An LLM-Powered Adaptive Practicing System

Md Rayhan Kabir[1], Fuhua Lin[1]

[1]Athabasca University, Alberta, Canada

Abstract

The deployment of artificial intelligence in online education systems has become very popular in recent years. Recently published large language models tend to have a huge application potential for developing intelligent online education systems. In this paper, we propose a personalized online practicing system where we will use ChatGPT to generate questions and provide appropriate feedback for the student's responses to the questions. We designed a prompt generator and a text analyzer to send prompts and process the responses by ChatGPT. We also integrated an adaptive feedback mechanism to determine whether a student has mastered a topic or not. We developed a prototype of our proposed system. From initial experiments with a given topic, we found that ChatGPT could accurately and effectively generate questions and feedback to enable adaptive practice.

Keywords

Adaptive Practicing, Large Language model, AI in education, ChatGPT

1. Introduction

In the last few decades, the popularity of online learning has been increasing tremendously [1]. One of the main advantages of online learning is that it removes many limitations of the traditional classroom method, such as it removes the geographical boundary as students from all over the world can attend any online learning session [2]. It also has vast potential to implement an adaptive learning system where the system can provide personalized exercise and feedback. As we know the rate and the pace of knowledge gain is different for different students, the traditional education system fails to provide any personalized feedback or opportunities for practice to the students. But in the case of online education system [3] there are more opportunities to design feedback that supports each learner.

There are several parts of a personalized adaptive practicing system. When a student starts practicing a topic on the system, the system should identify the knowledge level of the students. This is a very important part and a large number of researchers are working on identifying the knowledge level of the student. To provide personalized sets of practice questions, the system should trace the knowledge level of the students. While students start practicing, practice question sequence should be a personalized sequence for each of the students as different students learn at different paces. Now while practicing the questions, the system should also keep track whether the student has mastered a topic or not. Several researchers have come forward with several approaches to detect the mastery level of the student on a topic. Another

Empowering Education with LLMs – the Next-Gen Interface and Content Generation, July 7, 2023, Tokyo, Japan
rkabir@athabascau.ca (M. R. Kabir); oscarl@athabascau.ca (F. Lin)
0009-0002-6898-1805 (M. R. Kabir); 0000-0002-5876-093x (F. Lin)

part of the system is the practice question itself. In the current situation, in the context of adaptive learning systems, researchers prepare their own question bank and let the system decide the sequence of the questions according to the knowledge level and learning rate of the student [4]. This is a time consuming task and after preparing each question, the difficulty level and other parameters of the questions also has to be set manually and there is a scope of mistakes in several points.

Recently Large Language Models (LLM) have shown great potential in many different tasks and they have also been used in the case of preparing questions with proper prompts and parameters [5]. From literature we can see there are educational intelligent chat-bots which can be very helpful in planning learning activities of the students [6]. Most of them were built with the help of previous LLM models like GPT3 and others [7]. Recently with the release of more advanced LLM models that power ChatGPT and GPT4, building a chatbot personalized to a specific task has been way easier. With recent LLM models it is also possible to generate questions with proper prompt and necessary parameters.

In this paper, we are proposing a personalized online practicing system where the questions, answers and feedback are generated by ChatGPT. In the proposed system we also integrate a few other modules like tracing the knowledge level of the student, detecting mastery of the students at different difficulty levels and preparing personalized feedback using the Multi Armed Bandit model. We also develop a prototype of the system to test the feasibility of our proposed system and there we observe our proposed model performs well in providing personalized practice question sequences following the knowledge level of the students.

2. Related Work

In the last few years many researchers came forward to develop personalized adaptive learning systems with state of the art technologies. Kazi [14] presented an intelligent learning environment where he included different essential components like communication module, pedagogical model and expert module. He proposed a client-server based architecture where all the information will be passed and processed in the server side module whereas the student interaction part will be handled in the client side module. Author also showed the reasoning and appropriate usage of using mLearning and intelligent tutoring in an online learning system. Sharma et al. [15] proposed an intelligent tutoring system which is designed for teaching programming to the students in online programming courses. Their model aims to enhance the performance of the student by providing prompt feedback and guidance to the students. Author showed their system is way more intelligent than the traditional learning systems. Among the other important work on the intelligent tutoring systems, work of Lin et al. [16], Munshi et al. [17], Rus et al. [18] are noteworthy. These works provide valuable insight and guidance to develop an intelligent tutoring system which enhances the learning of the students in an intelligent system.

