Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #48952

Investigating the Impact of Codio Coach: A Specialized AI Learning Assistant on Computing Student Engagement and Performance

Mohit Chandarana, Codio

Mohit has a BE in Computer Engineering and an MS in Computer Science. From generating insightful learning analytics for CS Educators to prototyping novel product features and algorithms, he works towards bridging the gap between cutting-edge academic research and its application in the industry in his role at Codio.

Sindhu Ramachandra, Codio

A Data Science professional with a foundation in data analytics, large language models (LLMs), and prompt engineering, currently expanding expertise at COdio. Skilled in extracting insights from complex datasets, with formal training through certification courses in Data Science. Holds a Master's degree in Biochemistry and has research experience from the prestigious Indian Institute of Science (IISc), Bangalore.

Mr. Joshua Ball, Codio

Joshua Ball is Codio's Vice President of Marketing and a Senior Fellow at the National Institute for Deterrence Studies. He has a MA in International Relations from the University of St Andrews.

Maura Lyons, Codio
Phillip Snalune, Codio

Investigating the Impact of Codio Coach: A Specialized AI Learning Assistant on Computing Student Engagement and Performance

Abstract

Recent research has demonstrated significant advancements in the applications of Large Language Models (LLMs) in educational environments, particularly in delivering immediate, personalized student feedback. This study examines the impact of Codio Coach, a specialized AI learning assistant integrated into the Codio platform, on student engagement and performance in asynchronous MOOC-style computer science courses.. It utilizes Large Language Models (LLMs) to provide support without supplying direct answers. It consists of three modules: Summarizer, which simplifies assignment instructions; Error Explanation, which clarifies programming error messages; and Hints, which provides Socratic-style hints by posing questions or suggestions to guide students toward solutions.

Analysis revealed an immediate and sustained uptake in assistant usage, with "Explain this error" being the most frequent interaction (56.3%), confirming engagement and highlighting student need for error comprehension support. Assignments where Coach was enabled showed improved student performance, with a 12% increase in Mean Grade and a 15% increase in Median Grade. Furthermore, an impressively low error event rate (0.12%) observed in these AI-assisted courses suggests early signs that such tools may contribute to more effective programming environments.

These findings provide valuable evidence for the efficacy of tailored AI learning assistants in enhancing student engagement and performance in CS education. We recommend educators guide students in leveraging custom, context-specific assistants to improve learning and develop critical AI application skills.

Introduction

Large Language Models (LLMs) enable educational platforms to support students through advanced tools with real-time personalized feedback, guidance, and engagement mechanisms. By employing methods like retrieval-augmented generation (RAG), LLMs are increasingly able to overcome challenges related to scalability and handling unexpected or unforeseen inputs, as are often experienced with intent-based chatbots [1]. RAG-powered assistants demonstrate significantly improved performance in terms of response accuracy, adaptability, and student satisfaction [2].

This study examines the impact of Codio Coach, an AI learning assistant within the Codio platform specifically designed to support computing education, on student engagement and learning outcomes. Codio Coach focuses on three primary functionalities: error explanation,

assignment summarization, and next-step guidance, aiming to facilitate independent problem-solving without directly providing answers.

Previous research has found that computing students primarily use Generative AI (GenAI) tools to understand complex jargon, such as teacher-written programming assignment prompts and developer-written compiler messages, rather than to generate code. Lyons et al. [3] surveyed 371 post-secondary computing students and found that 65% used GenAI tools to complete programming tasks, with 45% using them to interpret homework or project prompts and 40% to explain code. These findings highlight the role of GenAI tools in enhancing comprehension and supporting coursework navigation, and align with educational trends that emphasize adaptive, personalized learning environments while maintaining academic integrity [4].

