Examining LLM Prompting Strategies for Automatic Evaluation of Learner-Created Computational Artifacts

Xiaoyi Tian
University of Florida
tianx@ufl.edu

Amogh Mannekote
University of Florida
amogh.mannekote@ufl.edu

Carly E. Solomon
University of Florida
csolomon1@ufl.edu

Yukyeong Song
University of Florida
y.song1@ufl.edu

Christine Fry Wise
The Findings Group
christine@thefindingsgroup.org

Tom Mcklin
The Findings Group
tom@thefindingsgroup.org

Joanne Barrett
University of Florida
jrbarrett@ufl.edu

Kristy Elizabeth Boyer
University of Florida
keboyer@ufl.edu

Maya Israel
University of Florida
misrael@ufl.edu

ABSTRACT

Recent advancements in automatic evaluation have made significant progress, yet evaluating learner-created computational artifacts such as project-based code remains challenging. This study investigates the capability of GPT-4, a state-of-the-art Large Language Model (LLM), in assessing learner-created computational artifacts. Specifically, we analyze the source code of 75 chatbots predominantly built by middle school learners. We compare four LLM prompting strategies ranging from example-based to rubric-informed approaches. The experimental results indicate that the LLM-based evaluation module achieves substantial agreement (Cohen's weighted $\kappa = 0.797$) with human evaluators in two of five artifact dimensions, moderate agreement in one, and fair agreement in the remaining two dimensions. We analyze the trade-offs between different LLM prompting strategies through qualitative error analysis. The findings demonstrate the potential of LLMs for automatically evaluating project-based, open-ended computational artifacts.

Keywords

automated assessment, large language models, GPT-4, artifact evaluation, AI education

1. INTRODUCTION AND RELATED WORK

Project-based learning is prevalent in STEM education and, more recently, in artificial intelligence (AI) education [25, 14]. This approach engages learners in an open-ended process of designing and developing applications [9, 11]. Project-based learning has demonstrated many advantages, such as increased engagement [14] and a deeper understanding of complex concepts [13, 10]. The ability to design and construct novel artifacts represents the pinnacle of cognitive achievement [2]. These learner-created projects are often crucial for evaluating their learning progress, serving as key tools for teachers to assess and offer constructive feedback [35].

A major challenge in project-based learning is evaluating learner projects and providing timely feedback. Traditional assessments often require domain experts to manually score the project against a rubric [10, 20], which can be time-consuming and resource-intensive. There is a growing interest in automating this process to enhance scalability. However, existing automatic computational artifact evaluation methods have several limitations. For instance, some rely on testing the outcome of student programs with predefined test cases or unit tests [32], which offer limited feedback to learners. Others compare the structural similarity of student code to expert solutions [24], but this approach is limited to sufficiently simple tasks with a set of solutions that can be tractably defined. While recent data-driven approaches aim to assess more complex projects [27, 4, 34, 28], these approaches usually require large datasets to train the model, which can be challenging to obtain in educational settings that are typically resource-constrained.

The rapid development of large language models (LLMs) and their use in education suggest promising directions for computational artifact evaluation. LLMs have performed well across many disciplines and tasks, such as grading short answers [6, 36] and evaluating essays [19, 21]. Yet, the potential of LLMs in assessing more technical and creative aspects of computational artifacts remains unexplored. This presents a notable research gap, particularly in understanding the LLMs' performance on analyzing learner-created artifacts and the trade-offs between different prompting strategies to improve automated assessment methods.

This study investigates the capability of GPT-4, a state-of-the-art LLM at the time of this research, in assessing learner-created computational artifacts. Specifically, we analyze the source code of 75 chatbot programs created mainly by middle school learners. Engaging young learners in building chatbots can foster AI learning and enhance attitudes in computing-related careers [30]. However, evaluating these learner-created chatbots presents significant challenges as it requires an understanding of both design and technical implementation of the chatbot and the logical flow of conversations.

A key aspect in chatbot development is the concept of *intent*, which represents the purpose or the goal behind a chatbot user's message, such as "seek music recommendations." Our analysis focuses on five artifact dimensions critical to chatbot evaluation: 1) *Greet Intent*, triggered at the conversation's onset with a greeting such as "Hi"; 2) *Default Fallback Intent*, triggered when the chatbot cannot confidently match a user input to known intents, responding with "Sorry, I didn't get that. Can you say that again?"); 3) *Follow-up Intents*, intents that maintain context from previous intent to facilitate multi-turn conversation; 4) *Training Phrases*, list of example user utterances that train the AI chatbot to recognize intents; and 5) *Responses*, the specific replies given by the chatbot once an intent is recognized. For example, for the intent "seeking music recommendations," *training phrases* could include "Can you recommend me a song?" or "What music do you like?" An example *response* could be "Sure, here is a classic one: Fireworks by Taylor Swift."