One important part of the system is to sequence the question pattern to develop an adaptive learning sequence. This sequencing is hard as the questions are selected from a definite size of question bank and with different parameters. From the literature we can see different versions of the Multi-armed Bandit algorithm [8] has been used named Upper Confidence Bound(UCB)[8], adversarial bandit [9], contextual bandits [10], stochastic bandits [11], recovering bandits [12] etc for sequencing questions with different goals like assessment or exercise. However, in our proposed model we divert this heavy part of the system on chatGPT as chatGPT can produce an unlimited number of questions with different difficulty from a context or topics. In our system we will prompt chatGPT using different parameters. Another important part of our study is to detect mastery of a student for different topics or learning objectives (LO). By mastery we refer to a good proficiency or expertise in a specific learning objective that is to have a high level understanding of the key concepts, principles and skills on that LO [13]. To determine the mastery of a student in a model, a mastery criteria is needed to be fixed which will determine whether a student mastered an LO or not. Many researchers defined mastery criteria in different ways in their work. Doroudi [13] reinterpreted two prominent mastery learning heuristics as model-based algorithms for the mastery criteria. From his study, he showed that heuristics can be proven as optimal policies for some Bayesian knowledge tracing models. These heuristics also offer insight into their effective learning assumptions. Following their studies several researchers put their effort to define a good mastery criteria while evaluating the knowledge gain of students or simulated student models. Work of Corbett and Anderson [19], Pavlik et al. [20], Pelánek [21] provides a huge contribution in the field of defining mastery criteria for a student. Kelly et al. makes a comparative analysis on the existing mastery criteria specifically defining mastery using knowledge tracing (KT) and NCCR (N Consecutive Correct Responses). From his analysis he concluded using a higher threshold for NCCR algorithm provides better performance in terms of accuracy compared to lower threshold for NCCR or KT. Another group of researchers Pelanek and Rihat [23] also performed a comparative analysis on different mastery criterias. Their result indicates that the setting of thresholds and data sources used for mastery decision has more impact on the result than the choice of learner modeling techniques. Our study is significantly influenced by the mentioned research works because while designing our student model and mastery criteria, we follow the procedure and findings of the mentioned research.

3. Proposed model

In our proposed system there are several modules named Database, SQL server, User Interface, Prompt generator, text processor and mastery detection module. In this section we will provide a brief description of each of the modules. The overall architecture of our proposed system is shown in Figure 1.

3.1. Database and sql server

The database of our proposed system is a simple mongodb database which is deployed in the cloud and all the necessary information is stored in the database. The login information for the users (both student and instructor), the topics and learning objectives, the questions that are answered by the students, the feedback generated from the ChatGPT, performance of each student and each topic and all other information are stored in the database as a student model. The information about a topic and the related learning objectives are also stored as domain models. The SQL server will deal with the database to store data and to retrieve data from

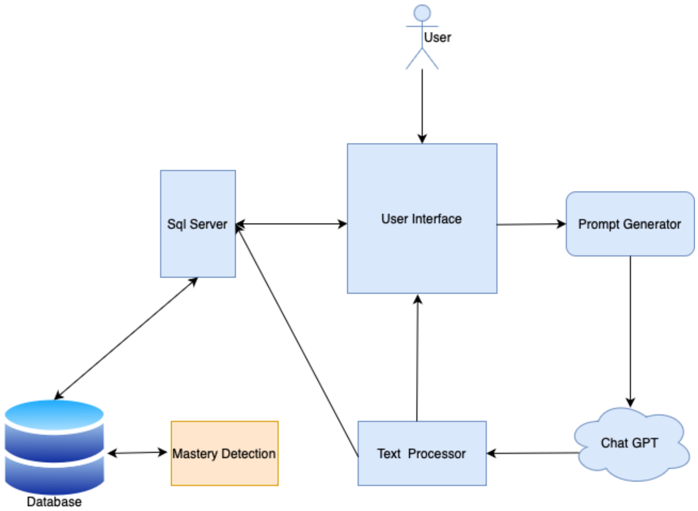


Figure 1: Architecture of the proposed system

the database. Some common operations on the database by the sql server is to store input information by the instructor or students, store the questions and feedback by ChatGPT, retrieve information to show the performance of the student etc.

