TreeQuestion: Assessing Conceptual Learning Outcomes with LLM-Generated Multiple-Choice Questions

ZIRUI CHENG, Tsinghua University, China
JINGFEI XU, University of Wisconsin Madison, USA
HAOJIAN JIN, University of California San Diego, USA

The advances of generative AI have posed a challenge for using open-ended questions to assess conceptual learning outcomes, as it is increasingly common for students to use tools like ChatGPT to generate long textual answers. However, teachers still have to spend substantial time reading the answers and inferring students' learning outcomes. We present TreeQuestion, a human-in-the-loop system designed to help teachers create a set of multiple-choice questions to assess students' conceptual learning outcomes. When a teacher seeks to assess students' comprehension of specific concepts, TreeQuestion taps into the wealth of knowledge embedded within large language models and generates a set of multiple-choice questions organized in a tree-like structure. We evaluated TreeQuestion with 96 students and 10 teachers. Results indicated that students achieved similar performance in multiple-choice questions generated by TreeQuestion and open-ended questions graded by teachers. Meanwhile, TreeQuestion could reduce teachers' efforts in creating and grading the multiple-choice questions in contrast to manually generated open-ended questions. We estimate that in a hypothetical class with 20 students, using multiple-choice questions from TreeQuestion may require only 4.6% of the time compared to open-ended questions for assessing learning outcomes.

CCS Concepts: • **Human-centered computing → User interface toolkits**.

Additional Key Words and Phrases: Generative AI, Education, Large Language Models, Multiple-Choice Questions, Open-Ended Questions

ACM Reference Format:
Zirui Cheng, Jingfei Xu, and Haojian Jin. 2024. TreeQuestion: Assessing Conceptual Learning Outcomes with LLM-Generated Multiple-Choice Questions. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW2, Article 431 (November 2024), 29 pages. https://doi.org/10.1145/3686970

1 Introduction

Recent surveys find that a large majority of students (89%) are using ChatGPT for homework assignments [61], which has sparked numerous discussions concerning the challenges posed by AI in terms of plagiarism, cheating, and learning [13, 15]. One challenge around knowledge-based open-ended questions, particularly prevalent in STEM education [57], is that large language models (LLMs) can significantly reduce students' efforts in coming up with long textual answers to open-ended questions. Imagine a computer science professor who wants to assess students' comprehension of two cryptography concepts: hash and encryption. An example question would be as follows: *Explain the difference between hashing and encryption and the types of algorithms*

Authors' Contact Information: Zirui Cheng, chengzr19@mails.tsinghua.edu.cn, Tsinghua University, Beijing, China; Jingfei Xu, jxu496@wisc.edu, University of Wisconsin Madison, Madison, WI, USA; Haojian Jin, haojian@ucsd.edu, University of California San Diego, La Jolla, CA, USA.

ACM 2573-0142/2024/11-ART431
https://doi.org/10.1145/3686970

for each and when you should use each[1]. If students utilize the provided question as a prompt for ChatGPT's May 24, 2023 version, they will instantly obtain a 507-word response. The professor will have to spend substantial time reading these answers and guessing students' learning outcomes.

In response, teachers are actively searching for solutions to mitigate these concerns [13, 15]. For example, some teachers are starting to use AI plagiarism detection software (e.g., GPTZero [1], ZeroGPT [2]) to determine if the student has utilized AI. However, the results can sometimes be incorrect [30]. Meanwhile, some teachers have opted to completely redesign their assignments to incorporate AI usage. For example, professors at UCSD ask students to write essays with ChatGPT, identify the errors generated by ChatGPT, and find authoritative sources to provide accurate information [23]. Nevertheless, it is not easy to generalize these attempts across disciplines, and designing and evaluating these non-standard assignments require significant extra effort.

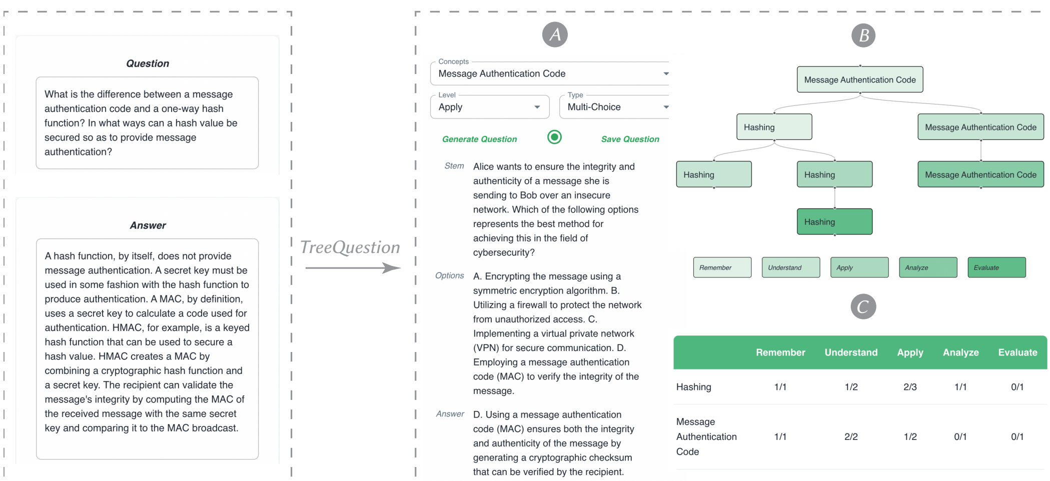

Fig. 1. TreeQuestion assists teachers in generating a set of auto-gradable multiple-choice questions, serving as an alternative to conventional knowledge-based open-ended questions that require manual grading. (A) TreeQuestion supports teachers in designing a scenario-based multiple-choice question to assess students' learning outcomes in terms of applying the four concepts including *MAC* and *Hashing* in a given situation. (B) TreeQuestion arranges a series of multiple-choice questions at varying levels into a tree-like structure. (C) TreeQuestion represents a student's learning outcome as a matrix, with rows indicating the concepts and columns specifying the levels of understanding.

We present TreeQuestion, a human-in-the-loop system designed to help teachers create a set of multiple-choice questions to assess students' conceptual learning outcomes. When a teacher seeks to assess students' comprehension of specific concepts, TreeQuestion taps into the wealth of knowledge embedded within LLMs and breaks down the open-ended question into a series of multiple-choice questions organized in a tree-like structure (Fig. 1). TreeQuestion then crafts distractor options for these questions systematically by leveraging the ability of LLMs to generate seemingly accurate answers. Students who grasp the concept can find the correct answer in TreeQuestion quickly, while those who do not, struggle. Meanwhile, teachers can grade the conceptual questions automatically.

Yet, generating a set of multiple-choice questions correctly and meaningfully is challenging. First, compared with open-ended questions, which allow teachers to delve into students' thought

[1]This question is excerpted from chegg.com [12]

processes through the examination of their open-vocabulary responses [17], a single multiple-choice question only gives a binary assessment of a student's mastery of a concept. To tackle this problem, TreeQuestion uses Bloom's taxonomy [8] to classify students' learning outcomes into six levels hierarchically: Remember, Understand, Apply, Analyze, Evaluate, and Create. For each level, TreeQuestion generates a multiple-choice question aimed at evaluating that level of mastery of the target concept. Therefore, learning outcomes assessed by TreeQuestion can be presented in the form of matrices (as shown in Fig. 1C), where the rows stand for the concepts and the columns denote the level of understanding.

Second, LLMs are prone to introducing errors and are unaware of redundancies among distractor options. We conducted a pilot study by instructing a commercial service (i.e., ChatGPT) to generate multiple-choice questions with annotated answers. The results indicate that more than half of the questions were illogical due to issues like wrong answers, out-of-context information, or ambiguous options. To address this problem, we introduce a new programming pattern for interacting with LLMs called "Explore-Validate-Generate": TreeQuestion begins by generating knowledge graphs that encompasses the given set of concepts as well as their related concepts. TreeQuestion then allows teachers to quickly validate the correctness of the knowledge graph and generate multiple-choice questions accordingly.

Evaluation: We evaluated TreeQuestion with 96 students and 10 teachers. First, we designed 6 pairs of matched open-ended questions and multiple-choice questions in computer science. 96 student participants answered these questions in a survey. We compared students' responses to both open-ended questions and multiple-choice questions generated from TreeQuestion. Student performance data suggest that in the areas we investigated, there are no significant differences related to question types, indicating that well-designed multiple-choice questions from TreeQuestion can assess students' learning outcomes similarly to open-ended questions. Second, we invited 10 teachers to create open-ended questions without TreeQuestion and multiple-choice questions with TreeQuestion for knowledge assessment. TreeQuestion could reduce teachers' efforts in creating and grading the multiple-choice questions compared with open-ended questions. Based on the results, we estimate that in a hypothetical class with 20 students, using multiple-choice questions with TreeQuestion requires only 4.6% of the time compared to the time needed for assessing learning outcomes using open-ended questions. Participants commented that TreeQuestion reduced their efforts by providing abundant choices in background knowledge, question distractors, and difficulty levels that could be easily incorporated into normal education practices.

Limitations: TreeQuestion is designed to only assess conceptual learning outcomes. However, educators may use open-ended questions for other educational purposes, such as fostering creative thinking. It is important to note that TreeQuestion is not designed to support these usages. Besides, a student may forward multiple-choice questions generated by TreeQuestion to LLMs. We consider this problem out of the scope of this paper.

Our specific research contributions are as follows:

- We develop and evaluate TreeQuestion as an end-to-end system, demonstrating the effectiveness and accuracy of assessing conceptual learning outcomes through multiple-choice questions generated by LLMs.
- We propose a method for generating multiple-choice questions, which can be used to systematically assess students' conceptual learning outcomes.
- We introduce a novel programming pattern, Explore-Validate-Generate, which streamlines human interactions with imperfect LLMs in creating multiple-choice questions.

2 Background and Related Work

TreeQuestion builds on ideas from three different areas: (1) open-ended questions and multiple-choice questions; (2) large language models in education; and (3) techniques for multiple-choice question generation.

