could be readable and describe a sensible topic, but use verbose wording and jargon that could confuse non-native English speakers – where over half of U.S. Latines speak Spanish at home [23]. We also added **appropriateness level of code examples** in order to verify that the examples that were being presented in a CS1 textbook are not out of scope. Each chapter went through the assessment separately to evaluate each prompt output from the LLM. For each aspect, evaluators chose "Yes", "No", or "Maybe". They described their reasoning if they chose "Maybe" and could write notes for any response. The two evaluators independently assessed each chapter in the textbook. Following Jordan et al.'s process, the evaluation team joined to discuss any conflicts on assessments and form a consensus on a Yes/No answer for each metric. The researchers shared their notes to make an informed decision.

We expanded on the cultural relevance aspect by adapting the Cultural Relevance Evaluation and Assessment Tool [17]. Our adapted version focused on the second criterion of the CREAT tool – providing opportunities for developing and maintaining cultural competence in relation with oneself or others [17]. This tool was previously used to evaluate the cultural relevance of learning materials found online. This tool evaluates cultural relevance through four levels. A detailed explanation of this evaluation tool's levels is found in Table 2. This metric was used by the evaluation team to analyze all examples and exercises provided by GPT-3.5-Turbo for the textbook. Examples are code snippets as part of explanation, while exercises are problems for students to practice. A total of 37 examples and 48 exercises were generated and evaluated. In order to assure evaluation consistency of cultural relevance, we calculated the interrater reliability across all examples and exercises using the Cohen's kappa statistic. Our Cohen's kappa value was 0.815 which translates to "strong agreement" [30]. In an effort to identify how the LLM incorporates Latine culture, the evaluation team performed thematic analysis [3] on the examples and exercises that

Table 1: Assessment rubric used to evaluate textbook chapters

Aspect	Question	Options	Notes
Sensibility	Does the chapter clearly state the information to the reader?	Yes/No/Maybe	
Correctness	Does the chapter not make mistakes in the material presented?	Yes/No/Maybe	
Readability	Is the chapter grammatically correct? Is it easy to read?	Yes/No/Maybe	
Linguistic Approachability	Does the chapter contain simple English words to help students understand the material?	Yes/No/Maybe	
Code Appropriateness	Are the examples in the chapter at the expected level of the textbook's audience?	Yes/No/Maybe	

were labeled "culturally-relevant". The themes were initially created by the first author based off all of the examples and exercises. Together, the team discussed the themes to verify its completeness. Any themes to be revised or added were addressed and updated.

4 RESULTS

See [15] for the generated textbook using Latine culture.

4.1 RQ1: Textbook Evaluation

Of the 10 chapters evaluated, we found six (60%) to be sensible, eight (80%) to be linguistically accessible, six (60%) to be an appropriate level of code examples, and all ten (100%) to be both readable and correct (see Table 3).

4.1.1 Sensibility. Six (60%) of the generated chapters were sensible. The chapters that were not sensible were 1) Loops, 2) References, Objects, and Methods, 3) Coding Best Practices and Debugging, and 4) Final Review. Chapters assessed as "not sensible" were due to topics not being properly introduced/explained or examples not matching the topic at hand. Examples of these can be found on the chapter on "References, Objects, and Methods". When covering the scope of variables, the textbook explains the concept of global and local scopes. However, the description of global scope is as follows: "Variables defined outside of any function or class have global scope. They can be accessed from anywhere in the program." At this point in the textbook, the concept of classes had not been introduced – potentially causing confusion to a new reader. Another example of material not making sense in this chapter occurs when introducing methods. Methods are introduced by highlighting the use of dot notation following an object's reference. An example in this chapter intends to get the length of a String using the `len()` function – potentially causing additional confusion to the reader regarding methods vs. functions. In this case, the code itself is correct as it correctly produces the length of a String, however it does not make sense in the context of the topic of methods. Note that *len(s)* is a function in Python which returns the length of the sequence that is passed in as a parameter. A method is a function that belongs to an object and is called on the object (e.g., *lst.append()*).

