Received 27 April 2024, accepted 19 June 2024, date of publication 28 June 2024, date of current version 1 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420709

RESEARCH ARTICLE

MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning

CHING NAM HANG[1], **(Member, IEEE), CHEE WEI TAN**[2], **(Senior Member, IEEE), AND PEI-DUO YU**[3]

[1] Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong
[2] College of Computing and Data Science, Nanyang Technological University, Singapore 639798
[3] Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan 320314, Taiwan

Corresponding author: Pei-Duo Yu (peiduoyu@cycu.edu.tw)

This work was supported in part by the Nanyang Technological University (NTU) startup fund; in part by the EdeX Grant (No. 03INS001595C130) from the NTU Centre for Teaching, Learning and Pedagogy; and in part by the National Science and Technology Council of Taiwan, under Grant 112-2115-M-033-004-MY2.

ABSTRACT In the dynamic landscape of contemporary education, the evolution of teaching strategies such as blended learning and flipped classrooms has highlighted the need for efficient and effective generation of multiple-choice questions (MCQs). To address this, we introduce *MCQGen*, a novel generative artificial intelligence framework designed for the automated creation of MCQs. MCQGen uniquely integrates a large language model (LLM) with retrieval-augmented generation and advanced prompt engineering techniques, drawing from an extensive external knowledge base. This integration significantly enhances the ability of the LLM to produce educationally relevant questions that align with both the goals of educators and the diverse learning needs of students. The framework employs innovative prompt engineering, combining chain-of-thought and self-refine prompting techniques, to enhance the performance of the LLM. This process leads to the generation of questions that are not only contextually relevant and challenging but also reflective of common student misconceptions, contributing effectively to personalized learning experiences and enhancing student engagement and understanding. Our extensive evaluations showcase the effectiveness of MCQGen in producing high-quality MCQs for various educational needs and learning styles. The framework demonstrates its potential to significantly reduce the time and expertise required for MCQ creation, marking its practical utility in modern education. In essence, MCQGen offers an innovative and robust solution for the automated generation of MCQs, enhancing personalized learning in the digital era.

INDEX TERMS Large language models, multiple-choice questions, personalized learning, prompt engineering, retrieval-augmented generation.

I. INTRODUCTION

In the rapidly changing landscape of education, blended learning and classroom flipping have emerged as innovative pedagogical strategies that breathe new life into the learning experience [12], [25], [26]. Blended learning synergizes the strengths of both online and face-to-face instruction, offering flexibility and personalized learning paths [7], while classroom flipping shifts the lecture component online, freeing classroom time for interactive and engaging activities [17]. Within these novel teaching methodologies, multiple-choice questions (MCQs) find an essential place, serving as quick, adaptable tools for both assessment and engagement [5]. They align with the on-the-go nature of these strategies, allowing for rapid feedback and tailored difficulty levels. However, the process of developing effective MCQs that align with diverse learning needs and subjects is far from simple. It is time-consuming, demands expert knowledge, and, without careful consideration, can result in generic or uninspiring questions. This challenge necessitates innovative solutions that not only facilitate the creation of quality MCQs but also resonate with the dynamic, student-centered essence

The associate editor coordinating the review of this manuscript and approving it for publication was Ka Wai Gary Wong.

of blended learning and flipped classrooms. Thus, the stage is set for exploring automated means of generating MCQs, marking a crucial advancement in modern education.

As artificial intelligence (AI) continues to evolve, its potential as a solution in the field of education becomes increasingly evident [13], [27], [31]. Building on the need for efficient and effective MCQ creation, the emergence of generative AI presents a promising solution to these educational challenges. Large language models (LLMs) [4], [20], [21], [28] have made notable impacts across various domains, including natural language processing (NLP), data analysis, and, more recently, the field of education. Their ability to generate contextually relevant and coherent text positions them as potential agents for automating the generation of educational content, especially MCQs. While this prospect is appealing, current implementations often fail to fully capture the detailed understanding of varying difficulty levels and the diversity of perspectives essential for blended learning and flipped classrooms. There is a clear and compelling opportunity to refine the use of LLMs for question generation, making them more aligned with the educational objectives and the diverse needs of modern learners. The assurance of quality in automatically generated questions and their consistency with the personalized, learner-focused ethos of contemporary education are areas ripe for exploration and innovation, emphasizing the need for a more targeted and advanced approach.

In this paper, we present *MCQGen*, an innovative generative AI framework designed for the automated generation of MCQs. Its unique feature lies in integrating an LLM, Generative Pre-trained Transformer 4 (GPT-4) [20], with retrieval-augmented generation (RAG) [11] and advanced prompt engineering techniques. The framework draws from an external knowledge base through RAG, which enhances the ability of the LLM to produce questions that are aligned with educational objectives and student learning requirements. The joint use of chain-of-thought (CoT) [14], [29], [30] and self-refine [15] within our prompt engineering process optimally enhances the LLM, leading to the creation of questions that are not only relevant and challenging but also reflective of common student misconceptions and errors. This comprehensive prompt engineering strategy effectively generates a set of MCQs that contribute to a personalized learning experience, driving engagement and deepening understanding among students. The framework also incorporates a feedback mechanism through crowdsourcing, where student performance on these questions informs further refinement of the MCQ generation process, embodying a continuous cycle of learning and improvement. As a result, MCQGen offers a robust solution for the automated generation of MCQs, effectively contributing to the enhancement of personalized education.

Overall, the contributions of the paper are as follows:

- We propose MCQGen, a novel framework that jointly combines an LLM with optimized prompt engineering and RAG. This integration is designed for the automated generation of MCQs, offering innovative insights within the spheres of blended learning and flipped classrooms.
- We present a unique dataset that serves as a comprehensive database for RAG, encompassing both instructor-designed and student-created MCQs. This collection is distinctively organized to cater to various difficulty levels and incorporates diverse creative insights. The dataset includes quality-assured, difficulty-categorized MCQs crafted by instructors, complemented by a range of student-created questions. This rich compilation lays a solid foundation for fine-tuning an LLM in educational applications, facilitating the generation of high-quality and diverse automated questions.
- We develop a robust prompt engineering strategy tailored to optimize the effectiveness of the LLM within the proposed framework. This strategy incorporates the joint application of chain-of-thought and self-refine prompting techniques, which collectively enhance the capability of the LLM to produce questions that are not only relevant and challenging but also resonate with common student misconceptions and mistakes. Our advanced prompt engineering approach is designed to create MCQs that enhance personalized learning experiences and boost student engagement, demonstrating the potential of prompt engineering in creating tailored learning content.
- We conduct extensive evaluations on the proposed framework, demonstrating its effectiveness in generating relevant and quality MCQs that align with varying educational needs and learning styles. Our results highlight the potential of the proposed framework to significantly reduce the time and expertise required for MCQ creation, showcasing its applicability in modern education.

This paper is organized as follows. In Section II, we examine existing literature and foundational studies in the automation of MCQ generation. Section III introduces MCQGen, which integrates an LLM with RAG and advanced prompt engineering for the automated creation of MCQs. The effectiveness of MCQGen in producing quality MCQs is assessed in Section IV. Student responses to their learning experiences with the generated MCQs are analyzed in Section V. In Section VI, further discussion on implications and insights of the MCQGen framework is presented, followed by an exploration of potential limitations and avenues for future research in Section VII. We conclude the paper in Section VIII.

II. BACKGROUND AND RELATED WORK

MCQs are a widely recognized assessment format where respondents are tasked with identifying the most accurate answer from a range of options [3], [23]. The popularity of this method stems from its versatility and efficiency in evaluating a broad spectrum of knowledge, from simple factual recall to complex problem-solving [22]. MCQs facilitate the swift grading of student responses and provide clear metrics for performance analysis. They are particularly conducive to large-scale educational settings where consistent and objective assessment is necessary. MCQs also support personalized learning by enabling the assessment of individual student responses to identify specific areas of strength and weakness. This can inform subsequent instruction and provide students with tailored feedback, contributing to a learning experience that adapts to their unique educational journey. The clear format of MCQs aids in reducing ambiguity in student responses, leading to more accurate assessments of student knowledge.