2.1 Open-ended Questions and Multiple-Choice Questions

For decades, educators have been dedicated to discovering the most effective question representations for evaluating students' learning progress and enhancing the quality of teaching [4]. Open-ended questions are one of the most effective ways of reflecting students' knowledge and performance. Using open-ended questions, teachers can prompt students to develop more complex higher-order thinking skills. Researchers often consider open-ended questions to be more adept at measuring critical thinking [49] and encouraging deeper learning [69]. Although it is generally easy to write high-order open-ended questions, grading them is time-consuming and not amenable to automation. Multiple-choice questions can also be effective assessment tools with a variety of benefits for both teachers and students [43, 66]. Despite some concerns that multiple-choice questions only measure memorization of facts, there is also evidence showing that well-constructed multiple-choice questions are capable of both assessing and encouraging deeper levels of processing [24, 37, 44]. Multiple-choice questions can be efficient, objective, easy to grade automatically, and can be successfully used to assess a variety of learning outcomes. However, writing high-quality multiple-choice questions is time-consuming because of the difficulties in figuring out key messages and coming up with distractors [56].

2.2 Large Language Models in Education

Recent advances in large language models have shown great promise for both teachers and students [32]. On the one hand, large language models can be used in different ways to improve student learning and engagement such as generating interactive materials [16], acting as tutor agents [3] or tutee agents [31], and providing peer feedback [29, 48]. On the other hand, researchers also addressed the use of large language models from the teacher's perspective to facilitate the assessment of student answers [42], generation adaptive feedback [71], and preparation of teaching content [46]. However, the misapplication of large language models also poses substantial challenges within the realm of education, especially the risk of academic dishonesty. For example, students can potentially use large language models to cheat on open-ended questions by copying and pasting the generated responses into their assignments [13, 15, 33]. In response to such concerns, automatic detection of machine-generated text has been widely discussed in the NLP community [6, 22, 28, 41, 53]. Available tools such as GPTZero [1] and ZeroGPT [2] can also help people investigate machine involvement in writing tasks to some extent. However, it is still difficult to completely distinguish whether a text is generated from humans or machines with these tools [13, 21, 32]. In the real world, these tools can only provide limited assistance. The results generated from these tools still make it difficult for people to draw valid conclusions. In this paper, instead of seeking post-hoc methods to detect cheating, we turn to an ad-hoc reflection on the process of knowledge assessment — we attempt to leverage the generative capabilities of large language models to enhance efficient knowledge assessment.

2.3 Techniques for Multiple-Choice Question Generation

Prior research has shown two main approaches for automatic multiple-choice question generation for educational purposes. One approach utilizes crowdsourcing techniques [58, 67]. The other approach focuses on developing end-to-end NLP models specifically designed for question generation, where questions are generated from a given text [51, 68, 70]. With recent advances in large language models, researchers have investigated the possibility of generating questions with large language models [16, 18, 47, 59]. However, existing automatic question generation techniques are good at creating factual questions [9, 14, 35], while not being able to generate questions that target educational goals lying in the higher levels of Bloom's Taxonomy [5, 8]. Therefore, the generated questions are often of low quality and limited in types and difficulty levels [10, 26, 35].

While automatic question generation might be insufficient to meet educational requirements, researchers have proposed different systems to support human-AI collaborative question generation. Wang et al proposed to modulate the automatic question generation process with different components to offer a flexible interface for instructors to control various aspects of the produced questions [56]. Lu et al proposed systems that support instructors to conveniently design high-quality questions to help students comprehend readings [38]. However, existing human-AI systems for multiple-choice question generation are still limited in the diversity of input sources, which may impose challenges on comprehensive knowledge assessment. In this paper, we see large language models themselves as the knowledge base and introduce a human-AI collaborative approach to explore systematic knowledge assessment.

3 Pilot Study

To inform the design of TreeQuestion, we examine the feasibility and challenges of using LLMs to generate multiple-choice questions for testing students' conceptual learning outcomes.

Method. We first gathered a preliminary collection of prompts from two professors who are actively using LLMs to generate quiz questions for their classes. We observed that most prompts are instructions (e.g., generate a multiple-choice question that ...), and these prompts frequently differ in terms of the number of questions desired by the professor, the number of concepts involved, and the cognitive depth required to respond to each question. To ease the quality assessment process, we then adapted these prompts to generate questions around a few common concepts in computer cryptography. Table 1 enumerates a few example prompts utilized in the experiment. For each prompt, we called the `gpt-3.5-turbo` API 10 times with a temperature of 1, which signals a relatively high degree of randomness. In the end, two authors manually assessed the quality of the generated questions.

Results. We made the following key observations. First, **LLMs can generate high-quality questions with various context information, particularly ones that simulate real-world experiences.** Fig. 1A illustrates an example question, which depicts a scenario in which Alice wants to ensure the integrity and authenticity of a message. Conventionally, generating scenario-based questions has been a time-consuming process for teachers, as they must brainstorm both suitable scenarios and corresponding options [39]. By utilizing scenario-based questions generated by LLMs, we can transform many multiple-choice questions from merely assessing simple memorization to evaluating higher-order thinking skills, such as analytical thinking, at a low cost.

Second, **LLMs may introduce inaccurate, ambiguous, and out-of-scope information, and teachers must manually inspect the content to ensure accuracy.** In our study, we observed that LLMs sometimes annotate the generated questions with incorrect answers and provide seemingly plausible explanations. Indeed, LLMs may not even be able to annotate the same question with consistent answers across different runs. Possible reasons are that LLMs work by modeling the probability of co-located words and are limited in reasoning capabilities [20, 55]. Moreover, LLMs frequently generate highly ambiguous options that are hard to falsify. There are also instances where LLMs introduce concepts far beyond the expected scope of the prompts. As a result, the two

Prompt Types	Example Prompts	Identified Challenges
Single question, low-level	`Generate one multiple-choice question related to the concept of symmetric encryption.`	Ambiguous expression in distractors.
Single question, high level	`Provide a multiple-choice question that involves a comprehensive analysis and evaluation regarding symmetric encryption.`	Inconsistent answers to questions.
Multiple questions, low-level	`Access the basic understanding of symmetric encryption and asymmetric encryption by giving a set of multiple-choice questions.`	Multiple answers among options.
Multiple questions, high-level	`Compare and contrast the in-depth comprehension regarding symmetric encryption and asymmetric encryption using a set of multiple-choice questions.`	Insufficient questions at higher levels.
Group questions, high-level	`Design a set of 10 multiple-choice questions on higher levels involving interaction between symmetric and asymmetric encryption.`	Homogenized answers across different questions.

Table 1. Example results from the pilot study to examine the feasibility and challenges of large language models in generating multiple-choice questions.

professors who provided the initial set of prompts mentioned that they often have to experiment with multiple prompts ($N > 5$) and manually scrutinize the content to identify a usable question.

Thirdly, **the questions generated in batches by LLMs often lack diversity**. Since it typically requires searching through blindly generated questions to find a usable one, a common approach for teachers is to prompt LLMs to generate 5-10 questions in batches. We found that the generated questions in these batch-generation experiments often turn out to be redundant and only assess a limited range of knowledge. In some cases, LLMs may even repetitively employ the same phrase. This production of redundant content [27, 60] can likely be attributed to the exposure bias inherent in large language models [45, 52]. Furthermore, blindly generated multiple-choice questions tend to exhibit a bias toward involving only low-level cognitive processes. For instance, when we instructed LLMs to create 10 multiple-choice questions related to symmetric encryption, 8 of them were restricted to memorizing and recalling tasks.

4 Assessing Conceptual Learning Outcomes using Multiple-Choice Questions

In this section, we discuss the rationale of our learning outcome formulation, and how we use multiple-choice questions to measure the learning outcomes.

Defining Fine-grained Learning Outcomes. We employ Bloom's Taxonomy [5] to define the learning outcomes of students. Bloom's Taxonomy is widely used today by educators around the world [54], which outlines six hierarchical levels of learning: Remember, Understand, Apply, Analyze, Evaluate, and Create. The lower levels of Bloom's taxonomy focus on the knowledge that we want our students to remember and understand. The middle levels focus on the application and analysis of information. At the top of Bloom's taxonomy are tasks that involve evaluating and creating.

TreeQuestion formulates the learning outcomes as the tested concepts and the associated levels of understanding. For instance, a student might have a nuanced understanding of the concept "Hash", yet only be able to recite the definition of "MAC". This could yield a granular score profile like:

"*Hash*" : "*Evaluate*", "*Sign*" : "*Understand*", "*MAC*" : "*Remember*", ...

Note that since students only answer multiple-choice questions without actively producing original content, TreeQuestion cannot assess a user's comprehension of a concept at the "create" level.

Assessing Learning Outcomes with Multiple-Choice Questions. Evaluating students' fine-grained conceptual learning outcomes has been challenging. Traditional examination scores only offer an aggregated view of learning outcomes within a particular subject. To unearth students' thought patterns and pinpoint their knowledge shortfalls, researchers often have to analyze open-ended responses [17], a process that demands substantial time investment.

In contrast, the central premise of TreeQuestion is to generate a set of multiple-choice questions, each tailored to gauge a specific level of understanding of certain concepts. Multiple-choice questions can be easy to grade automatically, thus enabling efficient assessment of students' learning outcomes. For instance, the question depicted in Fig. 1 prompts students to apply acquired knowledge in a novel context. A correct response to this question suggests that the student has likely reached the "apply" stage of Bloom's taxonomy for the four intertwined concepts: encryption, firewall, VPN, and MAC.

TreeQuestion represents a student's learning outcome in the form of a matrix (as shown in Fig. 1C), where the rows stand for the concepts and the columns denote the level of understanding. The scores in each cell of the matrix can be computed according to students' responses to questions specified for certain concepts and levels. It is important to note that due to the nature of multiple-choice questions, a student might select the correct answer without fully understanding all the related concepts. As a result, we present the outcomes in their raw form, keeping track of instances where a student correctly selects or refrains from selecting a concept.

5 Explore-Validate-Generate

To generate the multiple-choice questions described above, we introduce a new programming pattern for interacting with large language models called "Explore-Validate-Generate", which guides the question-generation process to meet the design goals. Prior works in both crowdsourcing [7] and large language models [64, 65] have also introduced similar ideas like modulating the scope of sub-tasks to ensure the quality of complex tasks. Inspired by them, TreeQuestion aims to use the Explore-Validate-Generate pattern to address the challenges of large language models in question generation tasks. The Explore-Validate-Generate pattern splits the whole task into three stages where large language models and humans can contribute complementary capabilities. The workflow of Explore-Validate-Generate in TreeQuestion is illustrated in Fig. 2.