4.1.2 Linguistic Approachability. Eight (80%) of the generated chapters were linguistically approachable. The chapters that were not linguistically approachable were 1) Loops, and 2) Nested For Loops and 2D Lists. Instances of the textbook not being linguistically approachable result from using verbose verbiage and introducing jargon without a lay description. An example of this is the introduction of loops to the reader: *For loops are used when we want to iterate over a sequence or a collection of items. The loop variable takes the value of each item in the sequence, one by one, and executes the block of code within the loop.* A more linguistically approachable version of the same text could be as follows: *We use for loops when we want to go through a list or a bunch of items one by one. The loop variable holds the value of each item in the list, and the code inside the loop runs for each item.*

4.1.3 Appropriateness level. Six (60%) of the generated chapters used an appropriate level for code examples. The chapters that did not have an appropriate level were 1) Coding Best Practices and Debugging, 2) Nested For Loops and 2D Lists, 3) Image Manipulation, and 4) Final Review. The chapters were assessed as having code that was not an appropriate level for an introductory computer science textbook, primarily when the material generated used libraries that are not typically used in an introductory setting. An example of this is the chapter on Best Practices and Debugging, where the system imported the `unittest` library to test functions. In introductory programming courses at our university, we typically do not use unit testing libraries for testing as they may introduce an additional level of complexity for students with no prior programming experience. Rather, we would prefer to have students write their own simple tests (using print statements) to verify whether their functions are working as expected.

4.1.4 Readability and Correctness. All ten (100%) of the generated chapters were deemed to be readable and correct, with no grammatical or programmatic mistakes found. Though there were instances of low linguistic approachability, the text was still readable. There were instances of code examples not making sense where they were presented in the text (See Section 4.1.1). However, the code blocks themselves were correct and solved the problem at hand.

4.1.5 Cultural Relevance. The evaluation of cultural relevance is found in Table 4. The totals for examples and exercises in the table reflect the assessments from both researchers on the evaluation team. There were no examples or exercises that fell under the -1 level of the rubric (see Table 2) – meaning there was no material that was racist or conveyed cultural supremacy. Level 0 with no cultural relevance (culturally-agnostic) has a total of 75 instances (44%) – showing that less than half of examples and exercises are culturally-agnostic. Level 1 with surface-level cultural relevance has a total of 63 instances (37%). Level 2 with deep cultural relevance has a total of 32 instances (19%). The means in Table 4 are a weighted average calculated by adding all the level values and dividing by the amount of elements (e.g., for Examples we added the 40 zeros, 26 ones, and 8 twos, then divided by 74). The cultural relevance of the examples is 0.57 while the exercises is at 0.89 – and an independent samples t-test between the means shows a statistically significant difference ($p < 0.01$). This shows that exercises are significantly more culturally-relevant than examples.

Table 2: Assessment rubric for cultural relevance of textbook chapters

Level	Definitions	Look-for
-1	Negation of any level below	Explicitly racist or praising one culture above others; cultural supremacy
0	Absence of cultural references	Cultural references not mentioned, only generic or mainstream references are present
1	Presence of cultural references for own/other underrepresented cultural groups	Meaningful objects to certain cultural groups are present, but the cultural context is not deeply embedded in the problem and can be irrelevant to the work of solving it
2	Deep and meaningful engagement with cultural references	The presence of the cultural references has a purpose and meaning behind, and they are relevant to the problem solving

Table 3: Textbook Evaluation Results for Sensibility (Sens), Readability (Read), Correctness (Correct), Linguistic Approachability (LA) and Code Appropriateness (CA)

Sens	Read	Correct	LA	CA
60%	100%	100%	80%	60%

Table 4: Textbook Evaluation Results for Cultural Relevance across Examples and Exercises Using the Rubric from Table 2

	-1 (%)	0 (%)	1 (%)	2 (%)	Mean
Examples	0 (0%)	40 (54%)	26 (35%)	8 (11%)	0.57
Exercises	0 (0%)	35 (36%)	37 (39%)	24 (25%)	0.89
Total	0 (0%)	75 (44%)	63 (37%)	32 (19%)	0.75