Background

Towards Effective Teaching Assistants: From Intent-Based Chatbots to RAG-Enabled LLMs

Alsafari et al. [2] explore the evolution of AI assistants in education, comparing traditional intent-based chatbot systems to modern retrieval-augmented generation (RAG) approaches powered by LLMs. The study highlights RAG-based systems' superior accuracy and adaptability, which dynamically retrieve and generate responses tailored to specific queries. Key findings include significant improvements in response accuracy (85-90%) and scalability, with RAG systems resolving 95% of student queries compared to 50% for intent-based systems. The authors emphasize the need for enriched datasets and dynamic retrieval methods to enhance chatbot performance and address unforeseen queries. These findings are built on foundational work, such as Lewis et al. (2020) [1], which examined the principles of RAG systems in AI education.

Evaluating the Effectiveness of LLMs in Introductory Computer Science Education

Lyu et al. [5] conducted a semester-long study on the impact of an LLM-powered virtual teaching assistant, CodeTutor, in introductory programming courses. The experimental group using CodeTutor demonstrated a 12.5-point improvement in scores compared to a 3.17-point decrease in the control group. Regression analysis revealed that first-time LLM users benefited most, achieving an 18.88-point boost in performance. This builds on earlier investigations by Zhang et al. [6], who explored the role of adaptive AI feedback in enhancing learner outcomes in STEM fields.

Experiences from Integrating Large Language Model Chatbots into the Classroom

Hellas et al. [7] investigated the use of GPT-4-powered chatbots in three university courses. Their findings reveal varied engagement levels, with the chatbot being most effective in courses aligned with its capabilities, such as software engineering. The study highlights the importance of tailoring chatbot functionality to specific course requirements to maximize effectiveness. Earlier work by Bender et al. [8] provided the groundwork for understanding the limitations of general-purpose chatbots in specialized learning environments.

Future of Education: AI and MOOCs

Verma et al. [9] examine the role of AI in enhancing MOOCs, emphasizing personalized learning and automated feedback. AI-powered tools have been shown to significantly improve learner retention and engagement by tailoring content to individual needs. However, ethical concerns such as data privacy and algorithmic bias remain critical. Verma et al. (2024) draw on the analytics framework proposed by Kumar et al. [10], which emphasizes proactive intervention strategies in MOOC platforms.

In favor of embracing a measured approach of introducing Generative AI in computing education, we decided to instrument Codio Coach, a conversational AI learning assistant with the following three modules available to learners:

a) Summarize what I need to do
    - Provides learners with a simple summary of the programming assignment's tasks as well as a list of requirements based on the question specification.
b) Explain this error
    - Offers plain English explanations for compiler error messages, pinpointing the cause of the error.
    - The explanation is concise (3-5 sentences) with no fixes or solutions.
    - If applicable, it also underlines common misconceptions relevant to the encountered error message.
c) Provide a hint on what to do next
    - Provides context-specific hints to help students make progress.
    - Hints are Socratic-style - phrased as questions or suggestions based on the question specification, a learner’s current progress towards the solution as well as any error messages that they’ve encountered when hints are requested.

We begin by working towards answering the following research questions:

RQ1: Did Error Explanation Assistant have an impact on error resolution time as compared to previous data without LLM?

RQ2: Did LLM assistance help improve student performance as compared to previous/historical data without AI assistants?

RQ3: What was the relative impact of each of the assistants on student engagement and performance compared to previous data without LLM assistance?

Methods

Dataset

This study investigates the impact of these assistants on student engagement and performance in computing courses by analyzing process metrics such as time spent, error resolution duration, and engagement patterns.

To calculate these metrics, we collected data from thousands of learners around the world who used the Codio platform between Jan 2023 and Dec 2024, following IRB-approved anonymization and consent protocols. The data was generated from MOOC-style, asynchronous courses taught in a wide variety of languages. Our system logged an 'event' data point every time a learner compiled and/or ran their code, as well as on assessment submissions - that consists of a User ID, Assignment ID, Course ID, Timestamp, Exit Code, Command and Output. The system is also instrumented to log 'assistant-event' data points that describe learner interactions with all three modules of the AI learning assistant, consisting of all the associated IDs like the event data point, along with the Assistant Type, Page Content, Code File Content, Error Message, and the Assistant Response.