5.1 Explore: Extracting Background Knowledge with Concepts

The Explore stage leverages large language models to produce background knowledge tailored for multiple-choice question generation. With user-specified concepts, fields, and levels, TreeQuestion leverage Bloom's Taxonomy to extract abundant background knowledge from large language models.

Imagine that users may want to test students' knowledge about symmetric encryption and asymmetric encryption in network security, they could specify the concept as "symmetric encryption"

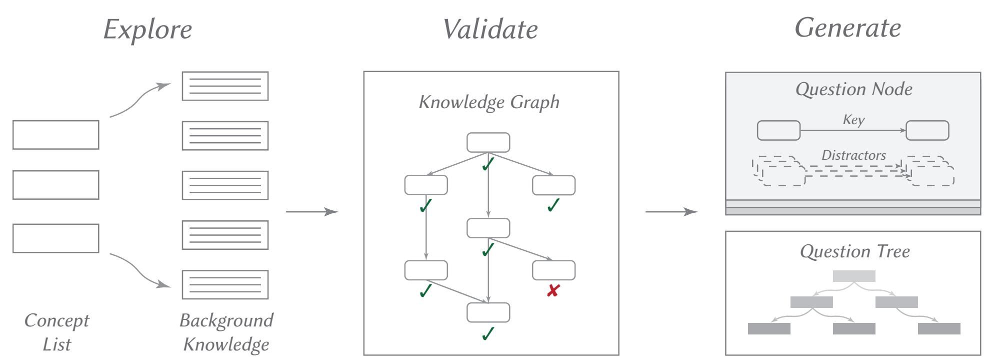

Fig. 2. The workflow of TreeQuestion: (A) *Explore*: Users specify a concept list and TreeQuestion leverages large language models to generate paragraphs of background knowledge based on pre-defined educational goals; (B) *Validate*: TreeQuestion transforms background knowledge into a knowledge graph, and users can review the generated content presented in an interactive interface to ensure accuracy; (C) *Generate*: TreeQuestion helps users generate the key and distractors in the multiple-choice questions with pairs of nodes and edges from the knowledge graph. Afterward, users can create multiple-choice questions targeted at different levels in the question tree.

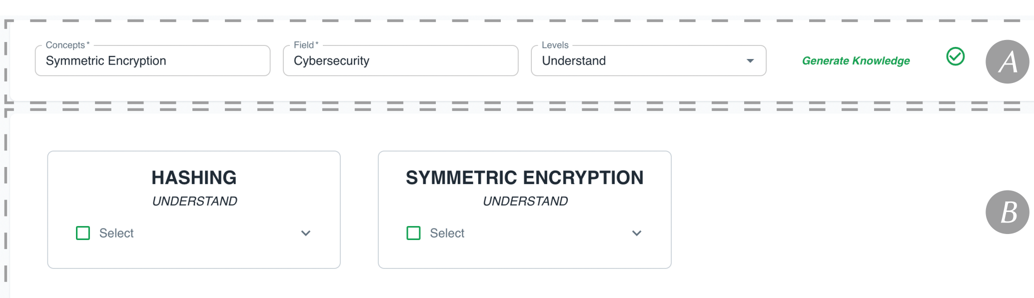

Fig. 3. The Explore interface of TreeQuestion: (A) users input *concept*, *field*, and *level* information of background knowledge; (2) large language models extract targeted knowledge according to user input.

and "asymmetric encryption" and specify the field as "network security" as Fig. 3 A shows. To instruct large language models to generate background knowledge, users can choose different levels in Bloom's Taxonomy including Remember, Understand, Apply, Analyze, Evaluate, and Create in the interface. The content generated from large language models is directly displayed in the interface as Fig. 3 B shows. When the users are satisfied with the generated content, they can click on the "select" checkbox to select the content for later use. It is also possible that the users might not think the generated background knowledge can meet their expectations. In this case, they can also update the specifications in the interface and re-generate the background knowledge.

To obtain background knowledge that can meet diverse purposes for question generation, it is still challenging to design precise prompting strategies to instruct large language models. Here we extracted detailed explanations of different levels in Bloom's Taxonomy from existing literature [5] as part of the prompts. As is suggested by previous work [5], there are also multiple cognitive

Input	Generate some statements regarding the learning objectives of **hashing** in the area of **cybersecurity**. The statements should cover the **understand** level in Bloom's Taxonomy. Each statement should start with a verb. The definitions of the **understand** level in Bloom's Taxonomy are: Understand means construct meaning from instructional messages, including oral, written, and graphic communication. Cognitive processes in the category of understand include interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining. Exemplifying means finding a specific example or illustration of a concept or principle.
Output	Providing examples of commonly used hashing algorithms.

(a) Stage 1: Generate learning objectives.

Input	Here are some statements regarding the learning objectives of **hashing** in the area of **cybersecurity**. Providing examples of commonly used hashing algorithms. Generate some detailed knowledge according to these statements.
Output	Examples of commonly used hashing algorithms are MD5, SHA-1, and SHA-256. These algorithms take the input data and produce a unique hash value as the output.

(b) Stage 2: Generate background knowledge.

Table 2. Example input and output of prompting large language models with the concept "hashing" targeted at the "understand" level in Bloom's Taxonomy in the Explore stage. First, we generate learning objectives with the detailed explanations of different levels and related cognitive processes in Bloom's Taxonomy extracted from previous literature [5]. Second, we leverage the learning objectives to instruct large language models to generate background knowledge.

processes involved in each level of Bloom's Taxonomy, which delineate the breadth and boundaries of them. Therefore, we also included the specific definitions of each cognitive process associated with each level in the prompts. For example, if a user seeks to get background knowledge of "hashing" at the "apply" level, TreeQuestion would incorporate the definitions of "apply" like "carrying out or using a procedure in a given situation" in the prompts. The "apply" level then consists of two cognitive processes: "executing" – when the task is a familiar exercise and "implementing" – when the task is an unfamiliar problem. All the definitions of the level and the related cognitive processes are provided in the prompts to give large language models more information for precise background knowledge generation.

In practice, we found that state-of-the-art large language models such as `gpt-3.5-turbo`[2] provided by OpenAI demonstrated excellent capabilities of understanding the provided explanations of Bloom's Taxonomy. The prompts used by TreeQuestion can successfully extract the background knowledge embedded within large language models related to different concepts. However, we still observed that within single runs, there were some unsatisfactory cases where large language models only gave abstract descriptions of the learning objectives without explaining the background knowledge in detail. To handle this problem, we split the generation task into two sub-stages where the learning objectives and background knowledge are generated sequentially as suggested by prior work [65]. By doing so, the previously mentioned unexpected cases rarely existed in our

[2]https://platform.openai.com/docs/models/gpt-3-5

system. Examples of the input and output are illustrated in Table 2. We first guided large language models to derive specific educational goals according to the explanations of levels and related cognitive processes provided in the prompts. With the goals obtained, we then instructed large language models to generate detailed knowledge. More example prompts covering different levels and related cognitive processes in this stage have been included in the appendix A.

5.2 Validate: Verifying Correctness over Knowledge Graphs

Since the background knowledge generated from large language models might contain inaccurate information, TreeQuestion enables users to check the correctness of background knowledge in the Validate stage. TreeQuestion transforms the background knowledge into knowledge graphs with the help of large language models. Users can *create*, *update*, and *delete* the nodes and edges which would be used as materials to generate the options in multiple-choice questions.

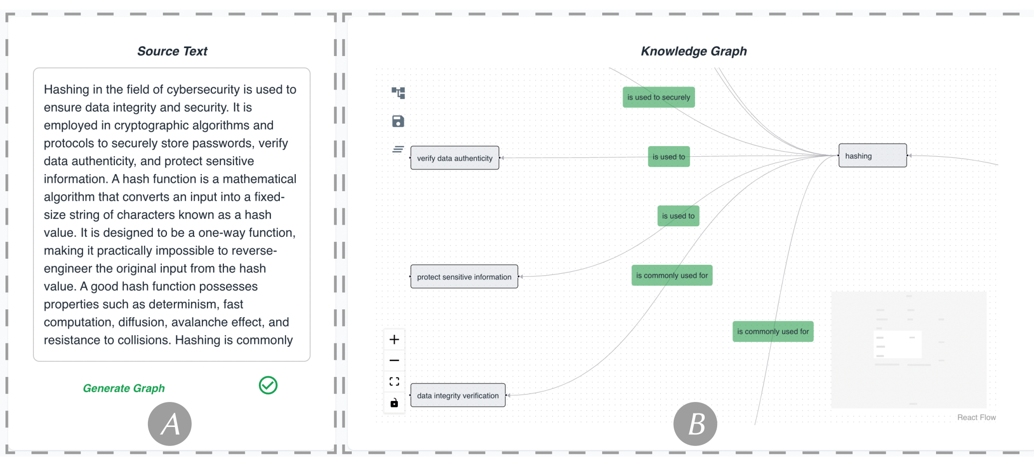

Fig. 4. The Validate interface of TreeQuestion: (A) users input *concept*, *field*, and *level* information of background knowledge; (2) large language models extracts targeted knowledge according to user input.

TreeQuestion initiates a knowledge graph by prompting large language models with the background knowledge generated in the Explore stage. User can also modify the content in the text field to customize the construction of knowledge graphs. TreeQuestion maps the results from large language models into an interactive interface where nodes denote the concepts and edges denote the relations. An example of the interactive interface in the Validate stage is shown in Fig. 4. We defined three basic operations of both nodes and edges including *create*, *update*, and *delete* in the interactive interface. Users can validate the knowledge graphs through iterative review and modification of nodes and edges.

- **Create**: Although the knowledge graphs have covered intensive knowledge related to the source concepts, it is still possible that large language models miss key knowledge related to the source concept. We allow users to create new nodes and edges by dragging them out from the hurdles of an existing node and connecting the hurdles of existing nodes, respectively.
- **Update**: The nodes and links presented in the knowledge graph might not be accurate since they are purely generated from large language models. We allow users to modify the labels simply by clicking on the nodes and links.