Table 5: Frequency of Categories of Cultural Relevance Using Levels from the Rubric from Table 2

Categories	Surface Level (1)	Deep Level (2)	Total
Geography	28	20	48
Names	16	0	16
Culinary	9	3	12
Linguistic	0	9	9
Activities	4	1	5
Items	4	0	4
Media	1	2	3
Measurements	2	1	3
Currency	0	2	2
Total	64	38	102

4.2 RQ2: Incorporation of Latine Culture

Cultural relevance evaluations are found in Table 4. Among the culturally-relevant levels (1 and 2), surface-level cultural relevance holds the majority of the examples and exercises. One of the main forms of incorporating Latine culture to examples is the use of Latin American names. For instance, an example in the textbook used a for loop to iterate through a list of student names: *students* = [“*Maria*”, “*Pedro*”, “*Luisa*”, “*Juan*”]. This incorporates the cultural signifier of common names in Latin America, but does not use it in a meaningful way for the problem at hand. A common technique for surface-level implementation of exercises was to prompt the reader to input Latine culture into the answer. Take the following exercise: *“Write a program that asks the user for their name and favorite Latin American dish, then prints a message incorporating their answers.”* In this case, we see that although the LLM incorporated the concept of Latine culture, it simply asked the reader to choose their favorite food that happens to be Latin American.

Deep cultural relevance was mainly present in exercises in the textbook. One such instance was the use of translation with dictionaries. The exercise is written as follows: *“Create a dictionary called ‘colors‘ with the following key-value pairs: “Blue” - “Azul”, “Red” - “Rojo”, “Yellow” - “Amarillo”, and “Green” - “Verde”. Prompt the user to enter a color name in English. If the color exists in the dictionary, print its translation in Spanish; otherwise, print a message stating that the color is not in the dictionary.”* This usage of Latine culture is deeper as it leverages one of the main languages in the culture (Spanish) in a meaningful way directly relating to the problem. If one were to add a new key-value pair, a correct translation is necessary in order to keep the dictionary correct (e.g., “Brown” - “Café”).

Frequencies for each category can be found in Table 5. Only examples and exercises that had surface-level (level 1) or deep engagement (level 2) were considered. An example or exercise could share across several categories, so the frequencies may exceed those found in Table 4. The following categories were identified through thematic analysis: 1) Geography, 2) Names, 3) Culinary, 4) Linguistic, 5) Activities, 6) Items, 7) Media, 8) Measurements, and 9) Currency. Geography pertains to the use of countries, capitals, and city names. Names refers to the use of Latine names (e.g., María, Jose, etc.). Culinary refers to the use of foods. Linguistic pertains to the use or reference of Latine languages. Activities refers to activities shared across Latine cultures (e.g., salsa dancing, soccer playing). Items refers to iconography that relate to Latine cultures (e.g., flags). Media pertains to the use of music and movies. Measurements capture the systems used in Latin America (e.g., Celsius, kilometers, etc.). Currency refers to the incorporation of Latine currencies (e.g., pesos, real, quetzal, etc.).

Listing 2: Deeply Culturally-Relevant Example

```
countries_capitals = {"Mexico": "Mexico City", "Brazil":
    "Brasília", "Argentina": "Buenos Aires"}
if "Brazil" in countries_capitals:
    print("Brazil is in the dictionary!")
else:
    print("Brazil is not in the dictionary!")
```

The most prominent form of cultural relevance was geography (47%) – particularly for deep cultural relevance (53%). For instance, in Listing 2 we see an example of key membership in a dictionary. The dictionary is explicitly highlighting Latin American countries and their capitals, providing a Latine audience with a cultural anchor,

but also provides a non-Latine audience with an opportunity to learn about Latin American culture.

