Empowering Private Tutoring by Chaining Large Language Models

Yulin Chen*
yl-chen21@mails.tsinghua.edu.cn
Shenzhen International Graduate School, Tsinghua University
Shenzhen, China

Ning Ding*
dn97@mail.tsinghua.edu.cn
Department of Electronic Engineering, Tsinghua University
Beijing, China

Hai-Tao Zheng†
zheng.haitao@sz.tsinghua.edu.cn
Shenzhen International Graduate School, Tsinghua Universiy
Pengcheng Laboratory
Shenzhen, China

Zhiyuan Liu†
liuzy@tsinghua.edu.cn
DCST, BNRIST, CollegeAI, Tsinghua University
Beijing, China

Maosong Sun
sms@tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing, China

Bowen Zhou
zhoubowen@tsinghua.edu.cn
Department of Electronic Engineering, Tsinghua University
Beijing, China

Abstract

Artificial intelligence has been applied in various aspects of online education to facilitate teaching and learning. However, few approaches have been made towards a complete AI-powered tutoring system. In this work, we explore the development of a full-fledged intelligent tutoring system based on large language models (LLMs). The proposed system ChatTutor, powered by state-of-the-art LLMs, is equipped with automatic course planning and adjusting, informative instruction, and adaptive quiz offering and evaluation. ChatTutor is decomposed into three inter-connected core processes: *interaction*, *reflection*, and *reaction*. Each process is implemented by chaining LLM-powered tools along with dynamically updated memory modules. To demonstrate the mechanism of each working module and the benefits of structured memory control and adaptive reflection, we conduct a wide range of analysis based on statistical results and user study. The analysis shows the designed processes boost system consistency and stability under long-term interaction and intentional disruptions, with up to 5% and 20% increase in performance respectively. Meanwhile, we also compare the system with scripts from real-world online learning platform and discuss the potential issues unique to LLM-based systems.

CCS Concepts

• **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → **Computer-assisted instruction**; **E-learning**.

*Both authors contributed equally to this research.
†Corresponding authors

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679665

Keywords

Large Language Models,Intelligent Tutoring System,Memory Mechanism,Adaptive Reflection

ACM Reference Format:
Yulin Chen, Ning Ding, Hai-Tao Zheng, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2024. Empowering Private Tutoring by Chaining Large Language Models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679665

1 Introduction

Online education, along with artificial intelligence (AI) technology, brought the aspiration of personalized tutoring within reach [4]. AI has been used to assist education in multiple aspects, ranging from adaptive content recommendation [9], automatic performance evaluation [24, 29], to personalized instruction and dynamic feedback [3, 13, 18, 22, 44]. Although a few early approaches have been made towards a dialogue-based intelligent tutoring system (ITS) [21, 41], most of them are domain-specific and focus primarily on guiding the users to solve a pre-defined problem. Nevertheless, a more ultimate exploration lies in the pursuit of a full-fledged AI-driven tutoring system with greater flexibility and generalizability that teaches in a systematic and consistent manner on a much broader range of knowledge.

While previous works often employ diverse techniques jointly, including learner style classification [32], data mining [14], Bayesian learning [22], etc, the recent emergence of large language models (LLMs) [1, 6, 45, 46], like ChatGPT [34], has broadened our imagination on new designs of intelligent tutoring systems. LLMs impressed people firstly with the ability to generate and transform information following human instructions, then with the potential in task planning and tool usage. Given their extraordinary ability in providing detailed and accessible content and summarizing information, LLMs become great sources for consulting a wide range of knowledge. Additionally, LLMs have shown remarkable capabilities in engaging in multi-turn dialogues, which makes them well-suited for tutoring applications where there is a back-and-forth exchange