To automatically assess the learner-created chatbots, our study examines four LLM prompting strategies: zero-shot without a rubric, zero-shot with a rubric, few-shot without a rubric, and few-shot with a rubric[1]. We investigate the following research questions (RQs):

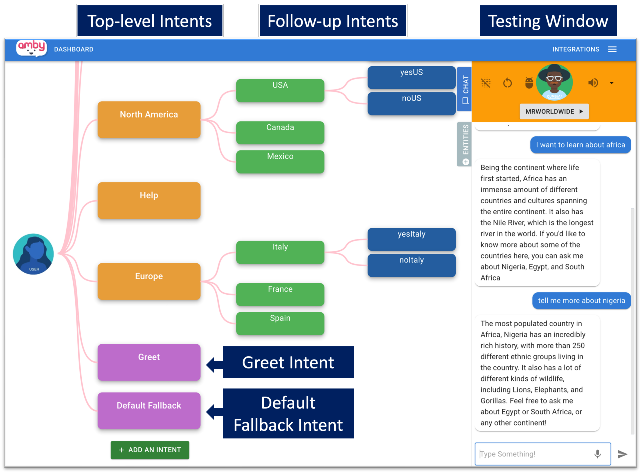
- RQ1: How do LLMs perform in assessing different aspects of computational artifacts created by learners?
- RQ2: What are the trade-offs among different prompting strategies for LLMs to automatically evaluate learner-created computational artifacts?

This paper makes the following contributions:

1. An examination of GPT-4's performance on evaluating learner-created computational artifacts, highlighting its effectiveness in evaluating isolated artifact components.
2. An analysis of the trade-offs between rubric-based and example-based prompting strategies, showing that few-shot learning with contextual examples improves LLMs' grading accuracy.
3. An exploration of LLMs' limitations when grading complex, interconnected components in computational artifacts.

2. STUDY CONTEXT

This study analyzes the chatbot artifacts created in a learning environment called AMBY ("AI Made By You"). AMBY (Figure 1) is a graphical interface designed for middle school-aged learners to create conversational agents without programming [31]. AMBY was utilized in three middle school AI summer camps across two years in the southeast United States, focusing on computer science and AI learning [30, 12]. During the summer camp, learners designed the chatbots to interact with target users and solve problems, such as introducing animal facts and offering movie recommendations.



I want to learn about africa

Being the continent where life first started, Africa has an immense amount of different countries and cultures spanning the entire continent. It also has the Nile River, which is the longest river in the world. If you'd like to know more about some of the countries here, you can ask me about Nigeria, Egypt, and South Africa

tell me more about nigeria

The most populated country in Africa, Nigeria has an incredibly rich history, with more than 250 different ethnic groups living in the country. It also has a lot of different kinds of wildlife, including Lions, Elephants, and Gorillas. Feel free to ask me about Egypt or South Africa, or any other continent!

Type Something!

Figure 1: AMBY development environment for learners to create chatbots.

Data. The dataset contains 75 chatbots, with 66 created by middle school learners (average age 12.7) during summer camp and 9 by undergraduate learners in a pre-camp workshop. It comprises the project source files of these learner-created chatbots, each a text file containing metadata such as intents, training phrases, and responses in structured natural language format.

Table 1: Chatbot artifact evaluation rubric. Each dimension was rated on a 1-4 scale.