5 DISCUSSION

5.1 RQ1: Textbook Evaluation

We see that the LLM-generated culturally-relevant textbook is mostly sensible, linguistically accessible, at an appropriate level for code examples, readable, and correct (Section 4.1). These results fall in line with the work finding that LLM-generated learning resources are not statistically significantly different from those that are human-generated [7]. These results bode well for the future of learning resource generation, as a purely human-generated textbook could take several years to write [32, 37]. The code appropriateness score could stem from LLMs being trained on online repositories which may feature more sophisticated code – which could also explain the correctness evaluation. The linguistic approachability assessment may be due to the formal writing on code documentation. The readability score may be due to LLMs generally not having writing errors – and are used for correcting such mistakes [6]. The sensibility score may be due to the LLM generating by chapter and not remembering the context beforehand.

Two-thirds of the cultural relevance implemented was surface-level – in line with research highlighting ChatGPT's cultural alignment with the United States over other cultures [2, 5]. One of the main implementations was by asking the reader to incorporate Latine culture (e.g., *Create a variable called "country" and assign it the string value of the name of a Latin American country. Print the value of the variable to the console.*). This method is particularly noteworthy as we explicitly asked the LLM in the prompt to not have the reader incorporate Latine culture (see Listing 1). We also found a statistically significant difference regarding cultural relevance when comparing the examples and exercises from the LLM-generated textbook. This may be due to examples being restricted to shorthand explanation of techniques, while exercises can use these techniques for a larger narrative.

5.2 RQ2: Incorporation of Latine Culture

In terms of the surface-level incorporation of cultural relevance, there seemed to be a "slap-on" effect that the LLM implemented. By this, we mean that examples and exercises would superficially add a cultural reference to a problem – similar to Section 5.1. Listing 3 and the following exercise are some ways that the LLM would "slap-on" Latine culture to a problem: *Try applying a red filter to an image of a traditional Latin American dish. Save the modified image with a new name and display it.*

Listing 3: Surface-Level Cultural Example

```
name = "Carlos"
print("Hello, " + name + "!") # Output: Hello, Carlos!
```

The main categories of cultural relevance can be found in Table 5. We see that "Geography" was the most incorporated category across surface-level and deep-level implementations. This is likely due to how GPT-3.5-Turbo is trained mostly on English data and might rely heavily on geographical markers (e.g., countries, capitals, cities) as cultural references to Latin America. A noteworthy aspect of the geographical implementation is little variability it implemented. The same five countries were used across the textbook (Argentina, Brazil, Chile, Mexico, and Peru) even though there are over 20 countries that could be represented. There were no Caribbean (e.g., Puerto Rico, Dominican Republic) or Central American (e.g., Nicaragua, Panama) countries.

A similar phenomenon can be seen with the "Names" category. The only names used in the textbook were Maria, Carlos, Juan, Pedro, and Luisa. Not only was there little name variability, but there was no representation of more commonly used names of indigenous groups in Latin America (e.g., Cuauhtémoc, Tenoch, Xochitl, Citlalli). Future work may want to implement a fine-tuning process incorporating Latine culture, similar to the work of Yao et al. with Chinese-English parallel data [44].

6 LIMITATIONS

A limitation to our study would be that GPT-3.5-Turbo was last updated September 2021, three years ago. It is possible that using the new GPT-4.0 model would yield more positive results, but it is not freely accessible. The free version was chosen to ensure the general public can create their own culturally-relevant examples. Note that the use of LLMs means that no two outputs may be identical – using the same prompts from this study may yield similar, but different results. Another limitation would be that two researchers reviewed the culturally-relevant textbook, both from the same 4-year research institution. There is a possibility that with a wider range of perspectives and more reviewers, the evaluation results would vary. Lastly, ChatGPT is significantly culturally aligned with the United States over others [5]. A fine-tuning process incorporating Latine culture may provide deeper cultural engagement.

7 CONCLUSION

In the U.S., culturally-relevant computing is a popular pedagogical approach for Latine students. However, creating culturally relevant learning resources can be time-consuming. To expedite the process, we utilized LLMs through prompt engineering to generate a culturally-relevant textbook for a CS1 course. The evaluation showed the generated textbook was mainly sensible, correct, readable, linguistically approachable, used code examples at an appropriate level, and was culturally-relevant. The depth of cultural engagement was typically surface-level, where exercises were more deeply engaging than coding examples. While LLMs can be a valuable initial step for writing culturally-relevant learning resources, further refinement is needed. This work may extend to generating culturally-relevant textbooks for other cultures in the future.