3.1.1. Student Model

For our experiment the student model is straight forward. In the database we create student instances to store necessary information like personal details, the question the student attempted, answer of the question, feedback on answer, the performance of the student and whether the student mastered a topic or not. The other modules of the system will interact with the sql server to update the information of a student in the database

3.1.2. Domain Model

In our proposed architecture we consider domain D as a set of topics that a student has to learn, practice and master. In each of the topics there will be several learning objectives (LO) which a student has to achieve to master that topic. These LO will be set by the instructor and will remain hidden from the student. Among the LOs there will be a prerequisite relationship and a directed graph can be used to represent the relationship [24]. The questions will be generated from the very first LO which does not have any prerequisite and gradually move forward. A student will only master a topic if that student can master all the LOs of that topic.

3.2. User interface

An user interface will be placed in the system for the interaction of the users with the systems. In the user interface the first page will be a login page from where both the student and the instructor can login with their login credential. After login, students will be able to see different

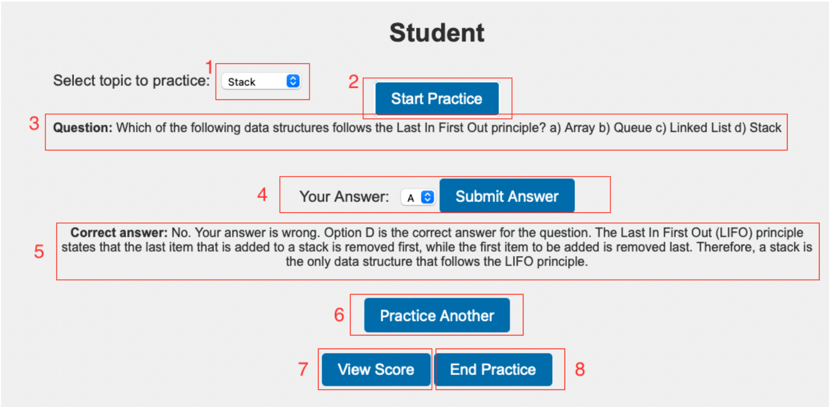


Figure 2: Sample student interface

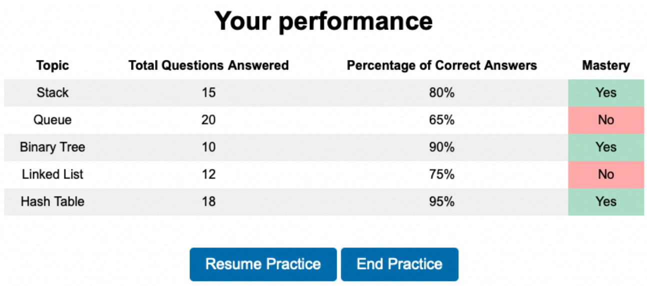
options for practicing or see his performance and the feedback on each of the topics or learning objectives. There are several parts of the student interface. We will explain them here.

3.2.1. Student Interface

A typical student interface is shown in Figure in 2. For better explanation we put a number for each of the elements in red which are not part of the interface. After login the student will be in this interface and will see the element 1, 2, 7 and 8 will be visible to the student. Element 1 is a drop-down list of the topics that the instructor wants a student to practice. The topics defined by the instructor will be visible here and the student will be able to select a topic from here. Upon selecting the topic and clicking on the start practice button a prompt will be sent to ChatGPT and the question generated by the ChatGPT will be shown to the student in element 3 along with a button to submit the answer. Students can select the answer and submit the answer. Then the answer will be sent again to the ChatGPT as a prompt and the feedback of the student answer will be shown in box 5. The question, student's answer and the feedback will be stored to the database along with the information whether the student answered the question right or wrong. Practicing another button will clear the screen and start form element 1 and 2 again. The view score function will show the number of questions attempted, percentage of Correct answer and whether the student mastered a topic or not. The mastery will be detected by the mastery detection algorithm and will be stored in the database. One sample of the view score page is shown in Figure 3. The end practice button will end the practice and logout the student.