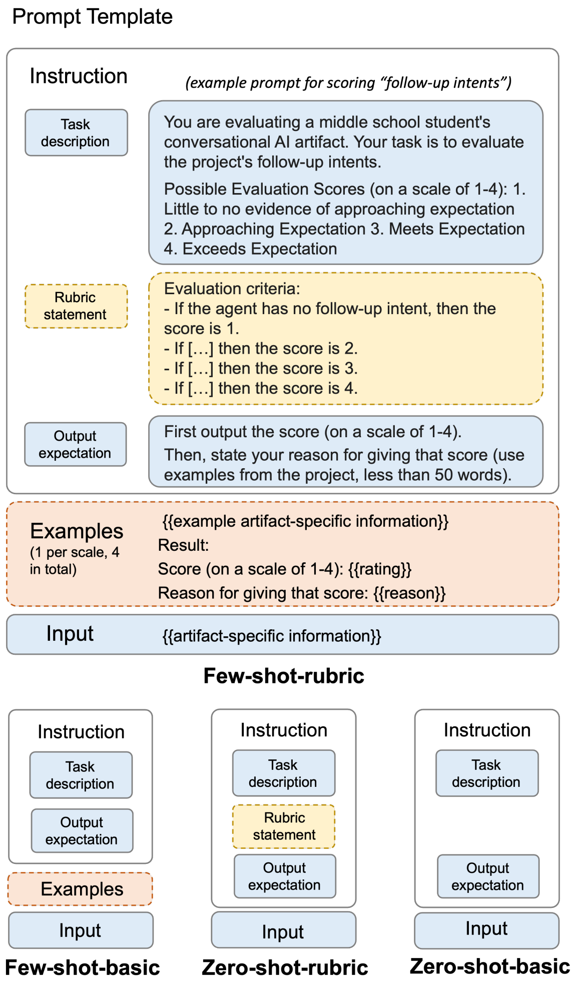
Artifact Dimensions	**Statement for Score of 3 (Meets Expectations)**
Greet intent	At least one customized greet response demonstrating its purpose. May not set exact user expectations.
Default fallback intent	At least one customized default fallback response that can redirect the users.
Follow-up intents	Multiple logical follow-up intents. Each follow-up intent is related to its parent intent mostly logically and can be triggered properly based on the responses from their parent intents.
Training phrases	Most training phrases are ample, cohesive, and varied within the intent.
Responses	At least one response is of appropriate length, logical, conversational, and mostly free from grammatical errors.

Rubric Development. To ensure systematic evaluation on learner's chatbots, we developed a rubric based on the camp's

---

[1] *Zero-shot learning* involves the model performs tasks without prior examples. *Few-shot learning* provides the model with a few examples to guide its outputs.

AI learning objectives and existing dialogue system evaluation frameworks [33]. The original rubric includes ten dimensions with a grading scale of 1-4. The Quadratic Weighted Cohen's Kappa between two human graders across all rubric dimensions is 0.82, indicating substantial inter-rater reliability [18]. After resolving any discrepancies through discussion, one grader proceeded to grade the remaining artifacts. In this paper, we focus on five dimensions pertinent to dialogue system architecture. The evaluation criteria for these five dimensions are in Table 1, with full scoring guidelines in Appendix B.



You are evaluating a middle school student's conversational AI artifact. Your task is to evaluate the project's follow-up intents.

Possible Evaluation Scores (on a scale of 1-4): 1. Little to no evidence of approaching expectation 2. Approaching Expectation 3. Meets Expectation 4. Exceeds Expectation

Evaluation criteria:
- If the agent has no follow-up intent, then the score is 1.
- If [...] then the score is 2.
- If [...] then the score is 3.
- If [...] then the score is 4.

First output the score (on a scale of 1-4).
Then, state your reason for giving that score (use examples from the project, less than 50 words).

{{example artifact-specific information}}
Result:
Score (on a scale of 1-4): {{rating}}
Reason for giving that score: {{reason}}

{{artifact-specific information}}

Figure 2: Prompt templates for four experiment conditions (zero-shot-basic, zero-shot-rubric, few-shot-basic, few-shot-rubric). Each prompt template might consist of Instruction, Examples, and an unseen student artifact as Input. The format of {{artifact-specific information}} is dependent on the dimension being evaluated.

3. LLM-BASED ARTIFACT ASSESSMENT IMPLEMENTATION

In this section, we introduce the implementation procedure of our LLM-based artifact assessment technique. First, we defined our prompt template and prompting strategies. Then, we preprocessed the dataset to extract features of the chatbot (e.g., intents, training phrases, responses). We used an open-source framework, LLM4Qual [17], to manage the elements of our prompt templates and conduct experiments. We iteratively developed prompt templates and report the evaluation results on the finalized prompt templates across each rubric dimension.

3.1 Prompt Components and Strategies

The goal for this work is to explore the influence of the prompting strategies on evaluation accuracy; thus we first define our prompt template and tailor the components of the prompt template for each prompting strategy.