ACKNOWLEDGMENTS

This work was supported in part by the UCSD Sloan Scholars Fellowship and Gates Millennium Scholarship. Thanks to Jackelyne García-Villegas for their feedback on this paper.

REFERENCES

[1] Monika Akbar, Lucia Dura, Ann Q. Gates, Angel Ortega, Mary K. Roy, Claudia Santiago, Jesus G. Tellez, and Elsa Villa. 2019. Sol y Agua: A Game-based Learning Platform to Engage Middle-school Students in STEM. *Proc. - Frontiers in Education Conference, FIE* 2018-October (3 2019). https://doi.org/10.1109/FIE.2018.8659071

[2] Arnav Arora, Lucie-Aimée Kaffee, and Isabelle Augenstein. 2022. Probing pre-trained language models for cross-cultural differences in values. *arXiv preprint arXiv:2203.13722* (2022).

[3] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[4] US Census Bureau. 2021. 2020 Census Statistics Highlight Local Population Changes and Nation's Racial and Ethnic Diversity. *Census. gov* (2021).

[5] Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study. *arXiv preprint arXiv:2303.17466* (2023).

[6] William Castillo-González, Carlos Oscar Lepez, and Mabel Cecilia Bonardi. 2022. Chat GPT: a promising tool for academic editing. *Data and Metadata* 1 (2022), 23–23.

[7] Paul Denny, Hassan Khosravi, Arto Hellas, Juho Leinonen, and Sami Sarsa. 2023. Can We Trust AI-Generated Educational Content? Comparative Analysis of Human and AI-Generated Learning Resources. *arXiv preprint arXiv:2306.10509* (2023).

[8] Ramon Dijkstra, Zülküf Genç, Subhradeep Kayal, Jaap Kamps, et al. 2022. Reading Comprehension Quiz Generation using Generative Pre-trained Transformers.

[9] Maureen Doyle, Kevin G. Kirby, and Gary Newell. 2008. Engaging constructions: Family-based computing experiences for immigrant middle school students. *SIGCSE'08 - Proc. of the 39th ACM Technical Symp. on Computer Science Education* (2008), 58–62. https://doi.org/10.1145/1352135.1352158

[10] NACE Employers. 2015. Salary Survey. (2015).

[11] Sheena Erete, Karla Thomas, Denise Nacu, Jessa Dickinson, Naomi Thompson, and Nichole Pinkard. 2021. Applying a Transformative Justice Approach to Encourage the Participation of Black and Latina Girls in Computing. *ACM Transactions on Computing Education* 21 (12 2021). Issue 4. https://doi.org/10.1145/3451345

[12] Diana Franklin, Phillip Conrad, Gerardo Aldana, and Sarah Hough. 2011. Animal tlatoque: Attracting middle school students to computing through culturally-relevant themes. *SIGCSE'11 - Proc. of the 42nd ACM Technical Symp. on Computer Science Education* (2011), 453–458. https://doi.org/10.1145/1953163.1953295

[13] Ebrahim Gabajiwala, Priyav Mehta, Ritik Singh, and Reeta Koshy. 2022. Quiz maker: Automatic quiz generation from text using NLP. In *Futuristic Trends in Networks and Computing Technologies: Select Proc. of Fourth Int'l. Conf. on FTNCT 2021*. Springer, 523–533.

[14] Leiny Y. Garcia, Miranda C. Parker, Santiago Ojeda-Ramirez, and Mark Warschauer. 2023. Confidence is the Key: Unlocking Predictive Factors of Latinx Elementary Students on a Computational Thinking Measure. *SIGCSE 2023 - Proc. of the 54th ACM Technical Symp. on Computer Science Education* 1 (3 2023), 326–332. Issue 7. https://doi.org/10.1145/3545945.3569856

[15] https://bit.ly/Culturally-Relevant-Introductory-CS-Textbook-with-Latin-American-Culture [n.d.]. https://bit.ly/Culturally-Relevant-Introductory-CS-Textbook-with-Latin-American-Culture. Accessed: 2024-01-19.