An MCQ is constructed using three fundamental components: (1) stem, (2) key, and (3) distractors [8]. The stem, also known as the item or question sentence, forms the basis of the question. This is the part that presents the problem or query to be answered, and it can stand alone as a question without the list of possible answers. The stem may be structured in either an assertive or interrogative format. The key, sometimes referred to as the target word, is the correct answer or solution to the question posed by the stem. Distractors are the incorrect answers provided alongside the key. These are crafted to challenge the examinee and create a level of uncertainty, testing the depth of their knowledge on the subject. For example, consider the following MCQ:

The most commonly used gas in light bulbs is

(A) Neon
(B) Argon
(C) Helium
(D) Oxygen

In this MCQ, the stem is the direct question, "The most commonly used gas in light bulbs is". The key, or the correct answer, is "Argon". The distractors, designed to test the knowledge of the examinee and potentially mislead those unsure of the correct answer, are the remaining options.

Research on automatic MCQ generation began over two decades ago, and since then, a significant amount of effort has been dedicated to its development [5]. For instance, the work in [10] presents a method for enhancing the quality of MCQ distractors through Automatic Item Generation. In [9], the authors propose a method for generating fill-in-the-blank questions with multiple choices from Thai text, using part-of-speech tagging and linear regression models to improve question and distractor quality. The work in [16] outlines an automated MCQ generation system using the BERT algorithm for text summarization and sentence mapping, along with WordNet for distractor generation. The authors in [19] introduce an NLP-based system for automatic MCQ generation for Computer-Based Testing Examination, utilizing keyword extraction from lesson materials to verify the effectiveness of the system in creating relevant exam questions. In [1], the authors propose an unsupervised dependency-based approach for extracting semantic relations in automatic MCQ generation, demonstrating high precision rates and positive user-centric evaluations in terms of readability, relevance, and overall usability for e-learning applications.

To the best of our knowledge, this paper is the first to jointly implement LLM with RAG and prompt engineering techniques for the automated generation of MCQs, targeting personalized learning. This unique combination leverages the strengths of each component to create a more efficient and contextually relevant question-generation process, ideally suited for blended learning and classroom flipping, adapting to diverse learning styles and educational dynamics.

III. METHOD

In this section, we introduce MCQGen, a comprehensive framework designed to automate the creation of personalized MCQs using advanced techniques in generative AI. The workflow initiates with instructors infusing the framework with domain-specific knowledge, contributing to a robust external knowledge base. This foundation enables a pre-trained LLM to engage in retrieval-augmented generation (RAG) [11], effectively synthesizing relevant information to construct questions that align with educational goals. In the subsequent phase, prompt engineering is strategically applied to guide the LLM in refining its question production, ensuring that the output meets the complexity required for effective learning assessments. Students then engage with these generated questions, and the insights gained from their performance and feedback guide instructors. This feedback serves as a cornerstone for the adaptive learning cycle of the framework, allowing for the continuous enhancement of the MCQs in response to student performance metrics. In doing so, MCQGen bridges the gap between the knowledge instructors wish to impart and the actual understanding students demonstrate, thereby supporting an adaptive and responsive educational experience. Figure 1 provides an overview of the MCQGen framework.

A. LARGE LANGUAGE MODEL

LLMs are transformative in the field of generative AI, setting new standards for machine understanding and generation of human language. These models excel in tasks ranging from text completion to complex problem-solving by leveraging vast amounts of data to predict and generate text sequences. LLMs can understand context, generate explanations, and even mimic human-like writing. In MCQGen, we particularly use the Generative Pre-trained Transformer 4 (GPT-4) [20] to facilitate the process of automated MCQ generation. GPT-4 is a state-of-the-art LLM that has been trained on an extensive corpus of textual data. Its design is grounded in deep learning algorithms that enable it to comprehend and produce human-like text across various subjects. In MCQGen, the extensive pretraining of GPT-4 allows it to understand and generate content that is both grammatically correct and contextually relevant, making it a powerful tool for creating educational materials. Its ability to process natural language inputs and generate accurate outputs is critical in the context of automated MCQ generation. The advanced text synthesis

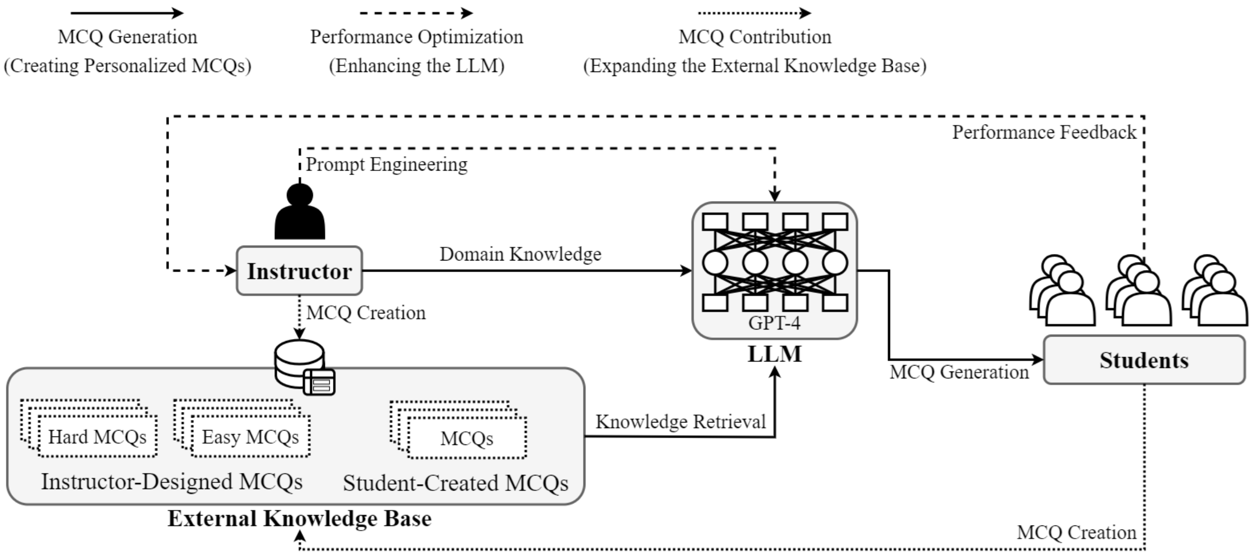


FIGURE 1. The architecture of MCQGen, illustrating the process from instructor input to personalized MCQ generation.

capabilities of GPT-4 are particularly useful for developing questions that not only test knowledge comprehension but also stimulate critical thinking, ensuring that the generated MCQs serve as effective learning tools.

The technical prowess of GPT-4 in MCQGen primarily addresses the challenge of generating MCQs that are semantically aligned with the specific domain topics provided by instructors. Most automated MCQ generation systems heavily depend on the ability of the learning model to understand and interpret the instructional input accurately to ensure the relevance of the generated questions. This reliance necessitates advanced text summarization capabilities, enabling the model to effectively search and identify key topic-related keywords within the inputs of the instructors. The design of GPT-4 incorporates artificial neural network architectures, particularly Transformer models, which excel in processing and understanding large sequences of text. This architecture allows GPT-4 to perform deep semantic analysis of the instructor-provided topics, ensuring that the generated questions are not only contextually relevant but also meaningful in relation to the specified content. The extensive pretraining of GPT-4 on diverse datasets equips it with a broad understanding of various subject matters, enhancing its ability to recognize and interpret a wide range of educational topics. The semantic processing capability of GPT-4 is crucial in maintaining the quality and relevance of the questions generated. It ensures that the MCQs produced are not just grammatically coherent but also accurately reflect the core ideas and themes of the instructional material. Therefore, the advanced semantic understanding and summarization skills of GPT-4 within MCQGen play a critical role in creating MCQs that are both educationally valuable and directly tied to the specific learning objectives set by the instructors.

