Automatic Generation of Multiple-Choice Questions for CS0 and CS1 Curricula Using Large Language Models

Tian Song(✉), Qinqin Tian, Yijia Xiao, and Shuting Liu

Beijing Institute of Technology, Beijing 100081, China
songtian@bit.edu.cn

Abstract. In the context of increasing attention to formative assessment in universities, Multiple Choice Question (MCQ) has become a vital assessment form for CS0 and CS1 courses due to its advantages of rapid assessment, which has brought about a significant demand for MCQ exercises. However, creating many MCQs takes time and effort for teachers. A practical method is to use large language models (LLMs) to generate MCQs automatically, but when dealing with specific domain problems, the model results may need to be more reliable. This article designs a set of prompt chains to improve the performance of LLM in education. Based on this design, we developed EduCS, which is based on GPT-3.5 and can automatically generate complete MCQs according to the CS0/CS1 course outline. To evaluate the quality of MCQs generated by EduCS, we established a set of evaluation metrics from four aspects about the three components of MCQ and the complete MCQ, and based on this, we utilized expert scoring. The experimental results indicate that while the generated questions require teacher verification before being delivered to students, they show great potential in terms of quality. The EduCS system demonstrates the ability to generate complete MCQs that can complement formative and summative assessments for students at different levels. The EduCS has great promise value in the formative assessment of CS education.

Keywords: Large Language Models · GPT-3.5 · MCQs · automatic generation · CS education

1 Introduction

CS0 and CS1 are widely offered courses in computer science majors in universities, and teaching and evaluation are inseparable. MCQs have become the most common forms of assessment due to their advantages of rapid evaluation. Nowadays, universities are increasingly focusing on formative evaluation, which has brought about a significant demand for MCQs. However, manual preparation of MCQs is time-consuming and expensive. All these have prompted researchers and educators to develop a multiple-choice question bank, but outdated question banks may soon not match teachers' curriculum goals. We have seen that

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Hong and G. Kanaparan (Eds.): ICCSE 2023, CCIS 2023, pp. 314–324, 2024.
https://doi.org/10.1007/978-981-97-0730-0_28

efficiently generating MCQs for CS0 and CS1 courses has become a significant development demand in CS education.

The latest progress in artificial intelligence has led to the emergence of large language models (LLMs) with excellent performance. These models have shown enormous application potential in education, medical diagnosis, social media management, and public opinion analysis, especially in computer education. For example, they have demonstrated near-human level in programming exercises [1], code generation [2], and code interpretation [3]. However, LLMs require domain-specific knowledge, and results cannot guarantee accuracy. Can we use them to generate high-quality MCQ for CS0/CS1 courses, thereby greatly supplementing formative and summative tests for different student levels? Regarding this, some scholars have initially attempted the possibility of using GPT to create multiple-choice questions for CS courses [4]. However, the existing problems include: 1) Automation still needs to be implemented; 2) Unable to generate a complete MCQ. Just extract the stem from the current question bank and generate distractors and correct answers based on the existing stem;3) There is no discussion on effectively applying LLMs in CS education.

While automating the generation of complete multiple-choice questions for CS0/CS1 courses, this study focuses on how to make LLMs perform well in CS education. The main contributions include:

1) We proposed an EduCS based on GPT-3.5, which can automatically generate multiple-choice questions based on the CS0/CS1 curriculum outline, which has a significant potential impact on the education field and can significantly reduce the workload of formative evaluation in the teaching process.
2) Design a set of prompt chains to enable LLMs to perform well in CS education.
3) Introducing user input to match the learning level of learners can generate different difficulty levels of MCQs for students at different levels, promoting personalized teaching.

2 Related Work

2.1 Multiple Choice Questions Generation

The generation of MCQ includes three parts: Stem Generation (SG), Answer Generation (AG), and Distractors Generation (DG). The paradigms of these studies have gone through rule-based [5], neural networks [6], transformers [7], and now large language models. Macaw [8] can execute SG, AG, and DG step by step. Dijkstra [9] et al. proposed an end-to-end quiz generator, EduQuiz, using GPT-3, which can generate complete multiple-choice questions and answers for reading comprehension tasks. Gabajiwala [10] et al. generated different types of questions based on keywords extracted by NLP. Experiments showed that about 60% of the questions generated by this model could not be correctly identified by survey participants. Andrew Tran and others used GPT3 and GPT4 to generate complete isomorphic multiple-choice questions using manual human-computer interaction. As far as we know, there is no work before this work that can be a one-stop solution, that is, simultaneously associate SG, AG, and DG to generate complete multiple-choice questions and simultaneously be dedicated to the research of automatic generation of high-quality questions.