Input	`Create a knowledge graph on the concepts related to` **`hashing`** `in the field of` **`cybersecurity`** `based on the following source text. Generate a list of relations with 3 columns in this order: subject, predicate, and object.` `Hashing is a process that converts data into a fixed-size string. In cybersecurity, hashing ensures data integrity, and authenticity, and protects sensitive information. Common hashing algorithms include MD5 and SHA-256, with MD5 used for checksums and SHA-256 for cryptographic purposes.`		
Output	**`Subject`**	**`Predicate`**	**`Object`**
	`Hashing`	`ensures`	`data integrity`
	`Hashing`	`ensures`	`authenticity`
	`Hashing`	`protects`	`sensitive information`
	`Hashing algorithms`	`include`	`MD5`
	`Hashing algorithms`	`include`	`SHA-256`
	`MD5`	`is used for`	`checksums`
	`SHA-256`	`is used for`	`cryptographic purposes`

Table 3. Example input and output of prompting large language models with the concept "hashing" in the Validate stage. We prompt large language models to generate a list of triplets with relations based on the background knowledge generated in the previous stage.

- **Delete**: Since large language models may produce some irrelevant knowledge that might not be valuable for question generation, we allow users to delete unnecessary nodes or edges by simply tapping the backspace key.

Through the design of graphical interfaces in the Validate stage, TreeQuestion aims to enhance the efficiency of both knowledge validation and distractor generation of multiple-choice questions. On the one hand, knowledge graphs can present abundant information about the underlying relationships of different concepts within the background knowledge, which enables the more flexible organization of concepts and reduces the cognitive load needed for comprehension [36, 62]. On the other hand, plausible distractors in multiple-choice questions are often created based on text transformation such as adding, removing, or changing the components in the original sentence [56]. With the node and edge representation of the concepts and relations in knowledge graphs, users can easily find and manipulate components to create potential distractors in multiple-choice questions in the Generate stage.

In order to construct knowledge graphs from background knowledge, we directly prompt large language models to generate a list of relations consisting of *subject*, *predicate*, and *object* in a zero-shot setting. The original background knowledge selected by users was directly incorporated into the prompts. In practice, state-of-the-art large language models also showed satisfactory performance in constructing the list of relations. An example of the input and output when calling large language models in the Validate stage is shown in Table 3. The parsed subjects and objects will be mapped into nodes and the parsed predicates will be mapped into edges between nodes.

5.3 Generate: Creating Questions at Multiple Levels

Given the abundant knowledge information provided in previous stages, TreeQuestion enables users to design questions targeted at different levels in Bloom's Taxonomy to test students' learning outcomes systematically in the Generate stage.

Fig. 5. In the Generate stage, TreeQuestion provides support for key and distractor generation based on the relationships gathered in the knowledge graph. Instructors can generate (A) key statements indicating correct options and (B) distractor statements indicating distracting options by (C) selecting proper question templates so as to control the options in the multiple-choice questions.

The interfaces of the Generate stage are shown in Fig. 5 and Fig. 6. Users can generate multiple-choice questions targeted at different levels with the knowledge graphs created in previous stages. To help users identify the knowledge gap between students, we allow them to customize the options by specifying keys and distractors. Yet, they can also proceed without specifying keys or distractors in the question generation stage. Prior studies argue that coming up with distractors is considered a difficult task in question generation tasks for users [56]. We hence leverage large language models to generate potential distractors based on the triplets of *subject*, *predicate*, and *object* obtained in the Validate stage. Reasonable distractors for multiple-choice questions should be incorrect in content but relevant to the questions. We provided multiple question templates curated from

Input	`Generate concrete rules for writing the distractors in multiple-choice questions regarding the statement "`**`MD5 is an example of a hashing algorithm`**`". The rules for creating the distractors should be based on the subject, predicate, and object presented in the statement. The statement might appear in the form of the following question template. [object] + is + [concept]. For example, [object] + is + [example 1] + [example 2] + [example 3].`

Output	`1. Ensure the distractors are other hashing algorithms.`
	`2. Include distractors that are not related to hashing algorithms.`
	`3. Create distractors that are incorrect or false statements about MD5.`

(a) Stage 1: Generate concrete rules.

Input	`Generate several distractors regarding the statement "`**`MD5 is an example of hashing algorithm`**`".`
	`The distractors should be generated based on the following rules.`
	`1. Ensure the distractors are other hashing algorithms.`
	`2. Include distractors that are not related to hashing algorithms.`
	`3. Create distractors that are incorrect or false statements about MD5.`

Output	`1. SHA-1 is an example of the hashing algorithm.`
	`2. RSA encryption is an example of the hashing algorithm.`
	`3. MD5 is a symmetric encryption algorithm.`

(b) Stage 2: Generate specific distractors.

Table 4. Example input and output of prompting large language models to generate distractors with the concept "hashing" in the Generate stage. First, we prompt large language models to create rules to guide the creation of distractions based on the question template provided in the interface. Second, we prompt large language models to craft distractors with the rules.

online multiple-choice questions[3] to guide potential distractor generation. Examples of the question templates are also demonstrated in Fig. 5.

After specifying potential keys and distractors, users can label the concepts and the levels they want to include in the questions after choosing the keys and distractors provided in previous steps. TreeQuestion then calls large language models to generate one multiple-choice question at the specified level in Bloom's Taxonomy. The generated question will be displayed in the left panel of the interface where users can further modify the stems, options, and answers of the questions.

After obtaining satisfactory questions, users can save them as question nodes in the right panel of the interface. TreeQuestion will arrange the generated questions in a tree-like structure according to the concept and level corresponding to the question. In this manner, TreeQuestion can generate a full list of questions simply through a pre-order traversal of the tree where questions of related concepts are organized based on their corresponding levels. Users can also update or delete any node or edge of questions in the interface thus changing the order of questions in the generated question list.

In the Generate stage, large language models are instructed to generate plausible keys and distractors as well as generate multiple-choice questions based on the keys and distractors. Both keys and distractors are generated from the relations consisting of subject, predicate, and object from the knowledge graphs in the Validate stage. To generate plausible distractors according to the question templates provided in the interface, we also split the prompting process into two

[3]The questions are from chegg.com [11]

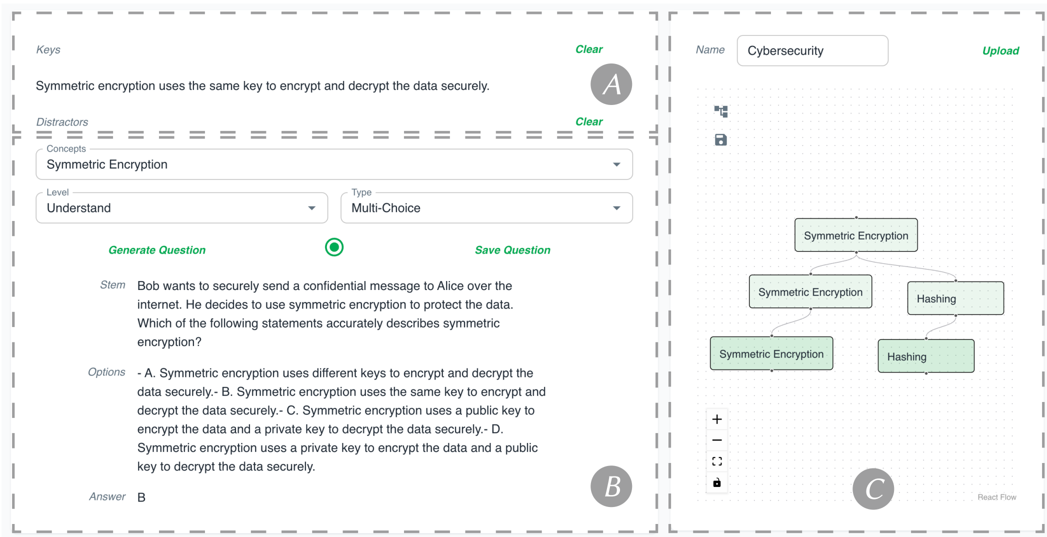

Fig. 6. In the Generate stage, TreeQuestion enables teachers to (A) customize keys and distractors in the question; (B) specify the concepts and levels to generate the question; (C) organize the questions at different levels into certain structures.

stages. First, we prompted large language models to generate rules for distractor creation. Then, we instructed the models to generate appropriate distractors based on the rules. Examples of the input and output for distractor generation are shown in Table 4. Users can select relevant templates based on their understanding to guide the generation of distractors. With keys or distractors specified, we prompt large language models to compose specific questions just with these keys or distractors.

6 System implementation

We implement TreeQuestion as a full-stack web application with a back-end written in Django[4] and a front-end written in React.js[5]. The front-end of TreeQuestion is deployed through DigitalOcean[6], and the back-end server is deployed as an AWS EC2 instance[7]. TreeQuestion uses `gpt-3.5-turbo`[8] API provided by OpenAI for all the large language model services. The `gpt-3.5-turbo` API provides tunable parameters such as the degree of randomness. TreeQuestion used the `gpt-3.5-turbo` engine, which was initially released on May 24, 2023. We used the default settings with a temperature of 1. All the API calls within TreeQuestion are in zero-shot settings without finetuning.

7 Evaluation Study

We conducted two IRB-approved studies to evaluate TreeQuestion. We collected students' responses to matched pairs of questions in certain fields of computer science through an online study to validate the effectiveness of multiple-choice questions generated from TreeQuestion. We also investigated teachers' experience in generating multiple-choice questions with TreeQuestion to understand the efficiency of TreeQuestion in question generation.

[4]https://www.djangoproject.com/
[5]https://react.dev/
[6]https://www.digitalocean.com/
[7]https://aws.amazon.com/
[8]https://platform.openai.com/docs/models/gpt-3-5

A Network Security

Q.1: *Which kinds of security protection activities should be better done by symmetric key encryption techniques? Which should be better done by public-key encryption techniques?*
Q.2: *What is the difference between a message authentication code and a one-way hash function? In what ways can a hash value be secured so as to provide message authentication?*

B Computer Organization

Q.3: *What effect does a cache and main memory have on a computer with two or more CPU chips?*
Q.4: *Differentiate CISC and RISC Architecture. Why does ARM use RISC architecture specifically?*

C Data Structures

Q.5: *Describe the algorithms used in solving mazes including Prim's Algorithm and Kruskal's Algorithm. How are they different from each other?*
Q.6: *Explain the differences between heap and stack. What are they used for in terms of memory?*

Fig. 7. Open-ended questions used in the evaluation study to collect students' responses. Our focus lies in conceptual questions in disciplines like network security, computer organization, and data structures within computer science.