Prompt Components. Our prompt template has three components: **Instruction**, **Examples** (optional), and **Input** (Figure 2). **Instruction** sets out the evaluation task and contains three possible subsections. The Instruction sets the evaluation task with a *task description*, *rubric statement* specifying criteria for the four-point scale, and *output expectations*. **Examples** are only included for few-shot conditions, which are manually chosen exemplars corresponding to each four-point scale of the rubric dimension. Each example includes the relevant component(s) from the learner-created artifact, human-grader's score, and their accompanying rationale text. The final component of the template is the **Input**, which specifies the artifact component related to the specific rubric dimension. For example, the **Input** for the *greet intent* dimension would be the chatbot's greet intent responses, and for the *training phrases* dimension, it would be the list of all intents' training phrases and their respective counts. By limiting the **Input** to only relevant parts of the artifact to the LLM, we ensure that the model is not "distracted" by irrelevant information.

Prompting Strategies. Based on the complexity of prompt **Instruction** and inclusion of **Examples**, we divide the prompt template into four conditions: zero-shot basic (neither rubric nor examples included), zero-shot-rubric (rubric included but not examples), few-shot-basic (examples included but not rubric), and few-shot-rubric (both rubric and examples are included). A comparison between different prompting strategies is shown in Figure 2.

3.2 Experimental Setup

Implementation. We utilized LLM4Qual [17], an open-source framework that simplifies development of LLM-based proxy annotators, to run our experiments. This framework employs Langchain[2] in the backend and allows the specification of rubric-wise prompt templates as well as few-shot examples through YAML files.

Input Feature Extraction. We developed Python scripts to extract "input features" (artifact components) from the learner-created chatbot artifacts. The input features contained the chatbot name, top-level intents, follow-up intents, follow-up intents count, training phrases, responses, training phrases count, responses count, intent tree (a tree structure of intent names), greet intent responses, and default fallback intent responses. These features were selected to comprise the **Input** in the prompt template. Our motivation behind selecting only a subset of features for the prompt was to minimize the number of input tokens, which, in turn, would minimize

---

[2] https://www.langchain.com/

Table 2: **Evaluation metrics of GPT4-generated scores with four prompting strategies and human-human agreement for five artifact dimensions: greet intent, default fallback intent, follow-up intents, training phrases and responses. Evaluation matrics include Spearman Correlation (ρ) and Quadratic Weighted Cohen's Kappa (QWK). Bolded numbers indicate the best performance for a metric.**

Artifact Dimensions	Metrics	Human-human	Human-GPT4			
			Zero-shot Basic	Zero-shot Rubric	Few-shot Basic	Few-shot Rubric
Greet intent	ρ	0.850	0.339	0.641	**0.659**	0.646
	QWK	0.820	0.325	0.623	**0.698**	0.645
Default Fallback intent	ρ	0.979	0.179	0.782	0.779	**0.816**
	QWK	0.984	0.252	0.750	0.781	**0.797**
Follow-up intents	ρ	0.839	0.133	0.217	0.203	**0.346**
	QWK	0.805	0.154	0.244	0.230	**0.388**
Training Phrases	ρ	0.819	0.231	0.406	0.464	**0.551**
	QWK	0.808	0.168	0.325	0.409	**0.479**
Responses	ρ	0.750	0.150	0.127	**0.235**	0.143
	QWK	0.715	0.083	0.105	**0.158**	0.094

the cost.

Data Splits. We split our dataset into three sets: training, validation, and test. However, unlike in a traditional machine learning setup, our training set contained only a hand-curated set of examples that were used for few-shot prompting (specifically, four examples for each dimension).

Prompt Engineering. During the process of prompt engineering, we iteratively refined our prompt templates over the validation set until we began noticing diminishing returns to the performance from further instruction modifications. We conducted our model runs over the test set only after finalizing our prompt template and then reported our final results.

Evaluation Metrics. To evaluate the LLM-generated scores, we used Spearman correlation (ρ) and Weighted Cohen's Kappa (QWK) to compare with human grading. Both are common agreement measurements for ordinal ratings between two grading parties [19, 21].

4. RESULTS

RQ1: How Do LLMs Perform in Assessing different Aspects of Computational Artifacts? Table 2 presents the level of agreement between human evaluations and GPT-4 across each of our four prompting strategies. Notably, GPT-4 demonstrates high alignment with human assessments particularly for the *greet intent* and *default fallback intent* dimensions, with Quadratic Weighted Cohen's Kappa scores of 0.698 and 0.797 respectively, signaling substantial agreement [18]. The Kappa score of 0.479 for the *training phrases* reflects moderate agreement, while the highest agreement score (among all four prompting strategies) for the *follow-up intents* and *responses* dimensions are 0.388 and 0.158 respectively, indicating only slight to fair agreement [18].