2.2 Quality Evaluation Metrics

Most quality assessments of automatically generated multiple-choice questions typically rely on human evaluators. There is currently no gold metric that is applicable to the general field. MCQ consists of three components, and researchers have defined different metrics for assessing the quality of individual components. Literature [11] summarizes the existing evaluation metrics.

Evaluation of the Stem and Answer. Common ones include sentence format, sentence length, sentence simplicity, difficulty, usefulness for learning, answerability, sufficient context, question difficulty, field-related, too much information, insufficient information, correctness, etc.

Evaluation of the Distractors. The methods and criteria used to evaluate the quality of distractors (interference items) in different research works are: Mitkov et al. [12]. Proposed to use difficulty, discrimination and usefulness to evaluate the quality of distractors. To determine whether distractors were relevant and grammatical to the original text, Pino, Heilman, and Eskenazi [13] used grammatical and semantic criteria to measure the grammaticality and collocation of sentences. Readability measure, to assess whether distractors affect the clarity of a question, Agarwal and Mannem [14] asked evaluators to replace the distractors in the white space to check readability and semantics in sentences. To help determine whether a distraction is appropriate, Bhatia, Kirti, and Saha use intimacy values to evaluate distractions. If the distractor is close to the answer, it can be considered "good". Araki et al. [15] proposed a three-point scale to measure the quality of distractors. This scale evaluates the quality of distractors based on whether they are confusing and easily identifiable as incorrect answers.

2.3 Prompting Engineering

LLMS are very few learners who are able to answer the questions without additional fine-tuning. Finding the best tips for a specific task is a challenge. Some research focuses on hint engineering used in LLMs [16]. In the field of programming education, Reeves et al. [17] found that small changes in prompts have a significant impact on Codex performance, and the impact varies across different types of questions. Denny et al. [2] explored how to improve Copilot's performance in CS1 exercises through hints. Research shows that hint engineering can significantly improve Copilot's performance on CS1 problems. However, research has also found that too lengthy and detailed prompts may lead to reduced model performance. In the study of MCQs generation, Andrew Tran et al. summarized three effective tips through manual interaction with the GPT interactive interface. Since LLMs do not always produce optimal results on the first try, some research considers multi-round hint engineering to improve the output. Some studies adopt supervised methods to solve this problem [18]. However, they may require large amounts of training data or expensive human annotation, so some studies have designed improved methods without requiring extensive supervision. Shi et al. proposed a three-round iterative prompt including task instructions, keyword constraint prompts and length constraint prompts for food effect research summary in the biomedical field.

3 Methodology

3.1 Multiple Choice Questions

MCQ consists of three components: Stem, Correct Answer, and Distractors. The question stem is the central part of the multiple-choice question, which presents the problem or situation to be solved. Correct answer matches the question or situation in the stem and is the only correct option. Distractors are incorrect answer options that are designed to lead students into making mistakes or confuse them. They may look similar to the correct answer but are wrong. The purpose of distractors is to test students' discrimination skills to ensure they can identify the correct answer and avoid confusion. The example is as follows:

The programming language that can be directly executed by computer is ().

A. C language
B. BASIC language
C. assembly language
D. machine language

The stem of this MCQ is "The programming language that can be directly executed by a computer is?". It could be presented in question form: "What programming language can be directly executed by a computer?". Four choices are given in the question. Among these, the correct answer is 'machine language'. The rest three choices are the distractors.

3.2 The Framework of EduCS

The workflow of EduCS is shown in Figure 1. It mainly consists of three parts: input, generation, and output. Our research focuses on the first two parts, and we will introduce the main content in the first two parts below.

We know that LLMs need to improve in solving domain-specific problems, and research shows that prompt engineering can be practical for this problem. A prompt is a set of instructions provided to an LLM that programs the LLM by customizing it and enhancing or refining its capabilities. It is essential to mulate effective prompts that drive informative conversation. In general, finding the best prompt for a specific task is challenging. To improve the quality of MCQs, we focused on designing prompt chains.