Instructors can leverage the capabilities of GPT-4 within the MCQGen framework to create a diverse set of MCQs that align with the lecture topics of specific domains. The intelligent design of GPT-4 facilitates the generation of questions across a spectrum of difficulty levels, addressing the diverse educational needs of the classroom. By inputting domain knowledge into the framework, instructors can swiftly generate a comprehensive suite of questions. This approach significantly enhances the efficiency of creating assessments that are both rigorous and aligned with the learning objectives of the course. The benefits of GPT-4 extend to both teaching and learning experiences. Instructors gain from the reduced time and effort in assessment creation, allowing them to dedicate more resources to teaching and personalized instruction. The automated generation of questions also provides instructors with the tools to quickly adapt and develop quizzes that can accurately evaluate student learning, giving them the flexibility to respond to class performance trends and the individual needs of their students. For students, the MCQs created by GPT-4 offer a rich, adaptive learning experience. The questions generated by the model test a wide range of skills, from basic recall to higher-order thinking, providing students with a comprehensive assessment that supports their learning journey. As instructors adjust the difficulty and focus of the assessments based on class performance, students receive quizzes that are continually optimized to challenge their understanding and foster growth. This dynamic between GPT-4 generated content, instructor oversight, and student performance creates a synergistic cycle that enhances the educational process.

B. RETRIEVAL-AUGMENTED GENERATION

RAG [11] fundamentally enhances the MCQ generation process in the MCQGen framework by augmenting the capabilities of the LLM. The technical function of RAG lies in its ability to dynamically access and incorporate external data during the question generation process. When the LLM generates a question, RAG intervenes by querying a comprehensive database to retrieve relevant information that complements the context of the question. This process involves real-time data retrieval, where RAG selects and synthesizes relevant information based on the initial input and the ongoing generation context. The integration of this external data enables the generation of MCQs that are not only grammatically and contextually sound but also enriched with additional details and depth. RAG works in tandem with the LLM to ensure that each generated question is not just a product of internal knowledge of the model but a well-rounded item that reflects a broader understanding of the subject matter. This technical orchestration between RAG and the LLM thus results in MCQs that are more comprehensive, accurate, and reflective of the depth required in educational assessments.

Preprocessing is an important step in preparing the data for effective use with the LLM in our MCQGen framework. This process encompasses several key activities: text cleaning to remove irrelevant or extraneous information, tokenization to break down text into manageable pieces, encoding to convert text into a format suitable for machine processing, and addressing natural language artifacts to improve data coherence. Additionally, techniques like stemming and spelling correction are implemented to enhance data quality and consistency, ensuring it is well-suited for the fine-tuning of the LLM. Our external knowledge base, a critical component of the RAG system, is thoughtfully structured into two main categories: MCQs designed by instructors and those created by students. This combination plays a pivotal role in achieving the desired quality and diversity of the questions.

The instructor-designed MCQs are developed by educational professionals who are well-versed in the subject matter. Each question is carefully designed to engage students with scenarios that reinforce the curriculum, ensuring that each question directly supports the learning objectives of the course. These MCQs undergo rigorous quality checks and are classified into two levels of difficulty:

- Hard Level: These MCQs are designed to challenge students, testing their in-depth understanding of concepts and analytical abilities. They are often used to gauge mastery over complex topics and require a higher cognitive effort to solve.
- Easy Level: Easy questions aim to test fundamental knowledge and basic understanding of the subject matter. They are often used for introductory topics and to build foundational skills.

The categorization of MCQs into hard and easy levels mirrors Bloom's taxonomy, targeting higher-order and lower-order thinking skills, respectively, thus ensuring that the MCQs cater to various educational needs and allow for differentiated instruction and adaptive learning experiences.

TABLE 1. Sample MCQs created by the instructor and student for RAG.

Category	Question
Instructor (Hard Level)	Please determine whether the following is convex or concave: 1) $f(x) = \max_i x_i - \min_i x_i$; 2) $f(x, y) = \sqrt{\log(y - 42) - (y - 42 + x)}$. (A) (1) is convex, (2) is concave. (B) (1) is concave, (2) is convex. (C) Both (1) and (2) are convex. (D) Both (1) and (2) are concave.
Instructor (Easy Level)	Identify which of the following statements is true about descent methods. (A) The specific choice of search direction does not matter so much in descent methods. (B) The specific choice of line search does not matter so much in descent methods. (C) If we start the gradient descent method from a point near the solution, it will converge very quickly. (D) None.
Student	Which is the wrong opinion about the given function f and its corresponding conjugate function f^*? (A) The conjugate function f^* must be convex. (B) The conjugate function f^* is convex iff function f is convex. (C) If f is convex and close, the conjugate function of f^* (f^{**}) is f itself. (D) The conjugate function can be applied in the Lagrange duality problems.

In contrast, the student-created MCQs are derived from learners across different educational backgrounds and understanding levels. These questions are created without direct supervision or quality checks and are included in the database for several reasons:

- Diversity of perspectives: By incorporating questions created by students, the database gains a wide array of perspectives, reflecting the various ways in which learners interact with and understand the material.
- Creativity and novelty: Students often approach subjects with fresh insights and unique angles, and their questions can introduce unexpected challenges or viewpoints, enhancing the overall richness of the content.
- Real-world relevance: The inclusion of student-created MCQs allows for insights into what learners find intriguing or challenging about a topic, thus enabling the creation of MCQs that resonate more closely with the target audience.

It is important to acknowledge that student-created MCQs while offering diverse perspectives, lack a systematic quality assurance process. This necessitates a strategic integration of these questions with the instructor-designed ones. Balancing quality and novelty becomes essential to utilize this part of the database effectively. A systematic approach is employed where student-created MCQs are reviewed and selectively integrated with instructor-designed MCQs at a ratio that ensures quality is upheld (e.g., incorporating one student-created MCQ for every five instructor-designed MCQs). This method maintains a balance between diverse perspectives and rigorous educational standards. Table 1

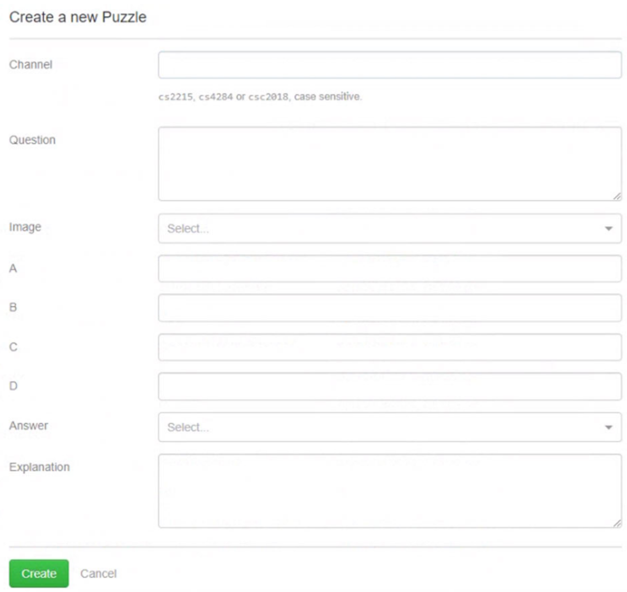


FIGURE 2. Interface of the platform used by students to create MCQs.

showcases MCQs created by both an instructor and a student for a graduate course on convex optimization and its applications in computer science.