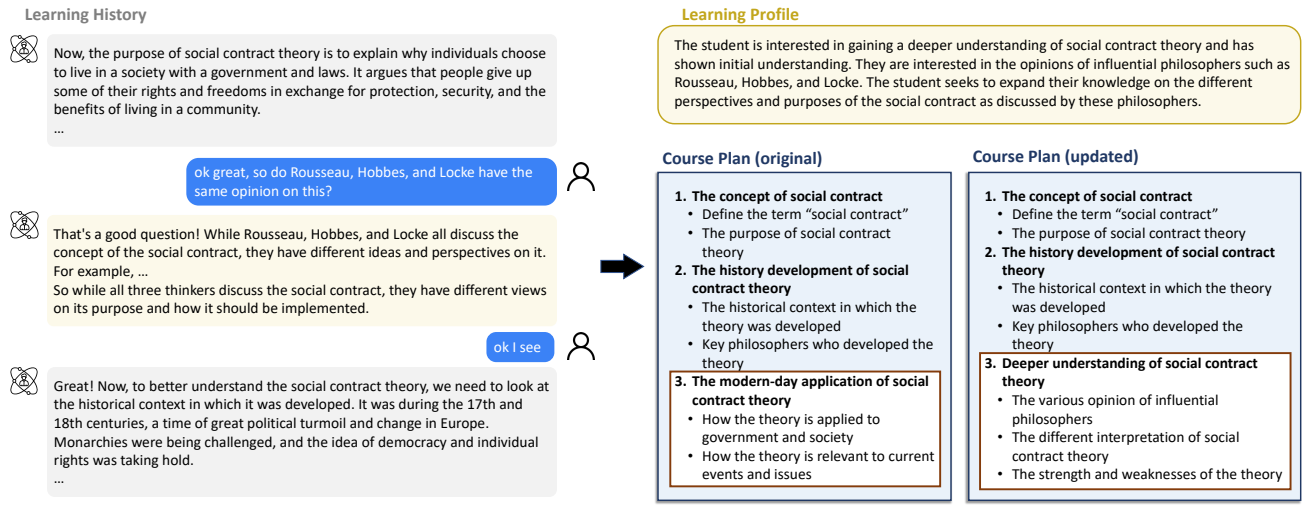

Figure 1: An example of the learning progress. The left side is the user interface directly controlled by the interaction process. The right side is the backend memory changes brought by reflection and reaction processes.

with a student. Although challenges like hallucination remain [2], LLMs can adapt to expertise in specific domains and pedagogical strategies with further fine-tuning.

In this work, we explore the potential of employing generative large language models to build a full-fledged dialogue-based personalized tutoring system. One specialty about an ITS compared to other LLM-powered agents is that, education is a long-term cooperative process accomplished by AI and human users jointly. A well-designed tutoring system should correctly infer the human user's mental state to achieve adaptive teaching, and meanwhile, the user should be informed about the learning progress in order to cooperate more effectively. Therefore, the system faces some unique challenges in how to maintain an explainable and consistent control over the learning progress and how to adjust dynamically according to the users' response.

ChatTutor has a modularized design, encompassing three core processes-*interaction*, *reflection*, and *reaction*, each further composed of chained LLM-powered tools to perform atomic tasks. The processes are connected to each other through various memory modules, which store the essential data describing the overall progress and support update and retrieval. The design enables **structured memory control** and **adaptive reflection** on status quo. ChatTutor carries out every stage in education systematically and dynamically, including instructing, question answering, exercise offering and evaluating. Note that the system is designed for general purpose of learning instead of targeting a specific subgroup.

Evaluation of the proposed system is conducted by analyzing statistics collected from learning logs and subjective human feedback. Results show that ChatTutor can satisfactorily handle various educational activities, including adaptive course plan design, consistent instructing, impromptu question answering, etc. Meanwhile ablation study demonstrates the advantage in performance stability and consistency over long-term interaction and faced with intentional disruptions.

2 Related Work

Ever since the development of artificial intelligence techniques, methods and tools have been proposed to assist in teaching and learning process. AutoTutor [21] is the first conversation-based intelligent tutoring system, which inspires a number of works followed [12, 13, 19, 30, 39, 47]. In addition to AutoTutor's application to various fields, enhancement of specific aspects of education are also investigated, including adaptive feedback [13, 38], learning material recommendation [31, 42], and classifying learners [22, 25, 32]. Commonly adopted techniques include data mining [14], condition-action rule based [25, 42], and bayesian based methods [22], and reinforcement learning [18, 27]. NLP-specific techniques like semantic analysis [21] and textual entailment [28, 40, 48] are also adopted. In terms of application field, existing systems often rely on well-structured knowledge bases and therefore only target a single domain, most popular among which are health [15, 29], computer science [23, 31], and language learning [16, 42].