7.1 Evaluating Students' Responses

To understand the effectiveness of the multiple-choice questions generated by TreeQuestion in assessing learning outcomes, we conducted a study on Amazon Mechanical Turk[9] to evaluate students' performance on both multiple-choice questions generated by TreeQuestion and open-ended questions.

7.1.1 Question Design. As suggested by prior work [58], we adopted *matched pairs of questions* to compare students' performance on both open-ended questions and multiple-choice questions generated by TreeQuestion. We conducted this study within the field of computer science. We collected open-ended questions in different college-level computer science courses from Chegg [11]. Three of our authors cross-validated these collected open-ended questions to make sure that: (1) the questions and answers provided are accurate within the corresponding disciplines; and (2) the questions are suitable for knowledge assessment. Examples of open-ended questions have been shown in Fig. 7.

Considering that different teachers have different expertise and preferences for designing questions, three of our authors created the multiple-choice questions for knowledge assessment following the standard procedure below. We created the multiple-choice questions based on the answers to each open-ended question to ensure that both the multiple-choice questions and open-ended questions can assess similar knowledge components [34]. The answers to each open-ended question were segmented by three of our authors into distinct sections based on the knowledge component involved in the question. We then generated the multiple-choice questions with the help of TreeQuestion by matching the segmented sections to similar keys provided in the Generate stage. The generated questions are distributed evenly across different levels of Bloom's Taxonomy.

7.1.2 Study Implementation. We recruited participants from Amazon Mechanical Turk located in the US to collect responses to the matched pairs of open-ended questions and multiple-choice

[9] https://www.mturk.com/

questions in a survey created by Qualtrics[10]. Participants were required to have studied a college-level course covering the knowledge to be tested in the question pairs. In this study, participants answered the open-ended questions first because we did not want the automatically generated multiple-choice questions to bias their answers. Participants were not allowed to modify their open-ended question answers after they had started answering the multiple-choice questions. The task took roughly 10 minutes to complete. Participants were compensated with $2 for their time.

We invited three teachers from an R1 institution in the US to collectively grade the responses collected from the student participants. All the teachers have teaching experience in the institution and have passing knowledge in the related fields. For multiple-choice questions, there is only one correct answer to each question. Students were given 1 if their answers were correct and 0 if their answers were incorrect. For each open-ended question, we used a strict grading criterion with a total score of 10 points. For each segmented section in the answers, we collectively determined analytic rubrics to grade the open-ended questions following the guidance proposed in prior works [4, 50]. We also set the total scores of each open-ended question to 10 points. We assigned similar points to each segmented section in the answers, which is proportional to the corresponding multiple-choice questions testing similar knowledge. For each segmented section, we divided potential responses into three to five levels from *unsatisfactory* to *satisfactory*, depending on different point values. For example, for the section "Why does ARM use RISC architecture specifically?", we gave students full points if they could justify the choice of RISC architecture for ARM processors with a comprehensive and well-structured argument without missing important factors. In contrast, we gave students no points if they failed to provide any reasonable explanation or offered incorrect reasoning for ARM's choice of RISC architecture. To estimate the time needed to grade open-ended questions, we randomly sampled 10 responses to these open-ended questions. Participants spent an average of 91.6 seconds (std. = 25.8, N = 30) to grade each response to these open-ended questions. Each teacher participant was compensated with a $30 Gift Card.

Number	Open-Ended Question		Multiple-Choice Question	
	avg.	std.	avg.	std.
Q.1	3.81	1.64	4.44	1.22
Q.2	3.68	2.18	4.00	2.57
Q.3	4.13	2.47	4.38	2.83
Q.4	3.56	2.48	3.69	2.44
Q.5	3.44	1.90	3.62	1.49
Q.6	3.75	1.80	3.94	2.51

Table 5. Comparison of participants' scores on both open-ended questions and multiple-choice questions. We use a 10-point scale for both formats of questions. Results from repeated measures ANOVA show that there is no main effect of questions, indicating that students can achieve comparable performance on TreeQuestion-generated multiple-choice questions and open-ended questions.

7.1.3 Findings. A total of 96 participants (46 identified as male, 50 identified as female) completed the task. The average age of participants was 33.03 ($std.$ = 8.83, N = 96). On average, each pair of questions created was answered by 16 different participants and each discipline of questions was answered by 32 different participants. All participants' answers to open-ended questions were graded by three of the teachers. Three of our teacher participants collectively decided the scores for answers to open-ended questions.

[10]https://www.qualtrics.com/

For the 10 multiple-choice questions generated by TreeQuestion, participants achieved an average accuracy of 0.40($std. = 0.26$, $N = 96$). Table 5 shows the results of students' scores on both open-ended questions and multiple-choice questions. We performed a repeated measures ANOVA on the scores of students on open-ended questions and multiple-choice questions. Results indicated a significant main effect of *students* ($F(1, 95) = 7.96, p < 0.01$) with no main effect of *questions* ($F(1, 95) = 3.58, p = 0.06$). Such results also suggest that students can achieve comparable learning outcomes in TreeQuestion generated multiple-choice questions in comparison with traditional open-ended questions.

7.2 Investigating Teachers' Experience

To assess whether TreeQuestion could enhance teachers' efficiency in creating appropriate questions for knowledge assessment across different contexts, we conducted a study with participants from various disciplines in computer science. These participants generated both open-ended questions without TreeQuestion and multiple-choice questions with TreeQuestion based on their personal expertise and experience.

7.2.1 Participant Recruitment. We recruited participants through social media (including mailing lists and social groups of professors). 10 course instructors (5 identified as male, 5 identified as female) at an R1 institution participated in the study. All participants had participated in teaching a college-level course and had designed exercise or quiz questions to assess students' knowledge. They were from disciplines within computer science including network security, machine learning, database, etc.

7.2.2 Procedure. Participants were asked to select the concepts they wanted to test within their familiar disciplines in the quiz before the study session. All participants experienced both multiple-choice questions and open-ended questions to compare the experience of generating questions in both forms. During the session, participants were asked the share their screens the whole time. We first got participants' consent, and then gave a demo on how to use the TreeQuestion system.

Participants had 30 minutes to design open-ended questions and multiple-choice questions with or without TreeQuestion. Participants were asked to imagine that they were assigning a quiz in a class and designing quiz questions to assess student knowledge. They were told that both the open-ended questions and multiple-choice questions should test similar knowledge related to the concepts they specified at the beginning of the study. After this, we required the participants to annotate the answers to each question. For open-ended questions, participants were asked to provide detailed explanations to evaluate students' answers. For multiple-choice questions, participants were asked to give the correct options for each question.

At the end of the task, participants were asked to review the automatically generated questions and share their experiences when creating these questions. We specifically asked them to comment on the knowledge coverage, content accuracy, cognitive levels, and perceived efficiency and effectiveness of TreeQuestion. The study session lasted for 45-60 minutes via Zoom. Participants were compensated with a $20 Gift Card.

7.2.3 Findings. Three of our authors watched the user study recordings to label the time spent by the participants on each part. Then, the recordings were transcribed and analyzed using affinity diagrams. The open-ended questions participants generated in our study are demonstrated in Table 6. Participants used the TreeQuestion system to create a set of multiple-choice questions that cover similar ranges of knowledge compared with these open-ended questions.

On average, each participant spent 281.9 seconds ($std. = 115.6$, $N = 10$) to create an open-ended question and 96.7 seconds ($std. = 27.5$, $N = 10$) to create a corresponding set of multiple-choice

Participants	Gender	Discipline	Open-Ended Questions
P1	Male	Data Structures	Suppose you have to design an algorithm for checking if for a given string every opening parentheses has a corresponding closing parentheses. Would you use a stack data structure or a queue data structure for this?
P2	Female	Network Security	What are the strengths and weaknesses of symmetric key cryptography? Give an example of where this type of cryptography is used.
P3	Male	Machine Learning	What is overfitting? How does it differ from underfitting? What are some techniques to address overfitting?
P4	Female	Database	What are the advantages and disadvantages of relational databases?
P5	Female	Game Theory	How does complete information or incomplete information affect a rational person's behavior in a dynamic game?
P6	Male	Operating Systems	What is the major difference between a thread and a process?
P7	Female	Data Structures	Suppose I have a graph and I want to use a search method to find the nearest path from node A to node B. Should I use a depth-first search or a breadth-first search?
P8	Male	Database	What would happen if a database session is terminated before ending the transaction? How to ensure any unwanted result is not obtained?
P9	Female	Discrete Mathematics	What are the connections and differences between group, ring, and field in discrete mathematics?
P10	Male	Machine Learning	Explain the trade-offs between batch gradient descent and stochastic gradient descent.

Table 6. Open-ended questions that participants created during the study cover different disciplines within computer science.

questions with TreeQuestion, which indicates that TreeQuestion only requires around one-third of the time to create a corresponding set of multiple-choice questions compared with open-ended questions. Combined with the estimated data in the previous study, suppose there is a class with 20 students, it will take 2113.9 seconds to create, annotate, and grade all the open-ended questions. However, it will take 96.7 seconds for multiple-choice questions with TreeQuestion since they can be automatically graded, which is around 4.6% of the time needed by open-ended questions. Examples of the matched pairs of open-ended questions and multiple-choice questions are illustrated in 8.

Further, we present our findings from interviews as below.