Our system also computes the time spent on assignments by logging the start and end timestamps for each active student session with the associated assignment and course IDs. It also automatically filters out periods of inactivity. We combined all the system-collected event data with assignment grades as seen in our system. This dataset represents 9733 unique students, with a total of 973 assignments across 60 unique courses.

Implementation

We started by splitting the dataset into 2 parts - data before and after the integration of AI assistants, respectively. We then combined the time spent and assessment grades data.

Then, we calculated the following metrics at the assignment level for comparison:

a) Total, Mean and Median time spent in assignment
b) Mean and Median assignment grades

In order to understand learner engagement patterns, we decided to combine the events, assistant-events datasets and sessions data, to build a process dataset that splits all learner process data by session activity.

This led to the calculation of the following metrics:

a) Total, Mean and Median learner session counts per assignment
b) Mean and Median time elapsed between the started and completed timestamps for each assignment

We also calculated the mean resolution times for errors encountered by learners by analyzing the event result and output of subsequent compile, run and submission events. Since all events are tagged by timestamps in our system, we define error resolution time as follows:

error_resolution_time = time elapsed between A and B where

A: an event that resulted in an error state
B: a subsequent/following event that resulted in an error-free state

By combining events and sessions data, we were able to filter out periods of inactivity out of the resolution time calculations.

We then compared the results of these calculations before and after the assistants were integrated.

Analysis and Results

Within approximately 3 months of integrating the AI assistants in 60 courses on Coursera, we registered:

9000+ API requests
by **1800+** unique students
in **350+** unique assignments
from **39** unique courses

Table 1: AI assistant usage data

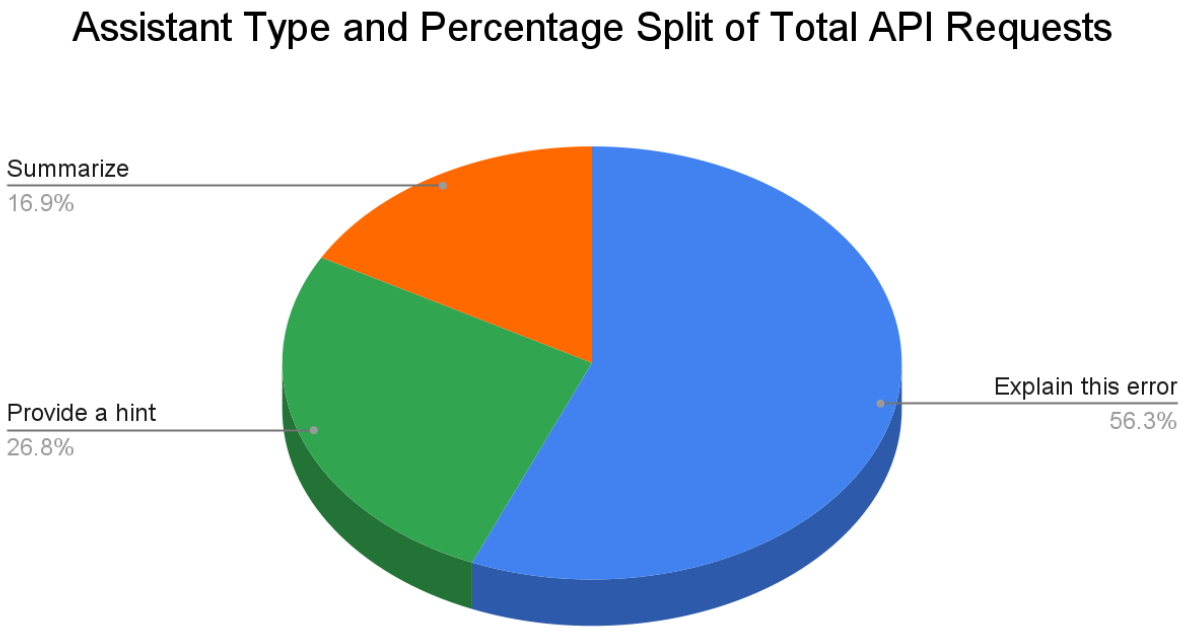

Figure 1: Assistant Type and Percentage Split of Total API Requests (9000+)