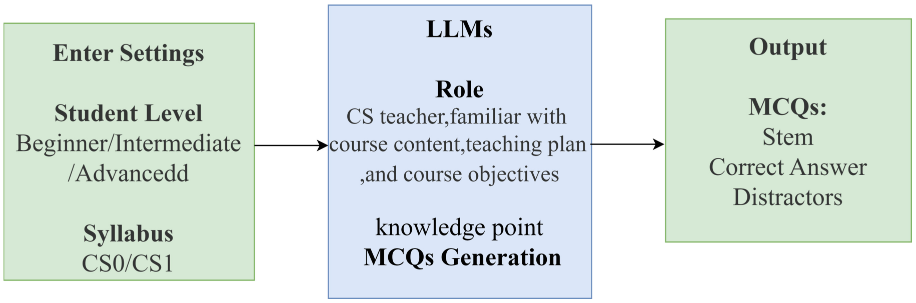


Fig. 1. The workflow of EduCS.

System User Input. CS0/CS1 syllabus, student level (beginner, intermediate, advanced). On the one hand, the input of the syllabus tells LLMs course information, and on the other hand, it is utilized by CS teachers to select knowledge points. The student-level option facilitates users to generate MCQs that match students' knowledge state.

Self-review. We introduce self-review that requires LLM to improve its previously generated results. Self-review can guide LLM self-correction. We feed the MCQs and reviews generated in the previous round into LLM and tell it to use these to regenerate the MCQs. Self-review design is crucial as it enables LLMS to look more closely at the generated MCQs from multiple perspectives.

Role-Playing. Telling LLMs its role in the dialogue, is used to guide LLMs to complete the responsibilities of the corresponding position. As we all know, in actual teaching environment scenarios, teachers of different subjects have different professional knowledge. Inspired by this situation, EduCS tells LLMs to play the role of CS teacher in the dialogue and what ability this role has, which is used to guide LLMs to complete the responsibilities of its position. Role-playing can limit the generated MCQs to a certain extent related to specific topics or fields. Making it professional helps improve quality.

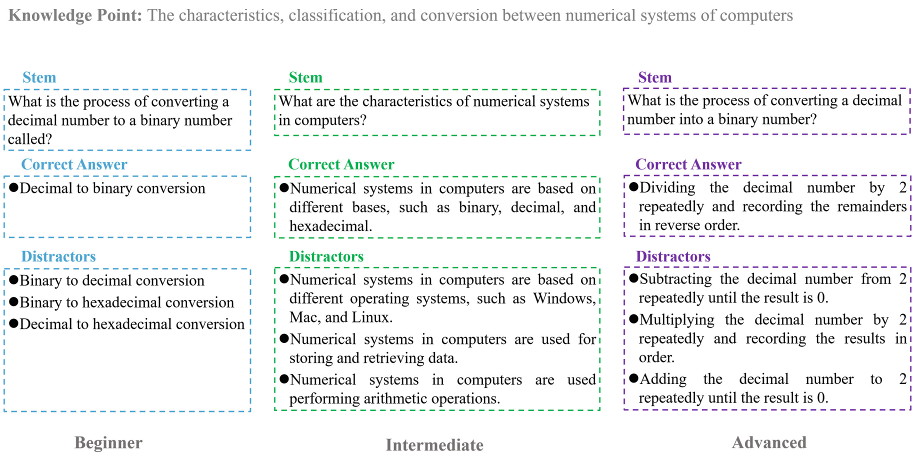
4 Experimental Evaluation

In this section, our objectives revolve around validating the efficacy of EduCS through a series of experiments conducted using GPT-3.5. Firstly, we input various learner proficiency levels to assess whether the difficulty of the generated MCQs aligns with expectations. Secondly, We investigated whether EduCS has the ability to generate different types of MCQs. Lastly, we evaluate the quality of MCQs based on the proposed evaluation metrics.

4.1 Student Level

Study 1. We entered three student proficiency levels sequentially into EduCS: beginner, intermediate, and advanced. We evaluate whether the generated questions can adapt to different students by observing their difficulty levels.

Results. Figure 2 shows an example of MCQs generated by EduCS based on three different student levels for the same knowledge point. For MCQs for junior students, Correct Answers are almost included in the stem, which can help beginners build basic knowledge. Questions for intermediate students require a more careful comparison of options to choose. Encourage students to expand their breadth of knowledge, which can help students build self-confidence. Questions for advanced students require students to deepen their understanding of knowledge. It can be seen that EduCS can generate MCQs of different difficulty levels according to the student level. It can promote personalized learning and is of great significance.


Knowledge Point: The characteristics, classification, and conversion between numerical systems of computers

Beginner

Stem
What is the process of converting a decimal number to a binary number called?