To collect MCQs created by students, we develop a platform that allows students to create and submit their questions, detailed in Figure 2. Each MCQ submitted should include the three basic components of an MCQ: stem, key, and distractors, along with an explanation of why the chosen answer is correct. For organization, a "channel" entry pre-assigned by instructors categorizes the MCQs based on the course code. As part of the in-class assessment, students are tasked with creating MCQs related to the content covered in each lecture at the end of each tutorial. To motivate the creation of high-quality MCQs, students receive one point for a standard MCQ and two points for a good question, which teaching assistants manually verify. All student-created and instructor-designed MCQs are stored in the cloud database. In particular, our knowledge base contains 605 MCQs in total, consisting of 530 created by instructors and 75 by students.

The dual nature of the database, combining quality-controlled instructor-designed MCQs with the diversity of student-created content, enhances its value in the RAG system of MCQGen. This unique integration offers the potential to generate questions that are not only precise and adhere to educational standards but are also augmented with varied creative insights and perspectives. The meticulous organization of these questions into distinct difficulty levels adds another layer of refinement, enabling MCQGen to offer nuanced, personalized educational experiences. Such structuring optimizes the database for adaptive learning, ensuring that the content generated is both educationally sound and richly varied to suit diverse learning styles and needs.

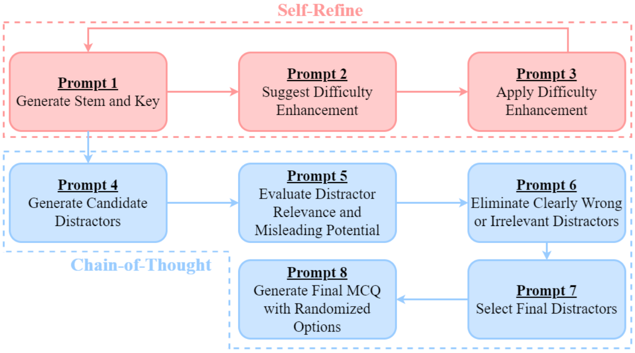

Self-Refine
Prompt 1
Generate Stem and Key
Prompt 2
Suggest Difficulty Enhancement
Prompt 3
Apply Difficulty Enhancement
Prompt 4
Generate Candidate Distractors
Prompt 5
Evaluate Distractor Relevance and Misleading Potential
Prompt 6
Eliminate Clearly Wrong or Irrelevant Distractors
Chain-of-Thought
Prompt 8
Generate Final MCQ with Randomized Options
Prompt 7
Select Final Distractors

FIGURE 3. The prompt engineering workflow, integrating chain-of-thought and self-refine prompting techniques for automated MCQ generation.

C. CHAIN-OF-THOUGHT WITH SELF-REFINE PROMPTING

Prompt engineering is a critical technique in optimizing the performance of LLMs for specific tasks such as MCQ generation. This process involves designing and structuring input prompts that guide the LLM towards desired outputs, effectively shaping its responses to align with specific objectives. In MCQGen, prompt engineering is employed to direct the LLM to produce questions that are not only relevant to the educational content but also structured in a way that challenges and engages students. By carefully crafting these prompts, the LLM is encouraged to apply its vast knowledge base in a focused manner, generating questions that are both educationally valuable and contextually appropriate. Optimization in prompt engineering is crucial for creating precise and clear instructions that guide LLMs accurately in the generation of MCQs, ensuring that their output aligns closely with the intended educational objectives. This technique ensures that the generative power of LLMs is harnessed effectively, resulting in the creation of MCQs that accurately reflect the subject matter and adhere to pedagogical standards. In MCQGen, we employ the chain-of-thought (CoT) with self-refine techniques to enhance the precision and depth of the output of the LLM. Figure 3 presents the prompt engineering workflow adopted in MCQGen, combining CoT and self-refine prompting methods to facilitate the process of automated MCQ generation.

The self-refine technique [15] in prompt engineering is a recursive loop designed to refine the content generated by the LLM. It begins with the LLM creating an initial stem and key for an MCQ. The process continues with the model critiquing its own output to suggest enhancements in difficulty without altering the correct response. This critique is crucial as it allows the model to evaluate and improve upon its initial attempt, informed by its own generated feedback. Meanwhile, this iterative critique provides valuable insights for the instructor, who can analyze the suggestions and make informed decisions to further refine the question, ensuring it meets the intended educational standards and learning objectives. Subsequent to this critique, the LLM integrates the suggested enhancements, updating the question to reflect

TABLE 2. **Prompt engineering with self-refine technique.**

Self-Refine Prompting (Prompts 1 – 3)
Prompt 1: Generate Stem and Key Generate a stem and key for a Multiple-Choice Question (MCQ) on the specified topic. Follow the format: Stem: [Insert Topic-Related Question] Key: [Insert Correct Answer]
Prompt 2: Suggest Difficulty Enhancement Given the MCQ stem and key, provide one suggestion to increase the difficulty without altering the correct answer. Question: [Stem] Correct Answer: [Key] Suggestion: [Provide suggestion to increase difficulty]
Prompt 3: Apply Difficulty Enhancement Integrate the difficulty enhancement suggestion into the MCQ. Ensure the stem reflects the increased difficulty. Original Question: [Stem]: [Key] Suggestion: [Suggestion] New Question: [Revised Stem]: [Key]

TABLE 3. **Prompt engineering with CoT technique.**

Chain-of-Thought Prompting (Prompts 4 – 8)
Prompt 4: Generate Candidate Distractors For the given MCQ stem and key, generate ten candidate distractors. Focus on common misunderstandings, errors, or related concepts. MCQ: [Stem]: [Key] Distractors: [Distractor 1], . . . , [Distractor 10]
Prompt 5: Evaluate Distractor Relevance and Misleading Potential Evaluate the relevance and potential to mislead for each distractor. Provide comments for each distractor. MCQ: [Stem]: [Key] Distractor Evaluations: [Distractor 1]: [Comment], . . . , [Distractor 10]: [Comment]
Prompt 6: Eliminate Clearly Wrong or Irrelevant Distractors From the evaluated distractors, eliminate those that are clearly wrong or irrelevant, retaining plausible ones. MCQ: [Stem]: [Key] Retained Distractors: [Distractor 1], . . .
Prompt 7: Select Final Distractors From the remaining distractors, select three that best serve as plausible yet incorrect options. MCQ: [Stem]: [Key] Final Distractors: [Distractor 1], [Distractor 2], [Distractor 3]
Prompt 8: Generate Final MCQ with Randomized Options Construct the final MCQ using the stem, key, and selected distractors. Randomize the order of all options. Format: [Stem] Options: A) [ ], B) [ ], C) [ ], D) [ ] Correct Answer: [Key]

a higher difficulty level. This iterative process ensures that each iteration of the question is more sophisticated than the last, challenging the understanding of students and ensuring the alignment of the stem with higher cognitive demand. By continuously looping through the generation and self-critique process, the self-refine prompting technique acts as an internal quality control mechanism, driving the LLM towards producing questions that meet educational standards and provide accurate measures of student comprehension. This method leverages the ability of the LLM to iterate over its creations, thus enabling a progressively refined question generation that is both dynamic and precise. In Table 2, we illustrate the self-refine prompting technique, showcasing a sequence of prompts that guide the LLM through its iterative refinement process in MCQ generation.