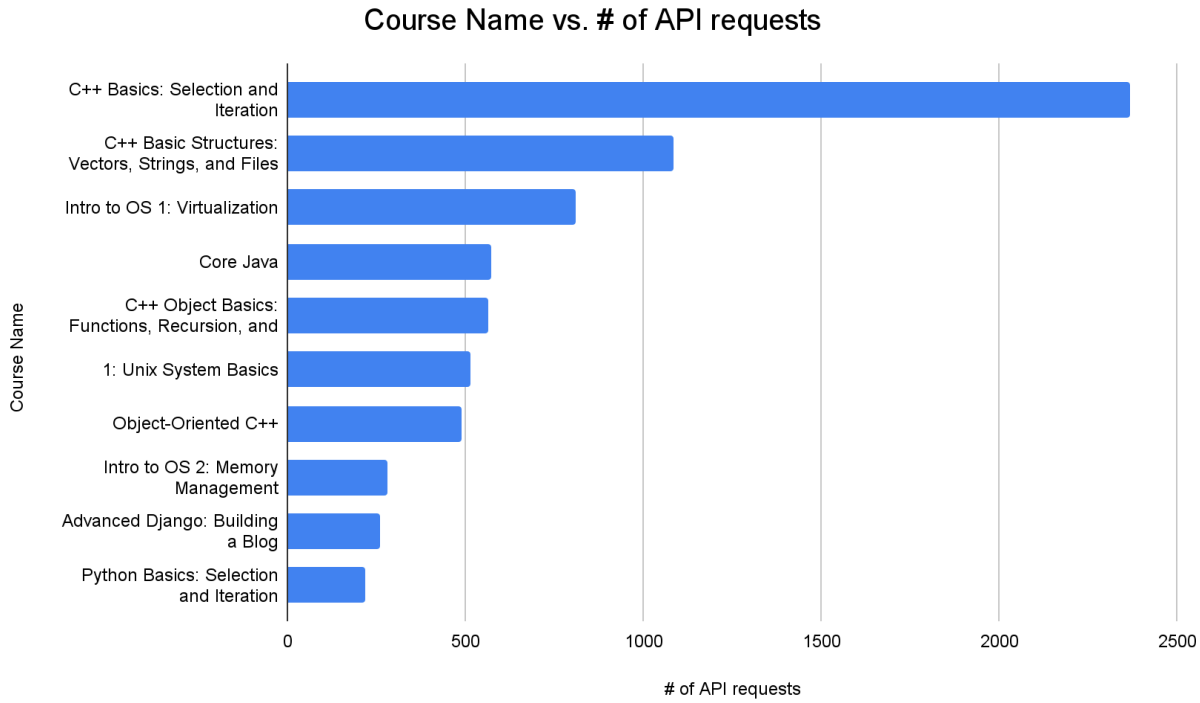

Figure 2: Course name and # of API requests

In terms of usage, the Explain this error assistant was more prevalent among learners. (Fig. 1)

This was consistent with findings by Lyons et al. [3] where students reported understanding error messages as one of the top 3 reasons they use Generative AI applications.

We also noted that the top 10 courses accounted for approximately 80% (~7000) of the total API requests. (Fig. 2)