RQ2: What Are the Tradeoffs among Different Prompting Strategies? Examining the impact of different prompting strategies reveals that the *few-shot-rubric* prompting strategy outperforms the other strategies in evaluating *default fallback intent*, *follow-up intents*, and *training phrases*. For the other rubric dimensions, *greet intent* and *responses*, the *few-shot-basic* (i.e., without rubric statement) setup is most effective. The two zero-shot (i.e., without examples) conditions, regardless of their rubric inclusion, do not perform as well as the few-shot conditions. Particularly, the *zero-shot-basic* approach consistently underperforms relative to other strategies.

5. DISCUSSION

In this section, we explain why LLM-generated scores deviate from human scores and why certain prompting strategies outperform others through qualitative error analysis. We contrast human expert rationales with those generated by the LLM. This discussion will shed light on the strengths and weaknesses of different prompting strategies and refine LLM-based evaluation methods.

Our findings suggest that LLM evaluation performance is task-dependent. We show that LLMs performs well in evaluating the *greet intent* and *default fallback intent*. These dimensions deal with relatively isolated elements of a singular intent in an artifact. For these dimensions, including 1) a clearly articulated rubric statement, and 2) a few illustrative examples results in high agreement with human expert annotators. This finding is promising for real-world applications such as using LMMs to provide immediate and precise feedback to learners.

Notwithstanding this optimistic finding, the evaluation performance reduces when evaluating *follow-up intents*, *training phrases* and *responses* dimensions (as illustrated in Table 2). This variability aligns with findings from other open-ended text evaluations [8, 1, 36], where accuracy varied across topics and types. For example, Zhang et al. [36] reported that accuracy in grading mathematical short answers varied across different question topics and types, with Kappa scores ranging from 0.4 to 0.758. We attribute the low performance in our study to the LLM's requirements to 1) carry out complex reasoning across multiple intents, and 2) infer the logical progression of the conversation. These requirements present challenges even for human evaluators to reach consensus (for example, the Kappa score for *responses* dimension between human-graders is only 0.715, lower than other dimensions). We present two additional examples in the Appendix A comparing the grades and rationals between human and LLM evaluations.

For our second research question, our results confirm the effectiveness of contextual examples in the prompt templates to improve LLM grading accuracy, which supports prior literature on few-shot learning [3, 26]. We observed that few-shot prompting strategies generally surpass zero-shot strategies, regardless of whether the prompt template contains the rubric statement or not. This finding is promising for the use of LLMs for the automated evaluation of computational artifacts as it shows that even a few examples can improve grading accuracy.

Table 3: Evaluation of the Greet Intent Response of a Chatbot: "*Hey, bro! My name is M&P game reccs, and you can ask me to start the quiz for my cracked game quiz to give you a broad game rec, man!*" Colored texts highlight the main discrepancy between human and LLM-generated rationales.

Evaluation	Score	Rationale
Human	4	**It sets user expectations properly because the main focus of the chatbot is the quiz which gives the user game recommendations.**
GPT-4: zero-shot-basic	2	The greeting does introduce the bot and its function, but **the language used is informal** and may not be understood by all users. **The message could be clearer and more professional.**
GPT-4: zero-shot-rubric	3	The greet intent is customized and introduces the chatbot as M&P game reccs. It also demonstrates its purpose by mentioning that it can start a quiz for game recommendations. However, **it doesn't set exact user expectations or clearly direct the user for next steps.**
GPT-4: few-shot-basic	3	The greeting is casual and friendly, but **it could be clearer in explaining the bot's purpose and how it can assist the user.**
GPT-4: few-shot-rubric	4	The greeting is customized, introduces the bot, demonstrates its purpose, and sets exact user expectations.

Additionally, our zero-shot prompting experiments highlight that even without examples for few-shot prompting, a well-designed rubric statement can greatly enhance grading accuracy (see Table 2). This finding addresses the "cold-start" problem where instructors initially lack clear exemplars for each evaluation scale point [23].

We conduct a comparative analysis of the four prompting strategies in their generated scores and rationales. Consider the example of a "greet intent" (Table 3), where the objective is for learners to demonstrate the chatbot's purpose and establish proper user expectations. The human evaluator grades this dimension as 4, noting the chatbot's precise setting of expectations for a game recommendation quiz. Among the four LLM prompting strategies tested, only the few-shot-with-rubric approach matches the human evaluator's score. The two strategies without a rubric penalize the chatbot for its informal language. However, this informal style, including slang and jargon common among the learners, reflects the learners' personalities, making it suitable for the greet intent where personal relevance is key. The strategies that include rubric criteria tend to produce responses more consistent with the evaluation criteria. However, the rationale provided under the few-shot-rubric strategy in this example merely echoes the rubric statement without discussing specifics about the artifact. This raises concerns about the depth and relevance of feedback LLMs can offer to learners based on their specific artifact. Studies suggest that alternative prompting instructions (e.g., Chain-of-Thought prompting [37]) can enhance the performance of LLMs in complex tasks involving deeper reasoning [5].