Following the self-refine step, we obtain an optimized stem and key for the MCQ, which meets the satisfaction of the instructor. Building upon this foundation, the CoT technique [14], [29], [30] is initiated to further refine the question-creation process. CoT prompting encourages the LLM to approach MCQ generation through a sequence of logical and analytical steps, mirroring human-like reasoning pathways. This method begins with the generation of candidate distractors, where the LLM proposes multiple plausible but incorrect answers, drawing on common misconceptions or errors related to the topic. The LLM then evaluates the relevance and potential of each distractor to mislead, providing justification for its choices, thereby engaging in a self-assessment akin to an educator reviewing possible exam answers. This generated evaluation concurrently assists instructors in understanding the rationale behind each considered distractor, equipping them with the knowledge to potentially eliminate certain options and estimate the overall difficulty of the question, thus allowing them to further tailor the assessment to the desired level of challenge. Distractors that do not meet the criteria based on the evaluation are discarded, ensuring that only the most suitable options remain. The LLM then selects the final distractors, which are plausible yet incorrect, to complete the MCQ. This selection is not random but is informed by the logical reasoning provided in the previous steps, ensuring that the final MCQ is challenging yet fair. Finally, the completed MCQ, with the stem, key, and selected distractors, is assembled with the options randomized to prevent answer pattern recognition. This entire CoT process allows the LLM to not only generate an MCQ but also to imbue it with a layer of cognitive reasoning, enhancing the educational value of the assessment. In Table 3, we illustrate the CoT prompting, which outlines a sequence of prompts guiding the LLM through a structured reasoning process in MCQ generation.

Combining CoT with self-refine techniques in prompt engineering provides significant benefits for personalized learning in MCQ generation. The joint approach leverages the strengths of both methods, where CoT enhances the logical depth and analytical rigor of the questions, while self-refine ensures iterative refinement for precision and relevance. Such integration yields MCQs that are not only contextually accurate and pedagogically rich but also tailored to individual learning levels, fostering deeper engagement and understanding. This methodology enables MCQGen to effectively generate a diverse set of questions that accommodate a wide spectrum of learners, making it a potent tool for personalized educational assessments.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the effectiveness of our MCQGen framework in creating MCQs, showcasing its application in facilitating personalized learning.

A. SETTING

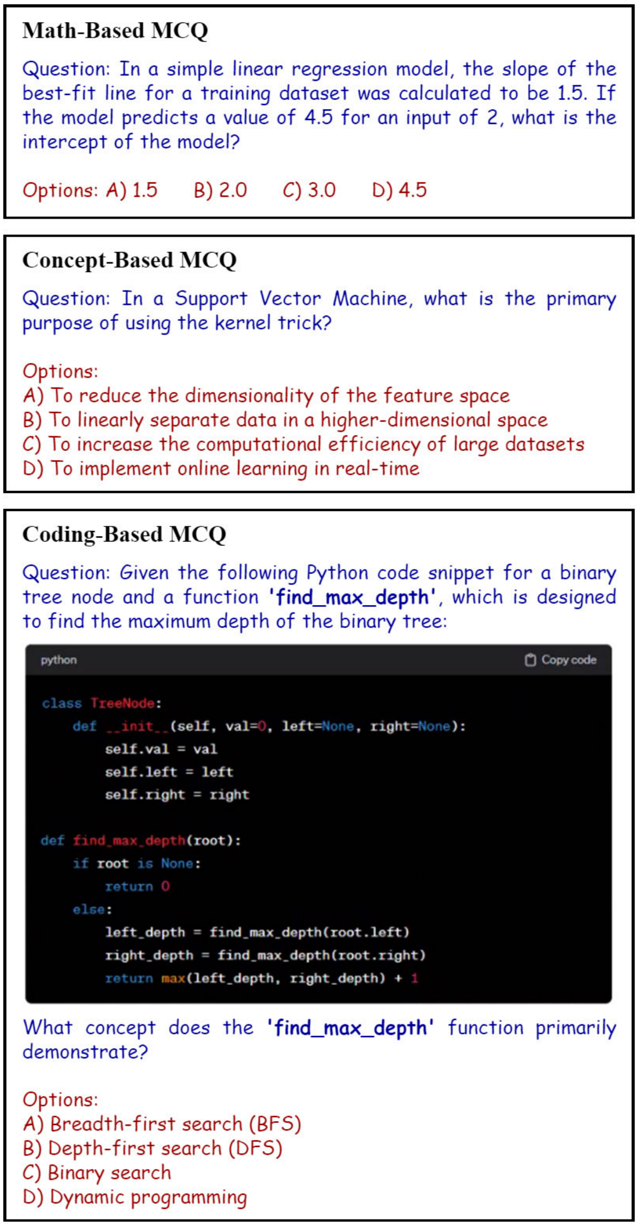
In our evaluation, the performance of MCQGen is assessed on three distinct types of questions: (1) math-based, involving numerical mathematics; (2) concept-based, focusing purely on textual content; and (3) coding-based, related to programming concepts. The chosen topic for these questions is computer science, tailored to the undergraduate level of understanding. To gauge the effectiveness and versatility of MCQGen, we generate a total of 60 questions, equally divided into 20 questions for each type. This approach allows us to comprehensively test the capability of the framework in handling diverse question formats within the specific academic discipline of computer science. By including questions that cover computational, theoretical, and programming knowledge, we can explore the proficiency of MCQGen across a broad spectrum of computer science education, thus presenting a wide-ranging challenge to the question generation abilities of MCQGen.

Following the methodologies proposed in [6], [24], and [18], we establish a set of criteria for the quality evaluation of the generated questions. These criteria, each to be scored on a scale from 1 to 5 with 1 being the worst and 5 being the best, include:

- Grammatical Fluidity: Ensuring each question is structurally sound and clear.
- Answerability: Verifying that the correct answer can be deduced from the provided context.
- Diversity: Emphasizing the generation of a wide range of question types from the same textual passage.
- Complexity: Demanding the creation of questions that require more than simple retrieval of facts.
- Relevance: Ensuring the questions are pertinent to the topic of computer science.

These criteria are designed to provide a comprehensive assessment of MCQGen, addressing both the linguistic and cognitive aspects of question generation.

For the evaluation of MCQGen, we employ qualitative analysis, integrating both human judgment and machine evaluation. The human evaluators comprise three university lecturers specializing in computer science, bringing expert insight into the academic validity of the generated questions. The team includes an experienced lecturer with over ten years in academia, a senior lecturer with a solid background of over four years, and a junior lecturer who has been teaching for under two years. On the machine side, we utilize advanced language models, including GPT-3.5 [4], [21], LLaMA-2 [28], and PaLM 2 [2], for an objective analysis. Notably, GPT-4 is excluded from the machine evaluators, as MCQGen utilizes GPT-4 as its core generation engine, making its inclusion a potential conflict of interest. This blend of human and machine evaluators ensures a comprehensive assessment, examining the outputs of MCQGen through the lens of expert educational insight and advanced computational analysis.



Math-Based MCQ

Question: In a simple linear regression model, the slope of the best-fit line for a training dataset was calculated to be 1.5. If the model predicts a value of 4.5 for an input of 2, what is the intercept of the model?

Options: A) 1.5 B) 2.0 C) 3.0 D) 4.5

Concept-Based MCQ

Question: In a Support Vector Machine, what is the primary purpose of using the kernel trick?