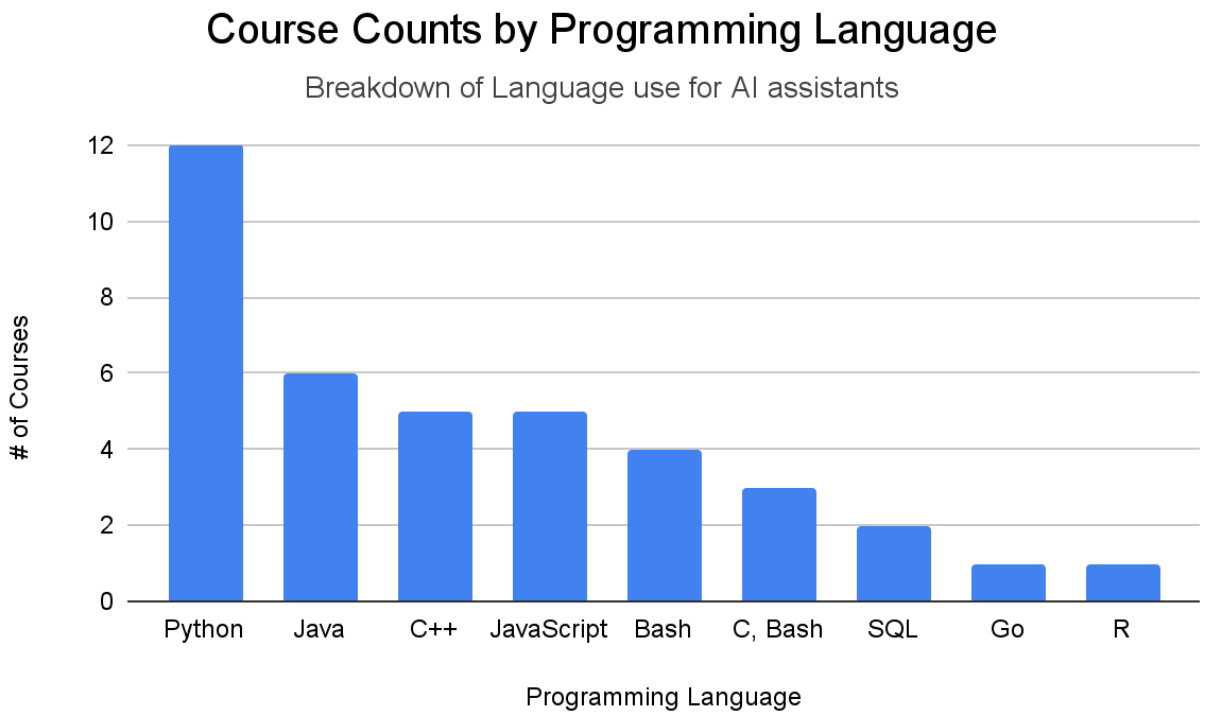

Figure 3: Course Counts by Programming Language

On further analysis of the assistants usage dataset, we found that AI usage was seen across a wider range of courses in Python than courses in any other programming language. (Fig. 3) This may speak to the rising popularity of Python based courses in the past couple years.

However, C++ courses logged the most usage of AI assistants - 4 out of the top 10 courses by number of API requests were C++ courses. (Fig. 2)

We also contrasted AI use with course difficulty and found that Beginner and Intermediate level courses reported higher use of AI assistants than Advanced courses. (Fig 4)