Correct Answer
- Decimal to binary conversion

Distractors
- Binary to decimal conversion
- Binary to hexadecimal conversion
- Decimal to hexadecimal conversion

Intermediate

Stem
What are the characteristics of numerical systems in computers?

Correct Answer
- Numerical systems in computers are based on different bases, such as binary, decimal, and hexadecimal.

Distractors
- Numerical systems in computers are based on different operating systems, such as Windows, Mac, and Linux.
- Numerical systems in computers are used for storing and retrieving data.
- Numerical systems in computers are used performing arithmetic operations.

Advanced

Stem
What is the process of converting a decimal number into a binary number?

Correct Answer
- Dividing the decimal number by 2 repeatedly and recording the remainders in reverse order.

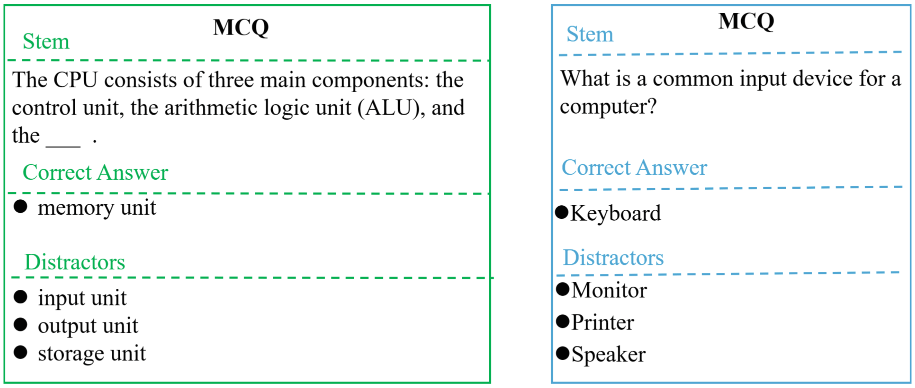
Distractors
- Subtracting the decimal number from 2 repeatedly until the result is 0.
- Multiplying the decimal number by 2 repeatedly and recording the results in order.
- Adding the decimal number to 2 repeatedly until the result is 0.

Fig. 2. An example of MCQs generated by EduCS from easy to difficult to the same knowledge point according to varying student levels.

4.2 The Different Types of MCQs

Study 2. We input the teaching outline of the Introduction to Computer Science into EduCS. Other settings remain unchanged. We investigated whether EduCS has the ability to generate different types of MCQs. The type here refers to declarative MCQs and questioning MCQs. Furthermore, conceptual MCQs, numerical MCQs.

Results. Figure 3 shows the two expression forms of MCQs generated by EduCS, with the description form on the left and the question form on the right. Figure 4 shows the MCQs generated by EduCS for two types of assessment content, with conceptual knowledge assessed on the left and numerical operation-related ability evaluated on the right. It can be seen that EduCS can generate different types of MCQs.



MCQ

Stem

The CPU consists of three main components: the control unit, the arithmetic logic unit (ALU), and the ___ .

Correct Answer

- memory unit

Distractors

- input unit
- output unit
- storage unit

MCQ

Stem

What is a common input device for a computer?

Correct Answer

- Keyboard

Distractors

- Monitor
- Printer
- Speaker

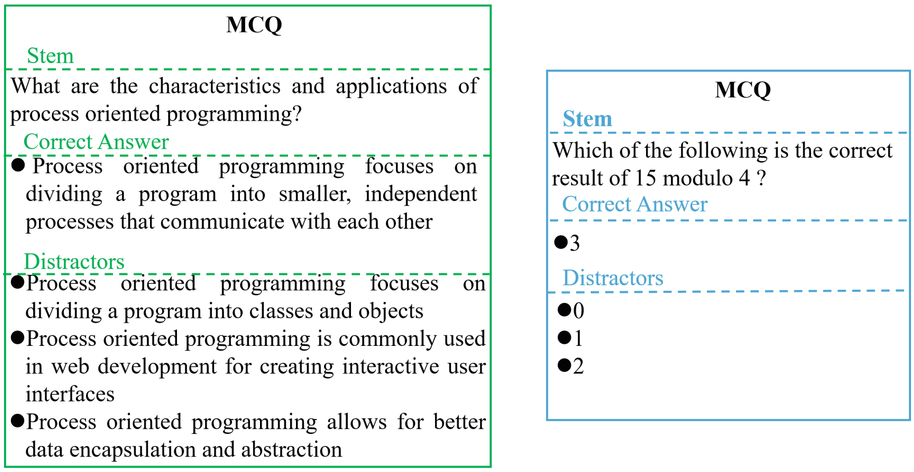
a) Declarative b) Dnterrogative

Fig. 3. MCQs for two expression forms generated by EduCS.