6. CONCLUSION AND FUTURE WORK

Our study leverages GPT-4 to automatically evaluate computational artifacts from AI education and provide feedback. Examining different prompting strategies, we uncover the potential and challenges of using LLMs for assessing learner-created artifacts. These findings highlight the importance of careful prompting for effective LLM utilization. This research paves the way for integrating LLMs into educational assessment, particularly in complex CS and AI pedagogy.

More broadly, there remain several critical roadblocks in the way of complete adoption of LLMs for automated evaluation of complex artifacts. These include the brittleness of natural language prompts [16], issues of prompt calibration pertaining to the order of few-shot examples [38], and lack of guarantees about LLM performance on individual samples beyond macro-level indicators. Robust solutions to such fundamental issues are needed to address stakeholder skepticism of automated grading methods.

Our research suggests exploring additional prompting strategies like Chain-of-Thought prompting [37] to improve grading performance. Second, although we evaluate the LLM's performance through agreement with expert human graders, we do not yet assess learner's responses on the LLM-generated scores and rationales. Understanding how learners perceive and trust the LLM's feedback is crucial, as it informs what kind of hints, feedback, or interactions might be most beneficial [29, 15]. For dimensions where LLM grading is less effective, future work can explore human-in-the-loop approaches to maintain reliability [22, 7].

Acknowledgements

This research was supported by the National Science Foundation through grant DRL-2048480.

References

[1] M. Ariely, T. Nazaretsky, and G. Alexandron. Machine learning and hebrew nlp for automated assessment of open-ended questions in biology. *International journal of artificial intelligence in education*, 33(1):1–34, 2023.

[2] P. Armstrong. Bloom's taxonomy. vanderbilt university center for teaching. retrieved from https://cft. vanderbilt. edu/guides-sub-pages/blooms-taxonomy, 2010.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Y. Dong, S. Marwan, P. Shabrina, T. Price, and T. Barnes. Using student trace logs to determine meaningful progress and struggle during programming problem solving. In *Proceedings of International Conference on Educational Data Mining*. ERIC, 2021.

[5] G. Feng, B. Zhang, Y. Gu, H. Ye, D. He, and L. Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[6] H. Funayama, Y. Asazuma, Y. Matsubayashi, T. Mizumoto, and K. Inui. Reducing the cost: Cross-prompt pre-finetuning for short answer scoring. In *International Conference on Artificial Intelligence in Education*, pages 78–89. Springer, 2023.

[7] H. Funayama, T. Sato, Y. Matsubayashi, T. Mizumoto, J. Suzuki, and K. Inui. Balancing cost and quality: an exploration of human-in-the-loop frameworks for automated short answer scoring. In *International Conference on Artificial Intelligence in Education*, pages 465–476. Springer, 2022.

[8] R. Gao, H. E. Merzdorf, S. Anwar, M. C. Hipwell, and A. Srinivasa. Automatic assessment of text-based responses in post-secondary education: A systematic review. *Computers and Education: Artificial Intelligence*, page 100206, 2024.

[9] M. M. Grant and R. M. Branch. Project-based learning in a middle school: Tracing abilities through the artifacts of learning. *Journal of Research on technology in Education*, 38(1):65–98, 2005.

[10] P. Guo, N. Saab, L. S. Post, and W. Admiraal. A review of project-based learning in higher education: Student outcomes and measures. *International journal of educational research*, 102:101586, 2020.

[11] Z. Z. GUVEN. Project based learning: A constructive way toward learner autonomy. *International Journal of Languages' Education and Teaching*, 2(3):182–193, 2014.

[12] G. A. Katuka, Y. Auguste, Y. Song, X. Tian, A. Kumar, M. Celepkolu, K. E. Boyer, J. Barrett, M. Israel, and T. McKlin. A summer camp experience to engage middle school learners in ai through conversational app development. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 813–819, 2023.

[13] J. H. L. Koh, S. C. Herring, and K. F. Hew. Project-based learning and student knowledge construction during asynchronous online discussion. *The Internet and Higher Education*, 13(4):284–291, 2010.