[16] https://platform.openai.com/docs/guides/prompt-engineering [n.d.]. https://platform.openai.com/docs/guides/prompt-engineering. Accessed: 2023-11-21.

[17] Sihua Hu, Kaitlin T Torphy, and Amanda Opperman. 2019. Culturally relevant curriculum materials in the age of social media and curation. *Teachers College Record* 121, 14 (2019), 1–22.

[18] Hazem Ibrahim, Fengyuan Liu, Rohail Asim, Balaraju Battu, Sidahmed Benabderrahmane, Bashar Alhafni, Wifag Adnan, Tuka Alhanai, Bedoor AlShebli, Riyadh Baghdadi, et al. 2023. Perception, performance, and detectability of conversational artificial intelligence across 32 university courses. *Scientific Reports* 13, 1 (2023), 12187.

[19] Jussi S Jauhiainen and Agustín Garagorry Guerra. 2023. Generative AI and ChatGPT in School Children's Education: Evidence from a School Lesson. *Sustainability* 15, 18 (2023), 14025.

[20] Mollie Jordan, Kevin Ly, and Adalbert Gerald Soosai Raj. 2024. Need a Programming Exercise Generated in Your Native Language? ChatGPT's Got Your Back: Automatic Generation of Non-English Programming Exercises Using OpenAI GPT-3.5. In *Proc. of the 55th ACM Technical Symp. on Computer Science Education V. 1*. 618–624.

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

[22] Marta M Koć-Januchta, Konrad J Schönborn, Lena AE Tibell, Vinay K Chaudhri, and H Craig Heller. 2020. Engaging with biology by asking questions: Investigating students' interaction and learning with an artificial intelligence-enriched textbook. *Journal of Educational Computing Research* 58, 6 (2020), 1190–1224.

[23] Jens Manuel Krogstad, Jeffrey S. Passel, Mohamad Moslimani, and Luis Noe-Bustamante. 2022. Key facts about U.S. Latinos for National Hispanic Heritage Month. *Pew Research Center* (2022). https://www.pewresearch.org/short-reads/2023/09/22/key-facts-about-us-latinos-for-national-hispanic-heritage-month/sr_23-09-22_hispanic-key-facts_13/

[24] Gloria Ladson-Billings. 1995. But that's just good teaching!: The case for culturally relevant pedagogy. *Theory into Practice* 34, 3 (1995), 159–165.

[25] Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 2023. Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in english education. *Education and Information Technologies* (2023), 1–33.

[26] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing code explanations created by students and large language models. *arXiv preprint arXiv:2304.03938* (2023).

[27] Amari N. Lewis, Joe Gibbs Politz, Kristen Vaccaro, and Mia Minnes. 2022. Learning about the Experiences of Chicano/Latino Students in a Large Undergraduate CS Program. *Annual Conf. on Innovation and Technology in Computer Science Education, ITiCSE* 1 (7 2022), 165–171. https://doi.org/10.1145/3502718.3524780

[28] Stephanie Lunn, Leila Zahedi, Monique Ross, and Matthew Ohland. 2021. Exploration of Intersectionality and Computer Science Demographics: Understanding the Historical Context of Shifts in Participation. *ACM Transactions on Computing Education* 21 (6 2021), 10. Issue 2. https://doi.org/10.1145/3445985

[29] Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, and Ziheng Huang. 2022. Generating diverse code explanations using the gpt-3 large language model. In *Proc. of the 2022 ACM Conf. on Int'l. Computing Education Research-Vol. 2*. 37–39.

[30] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.

[31] US Department of Labor, Bureau of Statistics Staff, and United States. Bureau of Labor Statistics. 2000. *Occupational Outlook Handbook*. Bernan Press (PA).