3.2.2. Instructor Interface

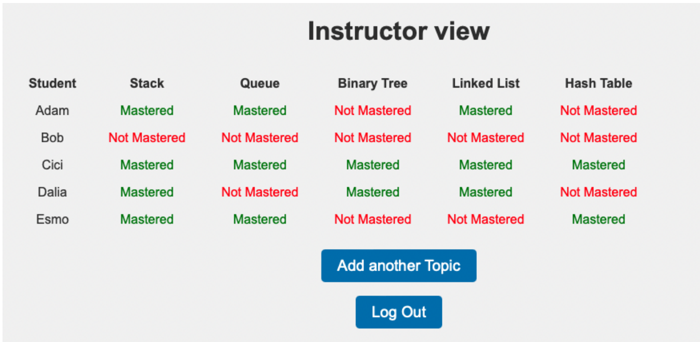
Compared to the student interface, the instructor interface is quite simple. In the instructor interface Instructor will be able to see the performance of each student in each of the topics. The instructor will also be able to add any specific topic to practice. A sample of the instructor interface is shown in Figure 4. In addition to seeing the real time performance of the student, the instructor will also be able to add more topics for the students. This added topic will be



Your performance

Topic	Total Questions Answered	Percentage of Correct Answers	Mastery
Stack	15	80%	Yes
Queue	20	65%	No
Binary Tree	10	90%	Yes
Linked List	12	75%	No
Hash Table	18	95%	Yes

Figure 3: Sample performance for a student



Instructor view

Student	Stack	Queue	Binary Tree	Linked List	Hash Table
Adam	Mastered	Mastered	Not Mastered	Mastered	Not Mastered
Bob	Not Mastered	Not Mastered	Not Mastered	Not Mastered	Not Mastered
Cici	Mastered	Mastered	Mastered	Mastered	Mastered
Dalia	Mastered	Not Mastered	Mastered	Mastered	Not Mastered
Esmo	Mastered	Mastered	Not Mastered	Not Mastered	Mastered

Figure 4: Instructor Interface

added to the database and will be shown to the students. For the prototype we kept this simple and kept the option to add more topics. Further options like add a new course, add topics for any individual students can be added to make it a full scale model and more dynamic. In future we want to implement such modules.

3.3. Prompt generator

The prompt generator is the module which takes input from the system and interacts with the ChatGPT. To generate a question and get feedback this prompt generator generates a proper prompt for ChatGPT and sends it to the ChatGPT. To generate a question at first, the student selects the topic and clicks on the start practice button. Upon clicking, the system sends the selected topic to the prompt generator. The prompt generator create the prompt in the following fashion:

```
"generate an MCQ question on topic ${topic} with learning objective ${LO}"
```

Here, the variable topic is the topic that the student has been selected for. The learning objective comes from the database according to the domain model provided by the instructor. A mastery detection model defines whether the student mastered the topic or not. If the student masters a topic for a given LO, a question from the next LO is generated and when a student masters all the LOs, then the system shows the student is mastered in that topic. This detail is kept hidden from the student. When a student selects an answer and submit, again another prompt is send to the ChatGPT with the previous question and the answer of the student in the following format:

```
"Is option ${answer} correct for question ${question}? provide necessary explanation."
```

Upon receiving the response from ChatGPT, the response is sent to the text processor as the response can be incomplete or can contain the answer.

3.4. Text Processor

The text processor performs two actions. One is receiving the response from ChatGPT for both the question and feedback. When it receives the question, it checks if the question is complete, if the question contains the answer etc. The text processor actually works as a filter. If the question is incomplete it discards the response and takes necessary steps to send the same prompt again to ChatGPT. If the response contains a complete answer it discards only the answer part and sends the question to the user interface to show it to the student. When the text processor receives feedback for an answer, it processes the response and checks whether the response is positive or negative. For this we incorporate a simple positive-negative detecting algorithm to determine whether the answer of the student was correct or not. Then the processor sends a query to the query server so that the server can update the performance of the student along with storing the question, answer and feedback in the database. The text processor also sends the feedback to the user interface so that it can be seen by the student.