Options:
A) To reduce the dimensionality of the feature space
B) To linearly separate data in a higher-dimensional space
C) To increase the computational efficiency of large datasets
D) To implement online learning in real-time

Coding-Based MCQ

Question: Given the following Python code snippet for a binary tree node and a function 'find_max_depth', which is designed to find the maximum depth of the binary tree:

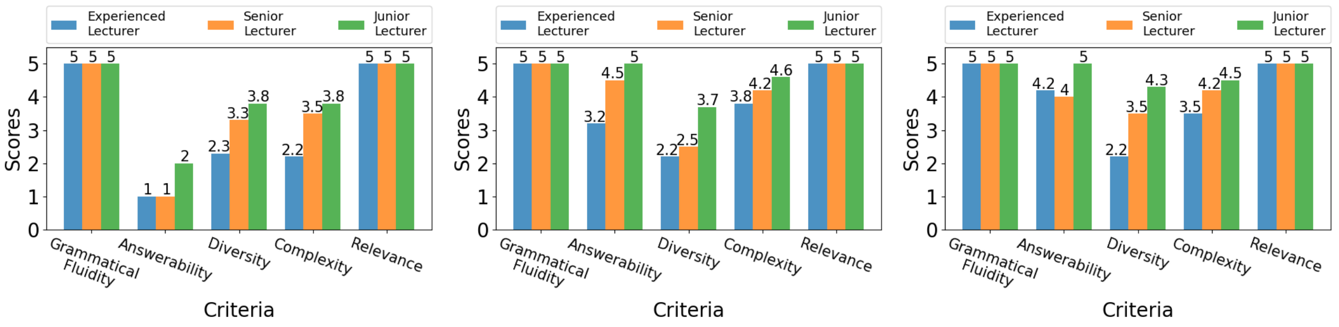
```python
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def find_max_depth(root):
    if root is None:
        return 0
    else:
        left_depth = find_max_depth(root.left)
        right_depth = find_max_depth(root.right)
        return max(left_depth, right_depth) + 1
```

What concept does the 'find_max_depth' function primarily demonstrate?

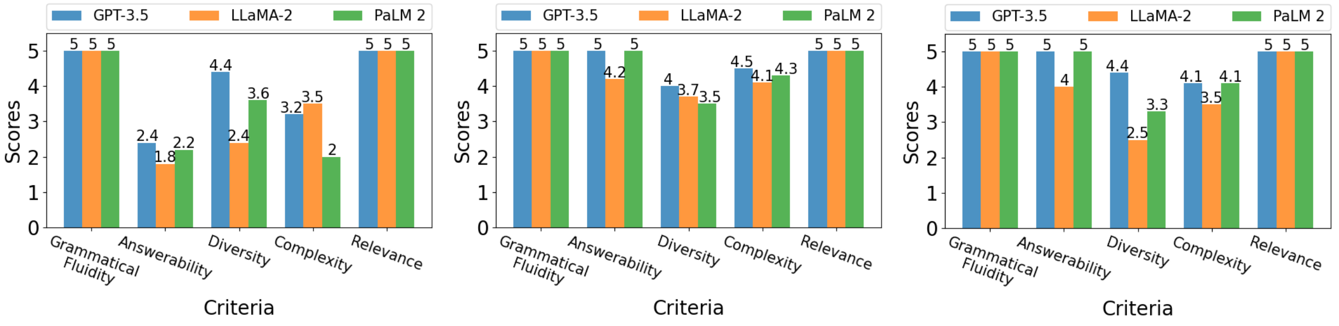
Options:
A) Breadth-first search (BFS)
B) Depth-first search (DFS)
C) Binary search
D) Dynamic programming

FIGURE 4. Sample MCQs produced by MCQGen, illustrating math-based, concept-based, and coding-based question types.



(a) Math-Based (b) Concept-Based (c) Coding-Based

FIGURE 5. Average evaluation scores for MCQGen by human evaluators across five criteria.



(a) Math-Based (b) Concept-Based (c) Coding-Based

FIGURE 6. Average evaluation scores for MCQGen by machine evaluators across five criteria.

B. RESULTS

In Figure 4, the capabilities of MCQGen are showcased with a sample of math-based, concept-based, and coding-based MCQs, reflecting the potential of the framework for crafting questions pertinent to an undergraduate computer science syllabus. Analyzing the evaluation scores depicted in Figure 5, a trend emerges across the five assessment criteria, with grammatical fluidity and relevance consistently receiving high scores, indicating the proficiency of MCQGen in generating questions that are linguistically clear and aligned with the subject matter. However, diversity and complexity show variability, suggesting room for enhancement in these areas. This is further corroborated by Figure 6, which illustrates the assessment of the performance of MCQGen by advanced language models. While the average scores for grammatical fluidity and relevance remain steadfast, the evaluations for diversity and complexity present opportunities for improvement, signaling the need for a broader spectrum of question types and deeper cognitive engagement in the questions generated by MCQGen.

The observed performance in answerability, particularly for math-based questions generated by MCQGen, reveals a potential area for improvement. Scores in this criterion are notably lower compared to those for concept-based and coding-based questions, suggesting that MCQGen might face challenges in crafting math-based questions where the options clearly contain the correct answer. Interestingly, evaluations from advanced language models yield slightly higher scores in answerability for math-based questions than those given by human evaluators. This discrepancy could be attributed to the interpretation of the language models, mistakenly considering certain unsolvable math-based questions (due to the absence of the correct answer among the options, rendering them unsolvable in an MCQ format) as potentially solvable using mathematical techniques like floor or ceiling. For instance, in a scenario where the precise answer is 2.4 with options being 1, 2, 3, 4, the models could deduce that the nearest correct answer is 2, thereby considering the question answerable. However, such an approach does not always align with the actual intent or accuracy required in educational settings, highlighting a limitation in the evaluation capabilities of these language models.

V. EDUCATIONAL IMPACT AND STUDENT FEEDBACK ANALYSIS

In this section, we examine the application of MCQGen to enhance student learning experiences in an introductory computer science course. The setting involves two distinct types of first-year undergraduate students: one demonstrating strong academic performance and the other facing challenges in grasping course concepts. This allows us to evaluate the capacity of MCQGen to tailor assessments to diverse learning needs, ensuring that questions are accessible yet

TABLE 4. Feedback scores from students on generated MCQs.

Student	Question (1)	Question (2)
Student A	3	4
Student B	2	5

challenging for students at varying levels of understanding. The instructor employs MCQGen to adjust the difficulty of MCQs issued after each of the 12 lectures covering various computer science topics. Initially, both students receive the same MCQ; subsequent questions are then tailored based on their responses. For correct answers, the following question may increase in difficulty to challenge the student further unless it is deemed already at a peak difficulty level. Conversely, an incorrect answer leads to a simplified question or maintains the difficulty if the initial question is at the baseline of complexity. This adaptive approach dynamically adjusts the difficulty of questions based on student performance, promoting a personalized learning experience that encourages progression while ensuring concepts are reinforced at an appropriate pace for the understanding level of each student. To guarantee the answerability of each MCQ, the instructor carefully verifies that the correct answer is indeed among the options provided, thus ensuring the questions are solvable.

After completing all the MCQs, the two students are asked the following two questions:

1) How would you rate the difficulty level of the MCQs generated by MCQGen in enhancing your understanding of the course material? (Scale of 1 – 5, with 1 being much too easy and 5 being much too difficult)
2) To what extent do you feel that the MCQs generated by MCQGen contributed to improving your overall learning experience in this course? (Scale of 1 – 5, with 1 being not at all and 5 being significantly)

Upon reflection of the feedback from two first-year undergraduate students on the MCQs generated by MCQGen, a clear pattern emerges in their perception of the difficulty level and the contribution of these questions to their learning experience (see Table 4). Student A, with a stronger academic foundation, found the difficulty level of the questions to be well-balanced, rating it as moderate. This indicates that the questions were challenging enough to stimulate deeper learning without being overwhelmingly difficult. The feedback suggests that MCQGen successfully identified and matched the skill level of the student, thus enhancing their understanding of the material. On the other hand, Student B, who initially struggled more with the course content, perceived the questions as slightly easy, reflecting the ability of MCQGen to adapt and provide accessible challenges that cater to their learning pace. Notably, both students acknowledged a significant improvement in their overall learning experience, with Student B experiencing a profound impact, highlighting the role of MCQGen in making the course material more engaging and comprehensible. This feedback highlights the effectiveness of MCQGen in creating an inclusive learning environment that accommodates the diverse needs of students, promoting advancement, and reinforcing course concepts at a pace that aligns with the comprehension level of each learner.