[14] D. Kokotsaki, V. Menzies, and A. Wiggins. Project-based learning: A review of the literature. *Improving schools*, 19(3):267–277, 2016.

[15] H. Kumar, I. Musabirov, M. Reza, J. Shi, A. Kuzminykh, J. J. Williams, and M. Liut. Impact of guidance and interaction strategies for llm use on learner performance and perception. *arXiv preprint arXiv:2310.13712*, 2023.

[16] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

[17] A. Mannekote. LLM4Qual. `https://github.com/msamogh/llm4qual`, Jan. 2024.

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

[19] A. Mizumoto and M. Eguchi. Exploring the potential of using an ai language model for automated essay scoring. *Research Methods in Applied Linguistics*, 2(2):100050, 2023.

[20] C. Mouza, A. Marzocchi, Y.-C. Pan, and L. Pollock. Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education*, 48(2):84–104, 2016.

[21] B. Naismith, P. Mulcaire, and J. Burstein. Automated evaluation of written discourse coherence using gpt-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, 2023.

[22] T. Phung, J. Cambronero, S. Gulwani, T. Kohn, R. Majumdar, A. Singla, and G. Soares. Generating high-precision feedback for programming syntax errors using large language models. *arXiv preprint arXiv:2302.04662*, 2023.

[23] K. Pliakos, S.-H. Joo, J. Y. Park, F. Cornillie, C. Vens, and W. Van den Noortgate. Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Computers & Education*, 137:91–103, 2019.

[24] T. Price, R. Zhi, and T. Barnes. Evaluation of a data-driven feedback algorithm for open-ended programming. In *Proceedings of International Conference on Educational Data Mining*. ERIC, 2017.

[25] R. Pucher and M. Lehner. Project based learning in computer science–a review of more than 500 projects. *Procedia-Social and Behavioral Sciences*, 29:1561–1566, 2011.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[27] Y. Shi, Y. Mao, T. Barnes, M. Chi, and T. W. Price. More with less: Exploring how to use deep learning effectively through semi-supervised learning for automatic bug detection in student code. In *Proceedings of International Conference on Educational Data Mining*, 2021.

[28] Y. Shi, K. Shah, W. Wang, S. Marwan, P. Penmetsa, and T. Price. Toward semi-automatic misconception discovery using code embeddings. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 606–612, 2021.

[29] A. Shibani, R. Rajalakshmi, F. Mattins, S. Selvaraj, and S. Knight. Visual representation of co-authorship with gpt-3: Studying human-machine interaction for effective writing. In *Proceedings of International Conference on Educational Data Mining*. ERIC, 2023.

[30] Y. Song, G. A. Katuka, J. Barrett, X. Tian, A. Kumar, T. McKlin, M. Celepkolu, K. E. Boyer, and M. Israel. Ai made by youth: A conversational ai curriculum for middle school summer camps. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Innovative Applications of Artificial Intelligence Conference and Thirteenth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2023.

[31] X. Tian, A. Kumar, C. E. Solomon, K. D. Calder, G. A. Katuka, Y. Song, M. Celepkolu, L. Pezzullo, J. Barrett, K. E. Boyer, et al. Amby: A development environment for youth to create conversational agents. *International Journal of Child-Computer Interaction*, 38:100618, 2023.

[32] B. Vander Zanden and M. W. Berry. Improving automatic code assessment. *Journal of Computing Sciences in Colleges*, 29(2):162–168, 2013.

[33] M. Walker, C. Kamm, and D. Litman. Towards developing general models of usability with paradise. *Natural Language Engineering*, 6(3-4):363–377, 2000.

[34] W. Wang, G. Fraser, T. Barnes, C. Martens, and T. Price. Automated classification of visual, interactive programs using execution traces. In *Proceedings of International Conference on Educational Data Mining*, pages 677–681, 2021.

[35] M. H. Wilkerson-Jerde. Construction, categorization, and consensus: Student generated computational artifacts as a context for disciplinary reflection. *Educational Technology Research and Development*, 62:99–121, 2014.

[36] M. Zhang, S. Baral, N. Heffernan, and A. Lan. Automatic short math answer grading via in-context meta-learning. In *Proceedings of International Conference on Educational Data Mining*, pages 122–132, 2022.

[37] Z. Zhang, A. Zhang, M. Li, and A. Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

[38] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. (arXiv:2102.09690), June 2021. arXiv:2102.09690 [cs].

APPENDIX

A. ADDITIONAL EXAMPLES OF COMPARISON BETWEEN HUMAN GRADER'S AND LLM'S GRADES AND RATIONALES

Table 4: Comparison between human grader's and LLM's rationales for evaluating the *responses* in a learner artifact. Colored texts highlight the main discrepancy between human and LLM-generated rationales (similarly for Table 5).