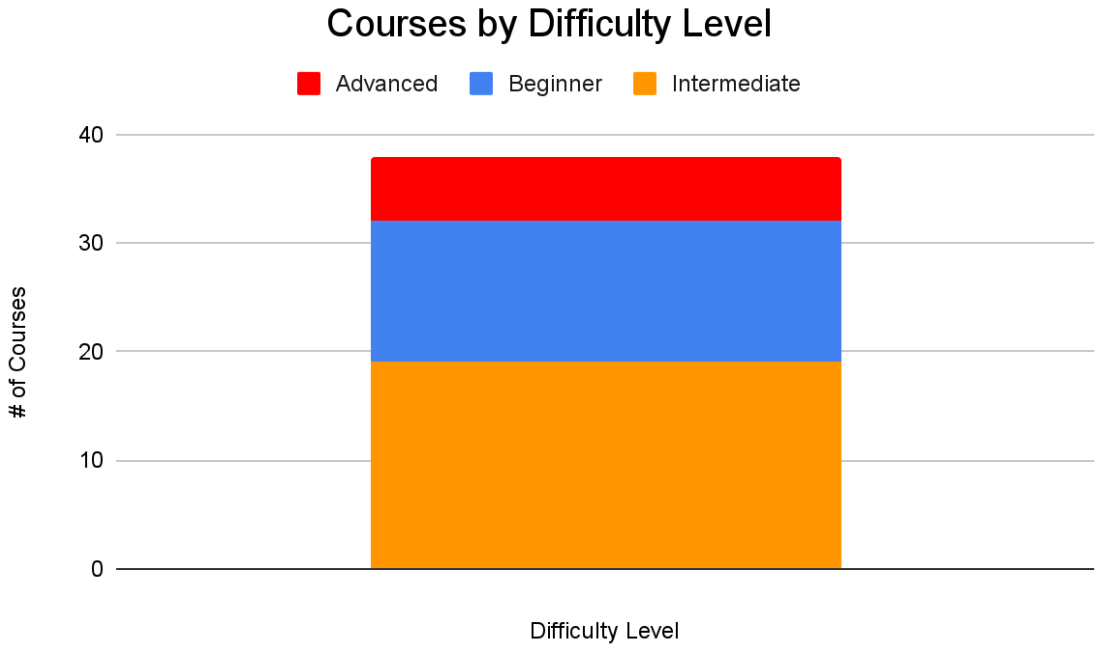

Figure 4: Course Counts by Difficulty Level

Results

RQ1: Did Error Explanation Assistant have an impact on error resolution time as compared to previous data without LLM?

	Error Message	Avg. time to fix (seconds)
1	error: #include expects "FILENAME" or <FILENAME>	743.333333
2	error: reference is ambiguous	362.000000
3	error: expected type-specifier	331.500000
4	error: braces around scalar initializer	237.000000
5	error: type invalid for identifier	234.000000

Table 3. Avg. resolution time in C++ courses

Table 3 shows the top 5 errors with the highest mean resolution time in historical C++ course data.

We hypothesized that there would be similarities in the types of errors encountered by learners after the integration of AI assistants as well - and we'd be able to compare these resolution times to estimate the impact of having an error explanation assistant.

Interestingly, we found that there were astonishingly low numbers of error events (0.12% of total events) in the courses where the AI learning assistant was integrated - thereby rendering this research question unanswered due to lack of data. One possible explanation is the fact that asynchronous MOOC-style courses have very different engagement and completion patterns. This might also be an indicator of really effective programming environments and to truly understand the effect of AI assistants in the context of error explanations, a more long form study needs to be done.

This has sparked a whole new set of inquiries that we will have to make as part of the future work to understand the following:

1. Which courses and topics are learners using AI assistants to help explain error messages?
2. What kinds of programming errors are learners having difficulty with?

RQ2: Did LLM assistance help improve student performance as compared to previous/historical data without AI assistants?

assignment_id	course_id	avg_grade	median_grade	avg_grade_without_llm	median_grade_without_llm
04e5470cb8e602c517bfe0deb469025a	a06f327ccc4d59b7057e351cdfb88b14	100.0	100.0	87.5	100.0
09fbae5746c5897c79619dac110cee85	c99c394544264f4054088963606be832	84.54820797576980	100.0	38.79582847519110	20.0
1dc9d997052df63adbcdc08e16fe0ed1	11a0f3ccd1058fb878440367d23b39dd	93.95555555555560	92.0	89.05151515151520	100.0
1f18bbbfe069a1ef1ee4469131fc7ac6	e11ecde969ec9dcae54c2f627f1e46c2	75.58620689655170	100.0	81.64112530075880	88.0
24395099a4d1e8c97e2abef6d4f714f7	cbd0f551d75f345401aa0a5e6181a7f5	34.09090909090910	25.0	78.80952380952380	100.0
2afd4033955d6d50e1be9df780358bfa	cbd0f551d75f345401aa0a5e6181a7f5	84.74015748031500	87.0	87.06662040249830	100.0

Table 4: Mean and Median Grades Data Snapshot

Table 4 shows the snapshot of what the final table calculation looks like for the mean and median grades, with and without LLM assistance. In the final dataset of 79 unique assignments, results indicate that the Mean Grade increased by 12% and Median Grade increased by 15% in assignments with LLM assistance.