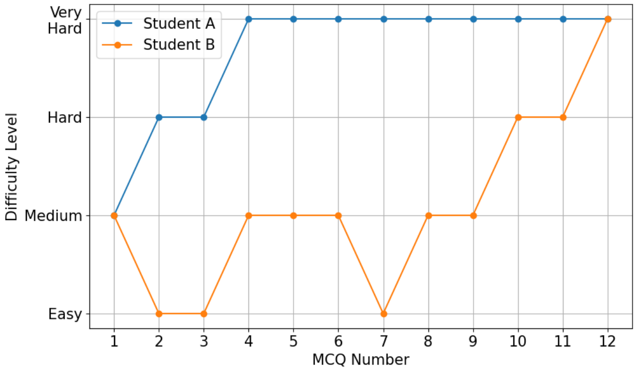


FIGURE 7. The learning performance of students on the generated MCQs with adaptive difficulty.

The observed performance of students on the generated MCQs with adaptive difficulty, as illustrated in Figure 7, provides compelling insights into the effectiveness of MCQGen in facilitating personalized learning. Performance-based difficulty adjustment is evident, with the framework dynamically modifying the complexity of subsequent questions based on previous answers of each student. For instance, the initial difficulties of Student B trigger an adjustment towards easier questions, facilitating a gradual build-up of confidence and knowledge before escalating to more challenging material. This scaffolding approach underscores the capability of MCQGen to provide a progressive learning experience, as demonstrated by the eventual engagement with very hard questions of both students. The trajectory of Student A, who consistently performs well, showcases how adept learners are continuously challenged, ensuring their engagement and growth. Conversely, the progression of Student B from struggling with basic concepts to successfully tackling very hard questions illustrates the flexibility of MCQGen in catering to diverse educational needs and learning speeds. Such personalized engagement fosters a deeper, more meaningful learning experience, allowing students to navigate the curriculum at a pace that matches their evolving comprehension levels. This adaptive strategy not only aligns with the principles of effective pedagogy but also highlights the potential of technology-enhanced learning tools like MCQGen to revolutionize educational practices by promoting individualized learning paths that can significantly improve educational outcomes.

VI. DISCUSSION

The outcomes of our study indicate that the MCQGen framework is successful in improving both the learning experiences and performance of students. Given these findings, it becomes critical to explore the underlying factors contributing to such positive results and examine the broader implications for educational technology.

The high performance in grammatical fluidity and relevance is likely due to the advanced training of GPT-4 on a diverse textual corpus, equipping it with the ability to construct sentences that are both syntactically correct and semantically precise. This underlying strength of GPT-4 ensures that the MCQs generated are clear and contextually pertinent, resonating with the core material of the subject. However, the observed limitations in question diversity and complexity are partially attributed to the inherent challenges of automated MCQ generation. The prompt engineering process, while directing the focus of MCQGen, may inadvertently limit the scope of question variability and depth. The external knowledge base might also not encompass the full breadth of potential subtopics or the nuanced complexities of certain concepts, leading to questions that, while correct, do not fully engage the higher-order thinking skills required for deeper learning. These aspects suggest potential refinements in MCQGen, such as expanding the knowledge base and optimizing the prompt engineering process to encourage a greater range of question styles and higher levels of difficulty.

The challenges in generating math-based MCQs primarily arise from the mathematical reasoning capabilities of GPT-4, which are less advanced, especially in numerical calculations. It is observed that while the question can be logically solvable, the absence of the correct answer among the options provided hinders its solvability. This issue is a common limitation in LLMs, which are primarily designed for text processing rather than for handling mathematical content. Addressing this requires additional strategies, such as employing CoT prompting to guide GPT-4 towards generating correct answers. Consequently, a thorough evaluation of all options, including the correct answer and not just the distractors, becomes crucial. This necessitates a collaborative effort involving both human (instructor) and machine (MCQGen) evaluators, highlighting the importance of human-in-the-loop in enhancing the quality and accuracy of the generated MCQs, thus ensuring they are both challenging and solvable. For instance, we can fine-tune GPT-generated questions to validate the correctness of convex optimization problems using *Disciplined Convex Programming* (details available at https://fenchel.stanford.edu), which constructs mathematical expressions with known curvature from a given library of base functions, ensuring rigor and precision in GPT-driven optimization question generation.

MCQs serve as a ‘‘low-stakes’’ assessment method, effectively reducing the pressure on students while offering educators a flexible tool for gauging understanding across a wide range of topics. The nature of MCQs, particularly when generated by MCQGen, supports personalized learning by allowing for the adjustment of difficulty levels and thematic focus based on individual student performance. This adaptability ensures that each learner is challenged appropriately, promoting engagement and facilitating deeper comprehension of the subject matter. Although students reported that the average difficulty of the generated MCQs is not exceedingly high, we can refine the capabilities of MCQGen by adjusting the MCQ format to allow for multiple correct options. For instance, by formatting questions where options include composite answers such as ‘‘Options A and C are correct’’, ‘‘Only Option B is correct’’, or ‘‘All of the above’’, we can introduce a more complex decision-making process for the students. Additionally, the ability of MCQGen to produce a large quantity of diverse and relevant MCQs aligns with the goals of learning at scale, making it possible to deliver quality education to a broad audience. The automated generation of MCQs by frameworks like MCQGen introduces efficiency and scalability into the assessment process, enabling educators to implement a more differentiated and responsive teaching approach. This fusion of personalized assessment with the capacity for widespread application highlights the transformative potential of MCQGen in both classroom and large-scale educational settings, offering a bridge between individualized learning experiences and the demands of educational expansion.

While MCQGen has proven effective in enhancing learning experiences and performance, it is crucial to align online assessments with the cognitive learning styles of students, as different studies have shown that educational outcomes improve when assessments and interventions match individual cognitive preferences. MCQGen can be enhanced to consider various cognitive learning styles by incorporating adaptive algorithms to address this. These algorithms could tailor questions not only to the performance level of a student but also to their preferred learning methods. For example, visual learners might benefit from questions that include diagrams or visual aids, whereas verbal learners could engage more with text-based questions. By analyzing data on student preferences and performance, MCQGen can dynamically adjust the format and complexity of MCQs, ensuring each student receives assessments that are both challenging and suited to their cognitive style. This enhancement would increase the relevance and engagement of the MCQs, fostering deeper and more effective learning. Integrating cognitive learning styles into MCQGen would significantly advance personalized learning and educational technology.

VII. LIMITATIONS AND FUTURE WORK

While our study sheds light on the effectiveness of the MCQGen framework in enhancing personalized learning experiences, several limitations are noteworthy:

- The absence of extensive numerical evaluations may limit our ability to provide more convincing quantitative evidence of the impact of MCQGen.
- The relatively small sample size for both performance evaluation and student feedback may not fully represent broader user experiences, suggesting the need for a larger cohort of evaluators and feedback from more students.

- The evaluation is confined to the domain of computer science, which may not capture the applicability of MCQGen across different subjects.
- The exclusive use of GPT-4 in this study without comparing other models may limit insights into its relative effectiveness.
- Focusing solely on English MCQs may overlook the challenges of multilingual question generation and cultural differences.

Despite the limitations, our study contributes valuable insights into the potential of automated MCQ generation to support personalized learning and represents a meaningful step forward in the application of generative AI in education. Future work will aim to address the current limitations of the study by expanding the scope of the evaluation, incorporating a wider variety of subjects, and exploring the integration of multiple LLMs and languages to enhance the diversity and complexity of the generated content. Automatic generation of MCQs for mathematical topics such as convex optimization can greatly benefit from formal methods and analysis, particularly the Disciplined Convex Programming framework. This approach can be further extended to design questions related to optimization algorithms. Another possible direction is to apply reinforcement learning techniques that incorporate student feedback directly into the LLM training process, further refining and personalizing the generation of MCQs to better meet educational objectives and student needs.