TreeQuestion provided abundant knowledge that could provide useful insights for question generation. All the participants were satisfied with the coverage of the generated questions and expressed that they would love to use the generated questions for knowledge assessment in

A Open-Ended Questions

Question: *Differentiate overfitting and underfitting in machine learning. Explain potential techniques to address overfitting. Justify with reasons.*
Answer: *Overfitting is the scenario where the model performs well on samples from the training data but is unable to generalize to unseen samples. If a model overfits, it is said to have high variance. Underfitting is the scenario where the model fails to perform well on samples from the training data. If a model underfits, it is said to have high bias. Some techniques to address overfitting include L1 and L2 regularization, using dropout or reducing the number of parameters in the model.*

B Multiple-Choice Questions

Question: *In the field of machine learning, what does overfitting refer to?*
A. Overfitting is when a model accurately predicts new data, leading to poor performance on the training data.
B. Overfitting is when a model learns the noise or random fluctuations in the training data and fails to generalize to new, unseen data.
C. Overfitting is when a model underperforms on both the training and test data due to a lack of complexity in the model.
D. Overfitting is when a model performs well on the training data but fails to predict new data accurately.
Answer: *B*

Question: *Imagine you are working on a classification problem and have built a complex model with millions of parameters. You notice that the model performs extremely well on the training data but fails to generalize to new, unseen data. What technique can you use to address this issue?*
A. Principal component analysis and whitening
B. Independent component analysis
C. Regularization techniques
D. Randomly reshuffling the training and testing data
Answer: *C*

Question: *How can L1 and L2 regularization, along with dropout, be effectively used to address overfitting?*
A. L1 and L2 regularization and dropout are commonly used to increase model complexity and improve model performance.
B. L1 and L2 regularization and dropout are techniques used to randomly remove data points from the training set to reduce the risk of overfitting.
C. L1 and L2 regularization are regularization techniques that add a penalty term to the loss function, encouraging the model to have smaller weights. Dropout is a technique that randomly drops out a fraction of the neurons during training.
D. L1 and L2 regularization, as well as dropout, are techniques used to increase the capacity of the model by adding more layers and neurons.
Answer: *C*

Fig. 8. Examples of paired questions designed and answers labeled by the participants in our evaluation study. Participants were told that both open-ended questions and multiple-choice questions should test similar knowledge related to the concepts they specified before the study.

exams or quizzes. We found that the background knowledge provided by TreeQuestion can enhance the comprehensiveness of questions from different perspectives. By specifying different levels from Bloom's Taxonomy in the Explore stage, P9 commented that "*The content generated by TreeQuestion is comprehensive enough for me to use. I do not think TreeQuestion missed any important points.*" Some participants held the view that TreeQuestion could generate knowledge that may be challenging for them to think of during a short period. P4 also mentioned the benefits of TreeQuestion by saying that "*I can obtain extensive background knowledge without initiating too much information. This approach also saves me a significant amount of time by eliminating the need to incorporate external content, such as textbooks, into the system.*" In the meantime, the human-AI collaborative approach also motivated participants to provide additional information that could prove beneficial for the generation of questions. P6 told us that "*There are potential clues and concepts to help me come up with more relevant knowledge.*"

TreeQuestion produced rich distractor choices that could bring great convenience. Most participants expressed that providing different types of distractors based on question templates would be very helpful for them. We found that TreeQuestion could enhance the quality of generated options for various reasons. Participants noted that the interactive interface in the Validate stage enabled them to correct inaccuracies in knowledge generated by large language models. For example, P3 efficiently rectified a mistake made by the models, which incorrectly stated that "outfitting" means failing to generalize to training data instead of test data. Although some participants acknowledged that distractors may not be useful in all cases, they still value the ability to control the distractors before generating questions. By specifying the templates for option generation, participants thought that they could select the options that are most relevant to the questions. They witnessed several cases where out-of-context information surfaced without proper control over the options, potentially diminishing the quality of the questions. By controlling the options beforehand, TreeQuestion enabled teachers to efficiently craft multiple-choice questions with high-quality distractors. Just as P10 told us, "*I think the most efficient thing for me would be getting a large number of distractors. Writing a question is always hard, and I like the system that helps me write everything quickly to fit into the questions.*"

TreeQuestion provided flexible choices to tune the difficulty of questions. TreeQuestion allowed participants to gain an overview of the difficulty distributions of questions, which could inspire them to design more comprehensive question sets for knowledge assessment. Participants first found that it is very time-saving to just select from a batch of questions at different levels in Bloom's Taxonomy which were generated from large language models. P10 said that "*When I was writing the exam, I was not fully aware of various levels of questions. I like the option of almost tuning the difficulty level of the questions. It would be great to see the full spectrum at first sight.*" While there were instances where the specified keys or distractors did not perfectly align with Bloom's Taxonomy levels, participants felt the generated questions had come close enough to their expectations. In fact, P4 argued that it might not be necessary to have all the questions *precisely* mapped to different levels in Bloom's Taxonomy. Many manually-created questions in exams or quizzes usually involve more than one level from Bloom's Taxonomy. Instead, a system like TreeQuestion that could provide them with questions at various difficulty levels would be helpful enough. Participants also emphasized the value of real scenarios and examples embedded in the questions, expressing satisfaction with the questions generated at higher difficulty levels. They agreed that incorporating plausible distractors with intricate scenarios would necessitate students to engage in more complex thinking processes to arrive at the correct answer. As mentioned by P3, "*The answers are not straightforward, I would say. It would be difficult for everyone to see the differences if they do not have in-depth knowledge.*" Combining all these factors, most of our participants believed that TreeQuestion could provide potential opportunities to generate questions for systematic knowledge assessment.

8 Limitations

8.1 Limitations of Bloom's Taxonomy

TreeQuestion leverages Bloom's Taxonomy to instruct large language models to generate multiple-choice questions involving different cognitive processes. It is worth noting that there are several inherent limitations of Bloom's Taxonomy in the context of question generation. First, in many multiple-choice questions, different cognitive processes such as remembering, understanding, or applying would occur simultaneously. However, due to the hierarchical nature of Bloom's Taxonomy, it would be problematic to assume each category is discrete. Second, new kinds of learning which are required and have been widely included in many real-world tests do not easily fall into Bloom's or the revised cognitive taxonomy. For example, teachers may consider testing the abilities of "learning how to learn", "leadership and interpersonal skills", or "communication skills" in their questions as suggested by previous work [19]. Such kinds of tests have gone beyond the scope of Bloom's Taxonomy. Future research could incorporate other theories, such as Fink's Taxonomy of Significant Learning [19], as alternatives to Bloom's Taxonomy.

8.2 Applicability of the Generated Questions

Although all the participants successfully created multiple-choice questions in the evaluation studies, there are certain limitations of the questions generated by TreeQuestion. First, while TreeQuestion can generate multiple-choice questions spanning various cognitive levels within Bloom's Taxonomy, it is still challenging for large language models to create questions that engage more complicated cognitive processes. It is widely acknowledged that large language models still lack satisfactory capabilities in complex reasoning. In the evaluation study, participants expressed concerns that sometimes the generated questions may require little complex reasoning, which can not fully meet their purposes in examination scenarios. Second, it is possible that the background knowledge generated in TreeQuestion may not be able to satisfy teachers working in all the subjects, especially for those with a lot of professional terms that might be rare in large language models' training data. We anticipate that future systems, potentially utilizing finetuned large language models or employing other finely crafted prompts tailored to specific domains, will enhance the applicability of generated questions.

9 Discussions and Future Work

9.1 Reshaping the Power Imbalance Caused by Generative AI

Cheating risks have become a growing concern in education with the emergence of generative AI. While teachers still need to invest amounts of time in creating and evaluating the questions, students could effortlessly generate human-like answers with generative AI. As an LLM-powered tool for learning outcome assessment, TreeQuestion attempts to remedy such a power imbalance with generative AI. Yet in this teacher-student game, we believe there are still potential opportunities that could transform the power between teachers and students. It is acknowledged that multiple-choice questions, even if well-designed, are still prone to cheating [40]. Multiple-choice questions generated by large language models can also be easily answered by similar models. Yet, leveraging large language models, future work could potentially reduce cheating opportunities or increase cheating cost leveraging personalization and randomization techniques [63]. We envision that future work can potentially make dynamic question generation of teachers much easier when configured with strategies for personalization and randomization. However, in the age of generative AI, we also call for a combination of different strategies including reducing the opportunities, incentives/pressures, and attitudes to cheating to reshape the power imbalance between teachers and students.

9.2 Accelerating Student Participation in Assessment

TreeQuestion aims to enable teachers to assess students' conceptual learning outcomes through structured multiple-choice questions. While TreeQuestion provides an innovative solution for enhancing the efficiency of knowledge assessment for teachers, it may not be optimized for students. However, we believe it might not be necessary to have students go through all the multiple-choice questions to accurately infer students' knowledge state. There is still a large design space for optimizing student participation based on the tree structure of questions. We envision that teachers can also configure future systems with heuristic-based strategies that enable students to skip familiar questions and repeat unfamiliar questions based on their performance, allowing teachers to elicit students' learning outcomes more efficiently [25]. As a result, students can be provided with more targeted multiple-choice questions, thus enhancing their efficiency in participation.

9.3 Designing Scalable Learning with Generative AI

While TreeQuestion provides teachers with an innovative *assessment* solution, it is also equally important to consider the design of *learning*. As suggested by prior work [4], goal-directed practice coupled with targeted feedback is critical to learning. Many studies have shown that feedback interventions improve learning more than non-feedback ones. More frequent feedback leads to more efficient learning because it helps students stay on track. But in reality, creating deliberate practice opportunities with regular feedback necessitates meticulous design and substantial effort from teachers. Since TreeQuestion can offer abundant opportunities for students to engage their knowledge and skills in answering the questions, we envision future systems to provide instant information to students about their performance to guide their future learning. Large language models also have great promise in alleviating the burden on teachers when it comes to generating detailed feedback based on diverse performances. For instance, when students make mistakes, we anticipate that future systems will not only offer detailed explanations for correct answers but also generate helpful tips for improving learning behaviors simultaneously.

10 Conclusion

This paper introduces TreeQuestion, a human-in-the-loop system that assists teachers in generating a set of structured multiple-choice questions to replace knowledge-based open-ended questions. In developing TreeQuestion, we propose a method for generating multiple-choice questions systematically, which can be used to assess students' fine-grained conceptual learning outcomes. We also introduce a novel programming pattern, Explore-Validate-Generate, to control imperfect large language models in creating diverse multiple-choice questions with correct answers. We evaluate TreeQuestion with 96 students and 10 teachers. Our results show that students can achieve similar performance on open-ended questions manually graded by teachers and multiple-choice questions generated by TreeQuestion. Meanwhile, compared with manually-crafted open-ended questions, TreeQuestion could significantly save teachers' efforts in generating and evaluating multiple-choice questions. We estimate that utilizing multiple-choice questions with TreeQuestion in an assumed class of 20 students requires merely 4.6% of the time compared with the time required for evaluating learning outcomes using open-ended questions.

Acknowledgments

We thank our participants for their time and input that shaped this research. We thank Zhiting Hu for his insightful advice for the project. We thank Qiyu Li and Yaqing Yang for their valuable feedback along the process.