RQ3 and Time Spent and Engagement metrics

Our preliminary analysis on time spent data in assignments, session counts and durations, revealed the need for extensive data collection (for the population with LLM assistance) to be able to map and compare these metrics at scale and at an individual assistant level. We plan to address this in future work.

Conclusion

Among assistant types, “Explain this error” was most used (56.32%), followed by “Provide a hint with what to do next” (26.77%) and “Summarize what I need to do” (16.91%), consistent with Lyons et al. (2024), which highlighted error understanding as a top reason for using generative AI. We were pleased to see an increase in Mean Grade of 12% and a Median Grade of 15% in assignments where Coach was enabled. The impressively low error event rate (0.12%) observed in AI-assisted courses reveal early signs of effective programming environments, outlining future work for broader, in-depth data collection approaches.

Our findings also highlight an opportunity to refine these metrics to better capture the nuanced learning dynamics present in diverse online educational environments at scale. Building on these insights, we propose that a comprehensive long-form study examining how learners engage with these assistants will help us understand and leverage this rich new landscape of learner process data.

We saw an immediate and continuing uptick in AI assistants usage as soon as they were integrated into the courses. This suggests that taking a measured approach and building tailored assistants seems to have a positive impact on learner engagement with these tools as well as the courses. We recommend that educators show learners how to best utilize Generative AI in the form of custom, context-specific assistants as part of their learning journey, not only improving engagement but also empowering learners to develop critical skills in applying AI tools effectively and appropriately.

Future research should focus on assessing their impact on different learning outcomes across a range of learning contexts and understanding the specific needs of varying learner populations to refine and enhance their effectiveness at scale.

References

[1] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., & Zettlemoyer, L. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 2144-2161). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.214

[2] Alsafari, B., Atwell, E., Walker, A., & Callaghan, M. (2024). Towards effective teaching assistants: From intent-based chatbots to LLM-powered teaching assistants. *Natural Language Processing Journal*, 8, 100101. https://doi.org/10.1016/j.nlp.2024.100101

[3] Lyons, M., & Deitrick, E., & Ball, J. R. C. S. (2024, June), Characterizing Computing Students' Use of Generative AI. In *Proceedings of the 2024 ASEE Annual Conference & Exposition*, Portland, Oregon. https://doi.org10.18260/1-2--48453

[4] M. A. Cardona, R. J. Rodriguez, and K. Ishmael, "Artificial Intelligence and the Future of Teaching and Learning Insights and Recommendations," *Office of Educational Technology*, May 2023. Available: https://www2.ed.gov/documents/ai-report/ai-report.pdf

[5] Lyu, W., Wang, Y., Chung, T. (R.), Sun, Y., & Zhang, Y. (2024). Evaluating the effectiveness of LLMs in introductory computer science education: A semester-long field study. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (pp. 63–74). Association for Computing Machinery. https://doi.org/10.1145/3657604.3662036

[6] Zhang, Y., Sun, Y., & Li, Q. (2023). Adaptive feedback systems in STEM education: An AI-powered approach. *International Journal of STEM Education*, 10(1), 12-25. https://doi.org/10.1186/s40594-023-00321-4

[7] Hellas, A., Leinonen, J., & Leppänen, L. (2024). Experiences from integrating large language model chatbots into the classroom. In *Proceedings of the 2024 ACM Virtual Global Computing Education Conference* (pp. 46–52). Association for Computing Machinery. https://doi.org/10.1145/3649165.3690101

[8] Bender, E., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2022). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623). Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922

[9] Verma, A., Lajwanti, & Gautam, A. (2024). Future of education: AI and MOOCs. *The Academic*, 2(10), 366. ISSN: 2583-973X. https://doi.org/10.5281/zenodo.14102780

[10] Kumar, R., Lajwanti, & Gautam, A. (2023). Analytics-driven interventions in MOOCs: Enhancing student success. *Journal of Educational Data Science*, 5(2), 45-63. https://doi.org/10.1234/edu.2023.56789