4.3 Quality Evaluation

Study 3. To verify the quality of MCQs generated by EduCS. We will design quality evaluation metrics based on the characteristics and educational needs of each part of the MCQs. An MCQ consists of three distinct parts, each with its own unique features. Accordingly, different metrics are defined to assess the quality of individual components. The metrics and their definitions are presented in Table 1. All metrics are measured on a scale of 0 to 10, where 0 indicates that the criteria are not met at all, and 10 signifies complete satisfaction of the criteria.

We utilize EduCS to generate 50 intermediate-level MCQs for computer introduction course. Concurrently, we select 50 MCQs that have been used in previous years for this course at the Beijing Institute of Technology. Subsequently, we subject these 100 MCQs to expert evaluation. Two experts are invited to participate in the assessment: one is a computer teacher with extensive teaching experience, and the other is a professor from the Department of Computer Science at the Beijing Institute of Technology. Finally, we analyze the data to demonstrate the quality of the MCQs.



MCQ

Stem

What are the characteristics and applications of process oriented programming?

Correct Answer

- Process oriented programming focuses on dividing a program into smaller, independent processes that communicate with each other

Distractors

- Process oriented programming focuses on dividing a program into classes and objects
- Process oriented programming is commonly used in web development for creating interactive user interfaces
- Process oriented programming allows for better data encapsulation and abstraction

a) Conceptual

MCQ

Stem

Which of the following is the correct result of 15 modulo 4 ?

Correct Answer

- 3

Distractors

- 0
- 1
- 2

b) Numerical operation

Fig. 4. Two types of MCQs generated by EduCS,one of which assesses conceptual knowledge, while the other focuses on numerical calculations.

Table 1. Quality evaluation metrics

Metrics for the Stem	Definition
answerableness	the ease or suitability of a question being answered
relevance to knowledge	knowledge-point relevance
adequate context	to effectively understand, interpret, or assess a stem
Metrics for the Correct Answer	**Definition**
semantic correctness	the accuracy of the meaning conveyed by the statement
Metrics for the Distractors	**Definition**
discriminating power	Confusing and disruptive to others
closeness	at least one is close to the true answer
Metrics for the overall	**Definition**
guidance for learning	helpful for learning
moderate difficulty	the difficulty level of the question
grammatical correctness	syntactically correc
clearness	at least one is close to true answer

Results. Based on the statistical analysis conducted on the scoring results of human-generated MCQs and EduCS-generated MCQs, we obtained Fig. 5. In this figure, the left bar represents the EduCS-generated MCQs, while the right bar represents the human-generated MCQs.

From the graph, we observe that the MCQs generated by EduCS achieved scores exceeding 8 points on the five metrics: relevance to knowledge, adequate context, semantic correctness, grammatical correctness, and clearness. Particularly, the metrics of relevance to knowledge and grammatical correctness were close to the scores of human-generated MCQs. This indicates that MCQs generated using EduCS have great potential in terms of quality.

There is a significant difference in the scores for discriminating power and closeness compared to the metrics scores of human-generated MCQs. This suggests that generating distractors is more challenging than constructing the stem and correct answers. The moderate difficulty score was 7.9, which is close to the level set for intermediate students.

Overall, the trend in the metrics scores of EduCS-generated MCQs aligns with that of human-generated MCQs. This demonstrates the powerful capability of EduCS in generating MCQs. However, it is important to note that even though EduCS has automatic generation capabilities, it is still recommended to manually review and correct the MCQs before delivering them to students. This is because automated systems may not always capture subtle nuances or context-specific details accurately. Manual review allows for human expertise and judgment to ensure the quality and appropriateness of the questions.

By combining the automatic generation capability of EduCS with manual review, educators can benefit from the efficiency of generating a large number of MCQs while still ensuring the accuracy, relevance, and educational value of the questions.