References

[1] 2023. GPTZero | AI Content Detector and Writing CAPTCHA for Chat GPT, OpenAI, BARD, Education. https://gptzero.me/.
[2] 2023. ZeroGPT - Chat GPT, Open AI and AI text detector Free Tool. https://www.zerogpt.com/.
[3] Rania Abdelghani, Yen-Hsiang Wang, Xingdi Yuan, Tong Wang, Pauline Lucas, Hélène Sauzéon, and Pierre-Yves Oudeyer. 2023. GPT-3-Driven Pedagogical Agents to Train Children's Curious Question-Asking Skills. *International Journal of Artificial Intelligence in Education* (June 2023). https://doi.org/10.1007/s40593-023-00340-7
[4] Susan A. Ambrose, Michael W. Bridges, Michele DiPietro, Marsha C. Lovett, Marie K. Norman, and Richard E. Mayer. 2010. *How Learning Works: Seven Research-Based Principles for Smart Teaching* (1 ed.). Jossey-Bass.
[5] Lorin W. Anderson and David R. Krathwohl (Eds.). 2001. *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives* (complete ed ed.). Longman, New York.
[6] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or Fake? Learning to Discriminate Machine from Human Generated Text. https://doi.org/10.48550/arXiv.1906.03351 arXiv:1906.03351 [cs, stat].

[7] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST '10)*. Association for Computing Machinery, New York, NY, USA, 313–322. https://doi.org/10.1145/1866029.1866078
[8] Benjamin S. Bloom. 1956. *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain* (2nd edition ed.). Addison Wesley Publishing Company.
[9] Dhawaleswar Rao CH and Sujan Kumar Saha. 2020. Automatic Multiple Choice Question Generation From Text: A Survey. *IEEE Transactions on Learning Technologies* 13, 1 (Jan. 2020), 14–25. https://doi.org/10.1109/TLT.2018.2889100 Conference Name: IEEE Transactions on Learning Technologies.
[10] Yllias Chali and Sadid A. Hasan. 2015. Towards Topic-to-Question Generation. *Computational Linguistics* 41, 1 (March 2015), 1–20. https://doi.org/10.1162/COLI_a_00206 Place: Cambridge, MA Publisher: MIT Press.
[11] Chegg.com. 2023. An example Open-ended question in an Economics assignment. https://www.chegg.com/homework-help/questions-and-answers/9-explain-difference-gross-domestic-product-gdp-gross-national-product-gnp--provide-exampl-q52470316. (Accessed on 06/20/2023).
[12] Chegg.com. 2023. Explain the difference between hashing and encrypt... | Chegg.com. https://www.chegg.com/homework-help/questions-and-answers/explain-difference-hashing-encryption-types-algorithms-use--please-respond-least-two-peers-q83333066. (Accessed on 07/15/2023).
[13] Debby R. E. Cotton, Peter A. Cotton, and J. Reuben Shipway. 2023. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International* 0, 0 (March 2023), 1–12. https://doi.org/10.1080/14703297.2023.2190148 Publisher: Routledge _eprint: https://doi.org/10.1080/14703297.2023.2190148.
[14] Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning* 16, 1 (March 2021), 5. https://doi.org/10.1186/s41039-021-00151-1
[15] Nassim Dehouche. 2021. Plagiarism in the age of massive Generative Pre-trained Transformers (GPT-3). *Ethics in Science and Environmental Politics* 21 (March 2021), 17–23. https://doi.org/10.3354/esep00195
[16] R. Dijkstra, Z. Genç, S. Kayal, and J. Kamps. 2022. Reading Comprehension Quiz Generation using Generative Pre-trained Transformers. (2022). https://dare.uva.nl/search?identifier=a1109043-92d4-4c63-be33-6e238780d3b7 Publisher: AachenCEUR-WS.
[17] Tommy Dreyfus. 1999. Why Johnny Can't Prove. *Educational Studies in Mathematics* 38, 1 (March 1999), 85–109. https://doi.org/10.1023/A:1003660018579
[18] Sabina Elkins, Ekaterina Kochmar, Iulian Serban, and Jackie C. K. Cheung. 2023. How Useful Are Educational Questions Generated by Large Language Models?. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, Ning Wang, Genaro Rebolledo-Mendez, Vania Dimitrova, Noboru Matsuda, and Olga C. Santos (Eds.). Springer Nature Switzerland, Cham, 536–542. https://doi.org/10.1007/978-3-031-36336-8_83
[19] L. Dee Fink. 2013. *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses* (2 ed.). Jossey-Bass.
[20] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds and Machines* 30, 4 (Dec. 2020), 681–694. https://doi.org/10.1007/s11023-020-09548-1
[21] Catherine A. Gao, Frederick M. Howard, Nikolay S. Markov, Emma C. Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T. Pearson. 2023. Comparing scientific abstracts generated by ChatGPT to real abstracts with detectors and blinded human reviewers. *npj Digital Medicine* 6, 1 (April 2023), 1–5. https://doi.org/10.1038/s41746-023-00819-6 Number: 1 Publisher: Nature Publishing Group.
[22] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. https://doi.org/10.48550/arXiv.1906.04043 arXiv:1906.04043 [cs].
[23] Donna George and Susan Svrluga. 2023. In education, AI is changing how teachers develop tests, emails and lessons - The Washington Post. https://www.washingtonpost.com/education/2023/07/13/ai-education-teachers-lesson-plans/. (Accessed on 07/14/2023).
[24] William J. Haynie. 1994. Effects of Multiple-Choice and Short-Answer Tests on Delayed Retention Learning. *Journal of Technology Education* 6, 1 (1994).
[25] Joy He-Yueya and Adish Singla. 2021. *Quizzing Policy Using Reinforcement Learning for Inferring the Student Knowledge State*. Technical Report. International Educational Data Mining Society. https://eric.ed.gov/?id=ED615585 Publication Title: International Educational Data Mining Society ERIC Number: ED615585.
[26] Michael Heilman and Noah A. Smith. 2010. 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*. Association for Computational Linguistics, Los Angeles, California, 609–617. https://aclanthology.org/N10-1086

[27] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rygGQyrFvH

[28] Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic Detection of Machine Generated Text: A Critical Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2296–2309. https://doi.org/10.18653/v1/2020.coling-main.208

[29] Qinjin Jia, Jialin Cui, Yunkai Xiao, Chengyuan Liu, Parvez Rashid, and Edward F. Gehringer. 2021. ALL-IN-ONE: Multi-Task Learning BERT models for Evaluating Peer Assessments. https://doi.org/10.48550/arXiv.2110.03895 arXiv:2110.03895 [cs].

[30] Kayla Jimenez. 2023. How AI detection tool spawned a false cheating case at UC Davis. https://www.usatoday.com/story/news/education/2023/04/12/how-ai-detection-tool-spawned-false-cheating-case-uc-davis/11600777002/. (Accessed on 06/20/2023).

[31] Hyoungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–28. https://doi.org/10.1145/3613904.3642349

[32] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stepha Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* 103 (April 2023), 102274. https://doi.org/10.1016/j.lindif.2023.102274

[33] Michael R King and ChatGPT. 2023. A conversation on artificial intelligence, chatbots, and plagiarism in higher education. *Cellular and Molecular Bioengineering* 16, 1 (2023), 1–2.

[34] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science* 36, 5 (July 2012), 757–798. https://doi.org/10.1111/j.1551-6709.2012.01245.x

[35] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education* 30, 1 (March 2020), 121–204. https://doi.org/10.1007/s40593-019-00186-y

[36] Jill H. Larkin and Herbert A. Simon. 1987. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science* 11, 1 (1987), 65–100. https://doi.org/10.1111/j.1551-6708.1987.tb00863.x _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1551-6708.1987.tb00863.x.

[37] Jeri L. Little and Elizabeth Ligon Bjork. 2015. Optimizing multiple-choice tests as tools for learning. *Memory & Cognition* 43, 1 (Jan. 2015), 14–26. https://doi.org/10.3758/s13421-014-0452-8

[38] Xinyi Lu, Simin Fan, Jessica Houghton, Lu Wang, and Xu Wang. 2023. ReadingQuizMaker: A Human-NLP Collaborative System that Supports Instructors to Design High-Quality Reading Quiz Questions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. https://doi.org/10.1145/3544548.3580957

[39] Connie Malamed. 2023. Writing Multiple Choice Questions For Higher Order Thinking. https://theelearningcoach.com/elearning_design/higher-order-multiple-choice-questions/. (Accessed on 07/17/2023).

[40] Sathiamoorthy Manoharan. 2017. Personalized Assessment as a Means to Mitigate Plagiarism. *IEEE Transactions on Education* 60, 2 (May 2017), 112–119. https://doi.org/10.1109/TE.2016.2604210 Conference Name: IEEE Transactions on Education.

[41] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. https://doi.org/10.48550/arXiv.2301.11305 arXiv:2301.11305 [cs].

[42] Steven Moore, Huy A. Nguyen, Norman Bier, Tanvi Domadia, and John Stamper. 2022. Assessing the Quality of Student-Generated Short Answer Questions Using GPT-3. In *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption (Lecture Notes in Computer Science)*, Isabel Hilliger, Pedro J. Muñoz-Merino, Tinne De Laet, Alejandro Ortega-Arranz, and Tracie Farrell (Eds.). Springer International Publishing, Cham, 243–257. https://doi.org/10.1007/978-3-031-16290-9_18

[43] Yasuhiro Ozuru, Stephen Briner, Christopher A. Kurby, and Danielle S. McNamara. 2013. Comparing comprehension measured by multiple-choice and open-ended questions. *Canadian Journal of Experimental Psychology / Revue canadienne de psychologie expérimentale* 67, 3 (2013), 215–227. https://doi.org/10.1037/a0032918 Place: US Publisher: Educational Publishing Foundation.

[44] Yena Park, Geoffrey T LaFlair, Yigal Attali, Andrew Runge, and Sarah Goodwin. 2022. *Interactive reading—the duolingo english test.* Technical Report. Duolingo Research Report.

[45] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence Level Training with Recurrent Neural Networks. https://doi.org/10.48550/arXiv.1511.06732 arXiv:1511.06732 [cs].

[46] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1 (ICER '22, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 27–43. https://doi.org/10.1145/3501385.3543957

[47] Shravya Bhat, Huy Nguyen, Steven Moore, John Stamper, Majd Sakr, and Eric Nyberg. 2022. Towards Automated Generation and Evaluation of Questions in Educational Domains. (July 2022). https://doi.org/10.5281/ZENODO.6853085 Publisher: Zenodo.