[32] Paul W Robinson and Kenneth L Higbee. 1978. Publishing a textbook: Advice from authors and publishers. *Teaching of Psychology* 5, 4 (1978), 175–181.

[33] Jean J. Ryoo, Alicia Morris, and Jane Margolis. 2021. "What Happens to the Raspado man in a Cash-free Society?": Teaching and Learning Socially Responsible Computing. *ACM Transactions on Computing Education* 21 (12 2021). Issue 4. https://doi.org/10.1145/3453653

[34] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proc. of the 2022 ACM Conf. on Int'l. Computing Education Research-Vol. 1*. 27–43.

[35] US BUREAU OF LABOR STATISTICS. 2019. *Labor Force Characteristics by Race and Ethnicity, 2018*. Technical Report. Washington, DC: US Bureau of Labor Statistics, 2019: 1-60.

[36] Chris Stephenson, Alison Derbenwick Miller, Christine Alvarado, Lecia Barker, Valerie Barr, Tracy Camp, Carol Frieze, Colleen Lewis, Erin Cannon Mindell, Lee Limbird, et al. 2018. *Retention in Computer Science Undergraduate Programs in the US: Data Challenges and Promising Interventions*. ACM.

[37] Robert J Sternberg. 2017. Why It Is so Hard for Academics to Write Textbooks. *Psychology Teaching Review* 23, 1 (2017), 79–84.

[38] Stephanie Tena-Meza, Miroslav Suzara, and Aj Alvero. 2022. Coding with Purpose: Learning AI in Rural California. *ACM Transactions on Computing Education* 22 (9 2022), 1–18. Issue 3. https://doi.org/10.1145/3513137

[39] James Vanderhyde and Florence Appel. 2016. With Greater CS Enrollments Comes an Even Greater Need for Engaging Teaching Practices. *Journal of Computing Sciences in Colleges* 32, 1 (2016), 38–45.

[40] Ismael Villegas Molina, Audria Montalvo, and Adalbert Gerald Soosai Raj. 2024. US Latines in Computing: A Review of the Literature. (2024), 1381–1387.

[41] Ismael Villegas Molina, Adrian Salguero, Shera Zhong, and Adalbert Gerald Soosai Raj. 2023. The Effects of Spanish-English Bilingual Instruction in a CS0 Course for High School Students. *Proc. of the 2023 Conf. on Innovation and Technology in Computer Science Education V. 1* 7 (2023). https://doi.org/10.1145/3587102

[42] Sara Vogel, Laura Ascenzi-Moreno, Christopher Hoadley, and Kate Menken. 2019. The role of translanguaging in computational literacies documenting middle school bilinguals' practices in computer science integrated units. *SIGCSE 2019 - Proc. of the 50th ACM Technical Symp. on Computer Science Education* (2 2019), 1164–1170. https://doi.org/10.1145/3287324.3287368

[43] Zichao Wang, Jakob Valdez, Debshila Basu Mallick, and Richard G Baraniuk. 2022. Towards human-like educational question generation with large language models. In *Int'l. Conf. on artificial intelligence in education*. Springer, 153–166.

[44] Binwei Yao, Ming Jiang, Diyi Yang, and Junjie Hu. 2023. Empowering LLM-based Machine Translation with Cultural Awareness. *arXiv preprint arXiv:2305.14328* (2023).

[45] Margaret Yau. 2013. ENGAGING HISPANIC/LATINO(A) YOUTH IN COMPUTER SCIENCE: AN OUTREACH PROJECT EXPERIENCE REPORT *. *Journal of Computing Sciences in Colleges* (2013). https://doi.org/10.5555/2458539.2458559

[46] Leila Zahedi, Hossein Ebrahiminejad, Monique S Ross, Matthew W Ohland, and Stephanie J Lunn. 2021. Multi-Institution Study of Student Demographics and Stickiness of Computing Majors in the USA. In *2021 CoNECD*.

[47] Stuart Zweben and Betsy Bizot. 2023. 2022 CRA Taulbee Survey. *Computing Research News* (2023).