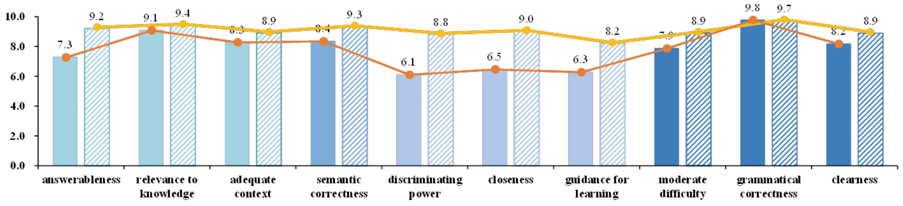


Fig. 5. Comparison of average scores of quality metrics between human-generated MCQs and EduCS-generated MCQs. The left bar corresponds to the EduCS-generated MCQs and the right bar corresponds to the human-generated MCQs.

5 Conclusion

In this study, we designed a set of prompt chains to optimize the performance of LLMs in CS education. Subsequently, we have developed and implemented EduCS, a system based on GPT-3.5, which is capable of automatically generating MCQs aligned with the CS0/CS1 course outline. Based on the experimental results, we made the following observations: 1) EduCS has the ability to generate MCQs of varying difficulty levels based on individual student proficiency. This personalized teaching approach is highly significant in education; 2) Although the generated questions require teacher verification before being delivered to students, they demonstrate great potential in terms of quality. 3) EduCS exhibits the ability to generate complete and diverse types of MCQs, which can be used for both formative and summative assessments. Automating the process of generating MCQs, significantly reduces the workload associated with formative evaluation in the teaching process. EduCS holds excellent value for formative assessment in CS education.

Acknowledgement. This work was supported by the grants of the following program: National Natural Science Foundation of China (NSFC, No. 62077004, 62177005)

References

1. Finnie-Ansley, J., Denny, P., Becker, B.A., Luxton-Reilly, A., Prather, J.: The robots are coming: exploring the implications of openai codex on introductory programming. In: Proceedings of the 24th Australasian Computing Education Conference, pp. 10–19 (2022)
2. Denny, P., Kumar, V., Giacaman, N.: Conversing with copilot: exploring prompt engineering for solving cs1 problems using natural language. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1, pp. 1136–1142 (2023)
3. MacNeil, S., et al.: Experiences from using code explanations generated by large language models in a web software development e-book. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1. pp. 931–937 (2023)
4. Tran, A., Angelikas, K., Rama, E., Okechukwu, C., Smith IV, D.H., MacNeil, S.: Generating multiple choice questions for computing courses using large language models
5. Heilman, M., Smith, N.A.: Good question! statistical ranking for question generation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 609–617 (2010)
6. Du, X., Shao, J., Cardie, C.: Learning to ask: neural question generation for reading comprehension. arXiv preprint arXiv:1705.00106 (2017)
7. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
8. Tafjord, O., Clark, P.: General-purpose question-answering with macaw. arXiv preprint arXiv:2109.02593 (2021)
9. Dijkstra, R., Genç, Z., Kayal, S., Kamps, J., et al.: Reading comprehension quiz generation using generative pre-trained transformers (2022)
10. Gabajiwala, E., Mehta, P., Singh, R., Koshy, R.: Quiz maker: automatic quiz generation from text using NLP. In: Singh, P.K., Wierzchon, S.T., Chhabra, J.K., Tanwar, S. (eds.) Futuristic Trends in Networks and Computing Technologies, pp. 523–533. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-5037-7_37
11. Ch, D.R., Saha, S.K.: Automatic multiple choice question generation from text: a survey. IEEE Trans. Learn. Technol. **13**(1), 14–25 (2018)
12. Mitkov, R., Varga, A., Rello, L., et al.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics, pp. 49–56 (2009)
13. Patra, R., Saha, S.K.: A hybrid approach for automatic generation of named entity distractors for multiple choice questions. Educ. Inf. Technol. **24**, 973–993 (2019)
14. Agarwal, M., Mannem, P.: Automatic gap-fill question generation from text books. In: Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 56–64 (2011)

15. Araki, J., Rajagopal, D., Sankaranarayanan, S., Holm, S., Yamakawa, Y., Mitamura, T.: Generating questions and multiple-choice answers using semantic analysis of texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1125–1136 (2016)
16. Reynolds, L., McDonell, K.: Prompt programming for large language models: Beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–7 (2021)
17. Reeves, B., et al.: Evaluating the performance of code generation models for solving parsons problems with small prompt variations. In: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 299–305 (2023)
18. Liu, M., Rus, V., Liu, L.: Automatic Chinese multiple choice question generation using mixed similarity strategy. IEEE Trans. Learn. Technol. **11**(2), 193–202 (2017)