3.5. Mastery Detection

One of the most important parts of our system is mastery detection. This is the module that traces the knowledge level of the student and declares whether the student has mastered a topic or not. We use this module in our system because while practicing a topic, both the student and the teacher might want to know how well the practice is going or whether the practice is enough to go for the evaluation test. From the literature we can see recently many researchers worked on mastery detection using different approaches. For our system we aim to use the mastery detection approach proposed by Lin et al. [24]. In their paper they showed their approach can detect the mastery level of a student in a topic efficiently. They used an improved form of top two Thompson sampling algorithms to solve the best arm identification problem in a multi-armed bandit framework. In our system we will make some improvisation of the proposed model as in their model they used graph based domain model and prerequisite relationship among the topics. But for our model initially we are not using the prerequisite or domain model that they used. Moreover we are detecting the mastery of the student for a number of LOs. When a student masters in an LO, a question from the nextLO is asked to that student. When a student masters in all LOs, then the system declares the student is mastered in that topic. This mastery detection module determines whether a student mastered a topic or not.

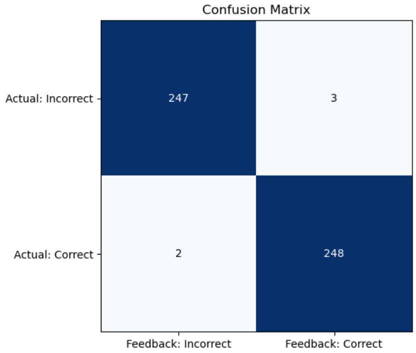


Figure 5: Simulation result

This is the general overview of the architecture and the working procedure of our proposed personalized adaptive practicing system using the Large Language Model. Here we propose the baseline of the system. In this baseline many other options can be introduced for different purposes and can be deployed in a real world learning system.

4. Experiment with the system

We developed the prototype using JavaScript. For our prototype we used 5 different Topics and for each of the topics we used our architecture for generating questions for 100 times, input answers for each of them and generated feedback. That is for each of the Topics we generated 60 questions for our experiment. We use domain experts to evaluate the answer and the feedback. It is to be mentioned that to find out whether the system works properly, we intentionally provided the wrong answer. In Figure 5 we show the confusion matrix where the True positives are the when in the feedback ChatGPT said the answer is correct and the answer was actually correct, true negative means in the feedback ChatGPT said the answer is incorrect and the answer was actually incorrect. False positive means when ChatGPT falsely categorizes incorrect answers as correct, False negative means when ChatGPT falsely identifies correct answers as incorrect.

From the figure we can clearly see that ChatGPT produces the correct feedback in 99% of the cases. For the rest 1% we can deploy a human in the loop to solve this. That is if the student thinks that the feedback that is generated by ChatGPT is wrong, they can send a notification to the instructor to recheck the generated feedback. The instructor will get a notification, check the feedback and take necessary action on that. This is how we can solve the issue of this 1% wrong feedback by ChatGPT.

5. Conclusion and Future work

In this paper, we propose a personalized adaptive practicing system using a Large Language Model. We explain how we can use a Large Language Model (ChatGPT) to generate questions and feedback for different topics. From our experiment we found ChatGPT can generate the questions with the topic name and the difficulty level very efficiently which can save a huge human work hours. In the case of determining the difficulty level we found ChatGPT is more efficient than human. Thus it will be more efficient to generate the questions using ChatGPT. In our adaptive practicing system we propose a mastery detection algorithm which considers different levels of difficult questions for a student to master in a topic. In the previous models, an expert had to prepare a question bank and then the model had to find a personalized sequence. But for our model we do not have to prepare a question bank; rather the ChatGPT produces the questions and feedback, the mastery detection module determines the level of mastery of the student and sends a prompt accordingly. Thus the student get a personalized sequence of practice questions according to their knowledge level.

In the future we want to further add other options for both the student and instructor so that this prototype becomes suitable to deploy in a real life scenario. We also want to work on the proposed mastery detection algorithm. For now we assume our proposed model will perform well based on the previous work on the same algorithm. but in future we want to work more on the simulation and mathematical proof of our proposed mastery detection algorithm and tune it for better performance.

6. Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), Alberta Innovates, and Athabasca University, Canada.