VIII. CONCLUSION

In this paper, we introduce MCQGen, a novel framework harnessing the capabilities of a large language model to generate multiple-choice questions (MCQs) for personalized learning. By integrating retrieval-augmented generation and prompt engineering techniques, MCQGen effectively produces questions that are both relevant and challenging, enhancing the learning experience. Our evaluations, combining human expertise and advanced computational analysis, demonstrate the effectiveness of the framework in creating diverse, complex, and contextually appropriate questions. MCQGen marks a significant step forward in educational technology, offering an efficient solution for generating high-quality MCQs, which is crucial in the fields of e-learning and digital assessment. Looking ahead, further exploration into expanding the capabilities of MCQGen across various disciplines and educational levels presents a promising avenue with the potential to revolutionize personalized learning and assessment methodologies in the digital age.

ACKNOWLEDGMENT

The authors are grateful for helpful discussions on the subject with X. Wang, D. Joyner, D.-M. Chiu, and G. M. Voelker.

REFERENCES

[1] N. Afzal and R. Mitkov, “Automatic generation of multiple choice questions using dependency-based semantic relations,” *Soft Comput.*, vol. 18, no. 7, pp. 1269–1281, Jul. 2014.

[2] R. Anil, “PaLM 2 technical report,” 2023, *arXiv:2305.10403*.

[3] A.-M. Brady, “Assessment of learning with multiple-choice questions,” *Nurse Educ. Pract.*, vol. 5, no. 4, pp. 238–242, Jul. 2005.

[4] T. B. Brown, “Language models are few-shot learners,” in *Proc. NIPS*, 2020, pp. 1877–1901.

[5] D. R. Ch and S. K. Saha, “Automatic multiple choice question generation from text: A survey,” *IEEE Trans. Learn. Technol.*, vol. 13, no. 1, pp. 14–25, Jan. 2020.

[6] Y. Chali and S. A. Hasan, “Towards topic-to-question generation,” *Comput. Linguistics*, vol. 41, no. 1, pp. 1–20, Mar. 2015.

[7] C. J. Bonk and C. R. Graham, *The Handbook of Blended Learning: Global Perspectives, Local Designs*. San Francisco, CA, USA: Wiley, 2012.

[8] T. M. Haladyna, S. M. Downing, and M. C. Rodriguez, “A review of multiple-choice item-writing guidelines for classroom assessment,” *Appl. Meas. Educ.*, vol. 15, no. 3, pp. 309–333, Jul. 2002.

[9] C. Kwankajornkiet, A. Suchato, and P. Punyabukkana, “Automatic multiple-choice question generation from Thai text,” in *Proc. 13th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2016, pp. 1–6.

[10] H. Lai, M. J. Gierl, C. Touchie, D. Pugh, A.-P. Boulais, and A. De Champlain, “Using automatic item generation to improve the quality of MCQ distractors,” *Teaching Learn. Med.*, vol. 28, no. 2, pp. 166–173, Apr. 2016.

[11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, and T. Rocktäschel, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9459–9474.

[12] J. Li, L. Ling, and C. W. Tan, “Blending peer instruction with just-in-time teaching: Jointly optimal task scheduling with feedback for classroom flipping,” in *Proc. 8th ACM Conf. Learn. Scale*, 2021, pp. 117–126.

[13] J. Li, C. W. Tan, C. N. Hang, and X. Qi, “A chatbot-server framework for scalable machine learning education through crowdsourced data,” in *Proc. 9th ACM Conf. Learn. Scale*, Jun. 2022, pp. 271–274.

[14] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch, “Faithful chain-of-thought reasoning,” 2023, *arXiv:2301.13379*.

[15] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. Prasad Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, “Self-refine: Iterative refinement with self-feedback,” 2023, *arXiv:2303.17651*.

[16] P. K. Mehta, P. Jain, C. Makwana, and C. Raut, “Automated MCQ generator using natural language processing,” *Internation Res. J. Eng. Technol.*, vol. 8, pp. 2705–2710, May 2021.

[17] K. Missildine, R. Fountain, L. Summers, and K. Gosselin, “Flipping the classroom to improve student performance and satisfaction,” *J. Nursing Educ.*, vol. 52, no. 10, pp. 597–599, Oct. 2013.

[18] P. Nema and M. M. Khapra, “Towards a better metric for evaluating question generation systems,” 2018, *arXiv:1808.10192*.

[19] C. A. Nwafor and I. E. Onyenwe, “An automated multiple-choice question generation using natural language processing techniques,” 2021, *arXiv:2103.14757*.

[20] J. Achiam, “GPT-4 technical report,” 2023, *arXiv:2303.08774*.

[21] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, “Training language models to follow instructions with human feedback,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2022, pp. 27730–27744.

[22] E. J. Palmer and P. G. Devitt, “Assessment of higher order cognitive skills in undergraduate education: Modified essay or multiple choice questions? Research paper,” *BMC Med. Educ.*, vol. 7, no. 1, pp. 1–7, Dec. 2007.

[23] J. Park, “Constructive multiple-choice testing system,” *Brit. J. Educ. Technol.*, vol. 41, no. 6, pp. 1054–1064, Nov. 2010.

[24] V. Raina and M. Gales, “Multiple-choice question generation: Towards an automated assessment framework,” 2022, *arXiv:2209.11830*.

[25] C. W. Tan, “The value of cooperation: From AIMD to flipped classroom teaching,” in *Proc. 1st Int. Workshop Teaching Perform. Anal. Comput. Syst.*, 2021.

[26] C. W. Tan, “Large language model-driven classroom flipping: Empowering student-centric peer questioning with flipped interaction,” 2023, *arXiv:2311.14708*.

[27] C. W. Tan, L. Ling, P.-D. Yu, C. N. Hang, and M. F. Wong, “Mathematics gamification in mobile app software for personalized learning at scale,” in *Proc. IEEE Integr. STEM Educ. Conf. (ISEC)*, Aug. 2020, pp. 1–5.

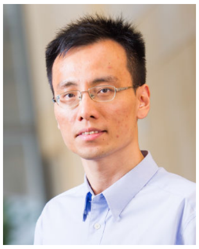
[28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[29] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2022, *arXiv:2203.11171*.

[30] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 24824–24837.

[31] M.-F. Wong, S. Guo, C.-N. Hang, S.-W. Ho, and C.-W. Tan, "Natural language generation and understanding of big code for AI-assisted programming: A review," *Entropy*, vol. 25, no. 6, p. 888, Jun. 2023.

CHING NAM HANG (Member, IEEE) received the B.S. (First Class Honors) and Ph.D. degrees in computer science from the City University of Hong Kong, Hong Kong. He is currently an Assistant Professor with the Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong. His research interests include data science, network science, and AI for Health-Tech and Ed-Tech.



CHEE WEI TAN (Senior Member, IEEE) received the M.A. and Ph.D. degrees in electrical engineering from Princeton University. He is currently an Associate Professor of computer science and engineering with Nanyang Technological University. His research interests include networks, distributed optimization, and generative AI. He served as an Editor for IEEE Transactions on Cognitive Communications and Networking, IEEE/ACM Transactions on Networking, and IEEE Transactions on Communications, and as a Distinguished Lecturer of IEEE Communications Society.

PEI-DUO YU received the B.Sc. and M.Sc. degrees in applied mathematics from National Chiao Tung University, Taiwan, in 2011 and 2014, respectively, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong. He worked as an Assistant Professor with the College of Electrical Engineering and Computer Science, Chung Yuan Christian University, Taiwan. Currently, he is an Assistant Professor with the Department of Applied Mathematics. His research interests include combinatorics counting, graph algorithms, optimization theory, and its applications.