[48] Anjali Singh, Christopher Brooks, Xu Wang, Warren Li, Juho Kim, and Deepti Pandey. 2024. Bridging Learnersourcing and AI: Exploring the Dynamics of Student-AI Collaborative Feedback Generation. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*. 742–748. https://doi.org/10.1145/3636555.3636853 arXiv:2311.12148 [cs].

[49] Kathrin F. Stanger-Hall. 2012. Multiple-Choice Exams: An Obstacle for Higher-Level Thinking in Introductory Science Classes. *CBE—Life Sciences Education* 11, 3 (Sept. 2012), 294–306. https://doi.org/10.1187/cbe.11-11-0100 Publisher: American Society for Cell Biology (lse).

[50] Dannelle D. Stevens and Antonia Levi. 2005. *Introduction to rubrics: an assessment tool to save grading time, convey effective feedback, and promote student learning / Dannelle D. Stevens, Antonia Levi* (1st ed ed.). Stylus Pub, Sterling, Va.

[51] Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and Position-aware Neural Question Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3930–3939. https://doi.org/10.18653/v1/D18-1427

[52] Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive Generation of Long Text with Pretrained Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4313–4324. https://doi.org/10.18653/v1/2021.naacl-main.341

[53] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship Attribution for Neural Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 8384–8395. https://doi.org/10.18653/v1/2020.emnlp-main.673

[54] Harvard University. 2023. Taxonomies of Learning | Derek Bok Center, Harvard University. https://bokcenter.harvard.edu/taxonomies-learning. (Accessed on 07/19/2023).

[55] Cunxiang Wang, Boyuan Zheng, Yuchen Niu, and Yue Zhang. 2021. Exploring Generalization Ability of Pretrained Language Models on Arithmetic and Logical Reasoning. https://doi.org/10.48550/arXiv.2108.06743 arXiv:2108.06743 [cs].

[56] Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards Process-Oriented, Modular, and Versatile Question Generation that Meets Educational Needs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 291–302. https://doi.org/10.18653/v1/2022.naacl-main.22

[57] Xu Wang, Carolyn Rose, and Ken Koedinger. 2021. Seeing Beyond Expert Blind Spots: Online Learning Design for Scale and Quality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445045

[58] Xu Wang, Srinivasa Teja Talluri, Carolyn Rose, and Kenneth Koedinger. 2019. UpGrade: Sourcing Student Open-Ended Solutions to Create Scalable Learning Opportunities. In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale (L@S '19)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3330430.3333614

[59] Zichao Wang, Jakob Valdez, Debshila Basu Mallick, and Richard G. Baraniuk. 2022. Towards Human-Like Educational Question Generation with Large Language Models. In *Artificial Intelligence in Education (Lecture Notes in Computer Science)*, Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova (Eds.). Springer International Publishing, Cham, 153–166. https://doi.org/10.1007/978-3-031-11644-5_13

[60] Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. Consistency of a Recurrent Language Model With Respect to Incomplete Decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 5553–5568. https://doi.org/10.18653/v1/2020.emnlp-main.448

[61] Chris Westfall. 2023. Educators Battle Plagiarism As 89% Of Students Admit To Using OpenAI's ChatGPT For Homework. https://www.forbes.com/sites/chriswestfall/2023/01/28/educators-battle-plagiarism-as-89-of-students-admit-to-using-open-ais-chatgpt-for-homework/?sh=68f598d7750d. (Accessed on 07/13/2023).

[62] Johannes Wheeldon. 2011. Is a Picture Worth a Thousand Words? Using Mind Maps to Facilitate Participant Recall in Qualitative Research. *The Qualitative Report* 16, 2 (March 2011), 509–522. https://doi.org/10.46743/2160-3715/2011.1068
[63] Brooke L. Whisenhunt, Christie L. Cathey, Danae L. Hudson, and Lydia M. Needy. 2022. Maximizing learning while minimizing cheating: New evidence and advice for online multiple-choice exams. *Scholarship of Teaching and Learning in Psychology* 8, 2 (2022), 140–153. https://doi.org/10.1037/stl0000242 Place: US Publisher: Educational Publishing Foundation.
[64] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. PromptChainer: Chaining Large Language Model Prompts through Visual Programming. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3491101.3519729
[65] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–22. https://doi.org/10.1145/3491102.3517582
[66] Xiaomeng Xu, Sierra Kauer, and Samantha Tupy. 2016. Multiple-choice questions: Tips for optimizing assessment in-seat and online. *Scholarship of Teaching and Learning in Psychology* 2, 2 (2016), 147–158. https://doi.org/10.1037/stl0000062 Place: US Publisher: Educational Publishing Foundation.
[67] Iman Yeckehzaare, Tirdad Barghi, and Paul Resnick. 2020. QMaps: Engaging Students in Voluntary Question Generation and Linking. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376882
[68] Shiyue Zhang and Mohit Bansal. 2019. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2495–2509. https://doi.org/10.18653/v1/D19-1253
[69] Alex Y. Zheng, Janessa K. Lawhorn, Thomas Lumley, and Scott Freeman. 2008. Application of Bloom's Taxonomy Debunks the "MCAT Myth". *Science* 319, 5862 (Jan. 2008), 414–415. https://doi.org/10.1126/science.1147852 Publisher: American Association for the Advancement of Science.
[70] Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Question-type Driven Question Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6032–6037. https://doi.org/10.18653/v1/D19-1622
[71] Mengxiao Zhu, Ou Lydia Liu, and Hee-Sun Lee. 2020. The effect of automated feedback on revision behavior and learning gains in formative assessment of scientific argument writing. *Computers & Education* 143 (Jan. 2020), 103668. https://doi.org/10.1016/j.compedu.2019.103668

A Example Prompts

In this section, we provide more example prompts targeted at different levels in Bloom's Taxonomy for background knowledge generation in Table 7. All the explanations of different levels in Bloom's Taxonomy are extracted from existing literature [5]. All the inputs listed in Table 7 are used to generate the learning objectives related to the concept. The outputs will then be used to generate the background knowledge as described in section 5.

Received July 2023; revised January 2024; accepted March 2024

Input	`Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "remember" level in Bloom's Taxonomy. Each statement should start with a verb.`
	`The definitions of the "remember" level in Bloom's Taxonomy are: "Remember" means retrieving relevant knowledge from long-term memory. Cognitive processes in the category of "remember" include recognizing and recalling.`
	`Recognizing means locating knowledge in long-term memory that is consistent with the presented material. Recalling means retrieving relevant knowledge from long-term memory.`

(a) This example prompt facilitates the generation of background knowledge at the "remember" level for "symmetric encryption." The first part instructs LLM to produce statements about the target concept at the "remember" level in Bloom's taxonomy. The second part provides definitions of Bloom's Taxonomy levels and the cognitive processes involved. Cognitive processes in the category of "***remember***" include ***recognizing*** and ***recalling***. Both the definitions and hierarchical relationships are adapted from [5].

Input	`Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "understand" level in Bloom's Taxonomy. Each statement should start with a verb.` `The definitions of the "understand" level in Bloom's Taxonomy are: "Understand" means construct meaning from instructional messages, including oral, written, and graphic communication. Cognitive processes in the category of "understand" include interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining.` `Interpreting means changing from one form of representation to another. Exemplifying means finding a specific example or illustration of a concept or principle. Classifying means determining that something belongs to a category. Summarizing means abstracting a general theme or major point(s). Inferring means drawing a logical conclusion from the presented information. Comparing means detecting correspondences between two ideas, objects, and the like. Explaining means constructing a cause-and-effect model of a system.`

(b) Example prompts at the "***understand***" level. Related cognitive processes include ***interpreting***, ***exemplifying***, ***classifying***, ***summarizing***, ***inferring***, ***comparing***, and ***explaining***.

Input	`Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "apply" level in Bloom's Taxonomy. Each statement should start with a verb.`
	`The definitions of the "apply" level in Bloom's Taxonomy are: "Apply" means carrying out or using a procedure in a given situation. Cognitive processes in the category of "apply" include executing and implementing.`
	`Executing means applying a procedure to a familiar task. Implementing means applying a procedure to an unfamiliar task.`

(c) Example prompts at the "***apply***" level. Related cognitive processes include ***executing*** and ***implementing***.

Input	Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "analyze" level in Bloom’s Taxonomy. Each statement should start with a verb. The definitions of the "analyze" level in Bloom’s Taxonomy are: "Analyze" means breaking material into its constituent part and determine how the parts relate to one another and to an overall structure or purpose. Cognitive processes in the category of "analyze" include differentiating, organizing, and attributing. Differentiating means distinguishing relevant from irrelevant parts or important from unimportant parts of the presented material. Organizing means determining how elements fit or function within a structure. Attributing means determining a point of view, bias, values, or intent underlying the presented material.

(d) Example prompts at the “***analyze***” level. Related cognitive processes include ***differentiating***, ***organizing*** and ***attributing***.

Input	Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "evaluate" level in Bloom’s Taxonomy. Each statement should start with a verb. The definitions of the "evaluate" level in Bloom’s Taxonomy are: "Evaluate" means make judgments based on criteria and standards. Cognitive processes in the category of "evaluate" include checking and critiquing. Checking means detecting inconsistencies or fallacies within a process or product; determining whether a process or product has internal consistency; and detecting the effectiveness of a procedure as it is being implemented. Critiquing means detecting inconsistencies between a product and external criteria, determining whether a product has external consistency; and detecting the appropriateness of a procedure for a given problem

(e) Example prompts at the “***evaluate***” level. Related cognitive processes include ***checking*** and ***critiquing***.

Input	`Generate some statements regarding the learning objectives of symmetric encryption in the area of cybersecurity. The statements should cover the "create" level in Bloom's Taxonomy. Each statement should start with a verb.`
	`The definitions of the "create" level in Bloom's Taxonomy are:`
	`"Create" means putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure. Cognitive processes in the category of "create" include generating, producing, and planning.`
	`Generating means coming up with alternative hypotheses based on criteria. Producing means inventing a product. Planning means devising a procedure for accomplishing some task.`

(f) Example prompts at the "***create***" level. Related cognitive processes include ***generating***, ***producing***, and ***planning***.

Table 7. Example prompts with the concept "symmetric encryption" in the field of "cybersecurity" targeted at the different levels in Bloom's Taxonomy in the Explore stage. With the same concepts and fields, we only change the detailed explanations of different levels and related cognitive processes in Bloom's Taxonomy if users specify different levels.