References

[1] Weller, M.: Twenty years of EdTech. Educa. Rev. Online. 53(4), 34–48 (2018)

[2] Arkorful, Valentina, and Nelly Abaidoo. "The role of e-learning, advantages and disadvantages of its adoption in higher education." International journal of instructional technology and distance learning 12.1 (2015): 29-42.

[3] Rafferty, A. N., Brunskill, E., Griffiths, T. L., Shafto, P. (2011). Faster teaching by POMDP planning. In G. Biswas, S. Bull, J. Kay, A. Mitrovic (Eds.), Artificial Intelligence in Education (pp. 280–287). Springer.

[4] Yudelson, Michael V., Kenneth R. Koedinger, and Geoffrey J. Gordon. "Individualized bayesian knowledge tracing models." Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16. Springer Berlin Heidelberg, 2013.

[5] Baidoo-Anu, David, and Leticia Owusu Ansah. "Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning." Available at SSRN 4337484 (2023).

[6] Io, H. N., and C. B. Lee. "Chatbots and conversational agents: A bibliometric analysis." 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2017.

[7] Tack, Anaïs, and Chris Piech. "The AI Teacher test: Measuring the pedagogical ability of Blender and GPT-3 in educational dialogues." arXiv preprint arXiv:2205.07540 (2022).

[8] T. Lattimore and C. Szepesvári, Bandit Algorithms (1st ed.)., Cambridge University Press, 2020.
[9] Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." Machine learning 47 (2002): 235-256.
[10] Littman, Michael L. "Markov games as a framework for multi-agent reinforcement learning." Machine learning proceedings 1994. Morgan Kaufmann, 1994. 157-163.
[11] Thompson, William R. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples." Biometrika 25.3-4 (1933): 285-294.
[12] Pike-Burke, Ciara, and Steffen Grunewalder. "Recovering bandits." Advances in Neural Information Processing Systems 32 (2019).
[13] S. Doroudi, "Mastery Learning Heuristics and Their Hidden Models," in Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, Springer, 2020.
[14] Kazi, Sabbir Ahmed. "VocaTest: An intelligent tutoring system for vocabulary learning using the" mLearning" approach." (2005).
[15] Juffs, Alan, and Benjamin E. Friedline. "Sociocultural influences on the use of a web-based tool for learning English vocabulary." System 42 (2014): 48-59.
[16] Chun Fu Lin, Yu chu Yeh, Yu Hsin Hung, and Ray I Chang. Data mining for providing a personal- ized learning path in creativity: An application of decision trees. Computers Education, 68:199 – 210, 2013.
[17] Anabil Munshi and Gautam Biswas. Personalization in OELEs: Developing a Data- Driven Framework to Model and Scaffold SRL Processes. In International Conference on Artificial Intelligence in Education, pages 354–358. Springer, 2019.
[18] Vasile Rus, Dan Stefanescu, Nobal Ni- raula, and Arthur C Graesser. DeepTutor: towards macro- and micro-adaptive conversational intelligent tutoring at scale. In Proceedings of the 1st ACM conference on Learn- ing@ scale, pages 209–210, 2014.
[19] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge.," User Modeling and User-Adapted Interaction, 4(4), 253–278, 1994.
[20] P. I. Pavlik Jr, H. Cen and K. R. Koedinger, "Performance Factors Analysis–A New Alternative to Knowledge Tracing," in Proceedings of the International Conference on Artificial Intelligence and Education, 2009.
[21] R. Pelánek, "Applications of the Elo rating system in adaptive educational systems," Computers Education, vol. 98, p. 169–179., 2016.
[22] K. Kelly, Y. Wang, T. Thompson and N. Heffernan, "Defining Mastery: Knowledge Tracing Versus N- Consecutive Correct Responses," in Proc. of Educational Data Mining, 2015.
[23] R. Pelanek and J. Rihak, "Experimental Analysis of Mastery Learning Criteria," in Proceedings of UMAP '17, Bratislava, Slovakia, 2017.
[24] F. Lin and S. Silva, "An apporach to Generative Adaptive Feedback for Online Formative Assessment," in proceedings ITS2023: 19th International Conference on Intelligent Tutoring System 2023 Corfu, Greece, June 2-5, 2023.