Evaluation	Score	Rationale
Human	3	Most contain one proper response that is logical and conversational with some conversational markers. However, I cannot give a 4 because **some responses are extremely long**. For example, "*Buckle Up! There are a lot of Penalty Umpire Signals! There are 11 signals in this division: Out, Not Out, No Ball, Free Hit, Wide Ball, Bouncer, DRS(Third Umpire), Dead Ball, Short Run, Penalty Runs, Revoke Decision, and Cancel Call. Out: [.....]*" This response contains almost 200 words to describe the different signals in a cricket ball. **This long utterance makes the chatbot talks in monologue instead of engaging users in a dialogue.**
GPT-4: few-shot-rubric	4	The responses are abundant, logical, and error-free. They contain hints to keep the conversation going and utilize conversational markers throughout the intents when appropriate. For example, after explaining the rules of cricket, the AI suggests asking about fun facts, famous players, types of no balls, or umpire signals. This keeps the conversation flowing and engages the user. **The responses are also of appropriate length and are mostly free from grammatical errors.**

B. FULL DESCRIPTION OF CHATBOT ARTIFACT EVALUATION RUBRIC

Table 5: Comparison between human grader's and LLM's rationales for evaluating the *training phrases* in a learner artifact.

Evaluation	Score	Rationale
Human	4	The training phrases are in great shape. **It consistently contains more than three (system required) training phrases per intent, and has plenty of linguistic variations among the phrases.** For example, some uses keywords, while the others are mixed sentences.
GPT-4: few-shot-rubric	3	The training phrases are ample and varied within the intent and also differ from those in other intents. They present variations in either syntactic structure or lexicon choices. **However, they do not consistently capture edge cases**, which is why the score is not a 4.

Table 6: **Full Description of Conversational AI Artifact Evaluation Rubric.**

Categories	**Artifact Dimensions**	**1. Little to no evidence of approaching expectations**	**2. Approaching Expectations**	**3. Meets Expectations**	**4. Exceeds Expectations**
Conversational design	Follow up intents	No follow-up intent	At least one follow-up intent OR most follow-up intents do not logically match with its parent intent OR they are unnecessary or repeated	Multiple logical follow-up intents AND Each follow-up intent is related to its parent intent mostly logically	All follow-up intents are logically related to main intent, numerous, and mutually exclusive
Conversational design	Greet intent	No customized greet response	At least one customized greet intent, however the purpose is not clear or actionable	At least one customized greet intent demonstrating its purpose. May not set exact user expectations: ("Ask me for song recommendations", "hey im blah bot do you need any assistance on video games?")	Effectively greet the user, introduce the chatbot, and demonstrate the purpose. AND Set exact user expectations (e.g., "I can talk about pop or hip hop music") or clearly directs the user for next steps (e.g., "simply state 'quiz me on math'")
Conversational design	Default fallback intent	No customized fallback response	The response is customized, however it cannot not redirect the users (e.g., "I didn't get that. Try it again.")	The response is customized and can redirect the users (e.g., "I didn't get that as I'm still learning. I'm more confident to talk about XYZ instead.")	The agent has multiple varied, customized and meaningful responses that can redirect the users
AI Development	Training phrases	The amount of training phrases is limited (less than system requirement) OR Most of training phrases are random in the customized intents	The amount of training phrases meet the system requirement, but the content does not show enough linguistic variations (syntactically and lexically) within the intent or topic variations across different intents	Most training phrases are ample, cohesive and varied within the intent; also differ from those in other intents. They present variations in either syntactic structure or lexicon choices	The project contains consistently more varied training phrases than what the system requires, which can capture some edge cases. Training phrases are given and they are unique in both lexical and syntactic structure
AI Development	Responses	The responses are random in most of the customized intents	Most Responses (60%+) are provided either too long or too short, or lack of information or contains grammatical errors that impede user's understanding If there are multiple responses, the content is not consistent enough to trigger similar user reactions Example: "Bad Romance by Lady Gaga" - not conversational	Most customized intents contain at least one response that is in proper length, logical, mostly free of grammatical errors, mostly mimic/display natural and conversational, may include some conversational markers.	Intents contain multiple logical, error-free responses OR The responses contain hints to keep the conversation going (e.g., "Alligators are dangerous animals... Now, do you want to learn about other animals?) OR Utilize the conversational markers throughout the customized intents when appropriate