As for applications with LLMs, with proper prompting and chaining, a number of works have exploited LLMs in following diverse instructions [7, 10, 11, 35], decomposing tasks [49], refining answers [26, 43], using external tools [37], and simulating human behaviors [36]. While our work focuses on building an interactive tutoring system that works cooperatively with human users, featuring dynamic reflection.

3 Overview of ChatTutor

ChatTutor is essentially a dialogue-based tutoring system that aims to help learners acquire knowledge on one given topic systematically. As shown in Figure 1, the whole learning process is carried out in natural language conversations, with time-to-time backend reflection and reaction to update the memories. This section gives a general picture of the system's workflow. We start with explaining the design principles by introducing three underlying processes

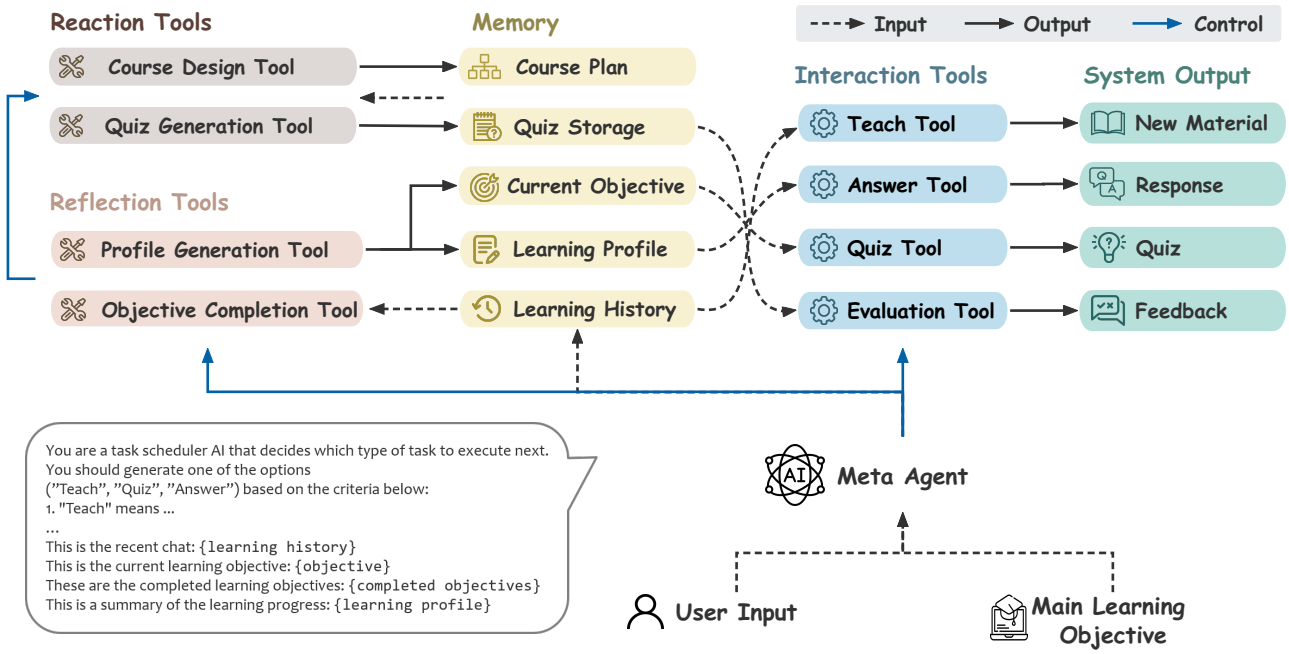

Figure 2: An overview of the system's modular implementation and execution in a single round of conversation.

within the system. Then, we briefly go over the components employed to realize each process. Finally, we provide a complete introduction to how each process and components work together. Note that the proposed system mainly explores autonomous tutoring and adaptive system reflection with chained LLMs, while we do acknowledge the potential of fallacious and biased output due to inherent flaws in LLMs.

3.1 Design Principles

The system design demonstrates the breakdown into three core system processes: *Interaction*, *Reflection*, and *Reaction*. They each has a modularized implementation and is connected to one another to form an execution loop that empowers the whole education process.

Interaction. The interactive dialogue between the system and the user is the media for tutoring and learning, and therefore is the major process of the designed system. LLMs like ChatGPT can interact with users in a responsive and robust way in daily chit-chat. However, tasking them with long-term purposeful interaction is still tricky given the restriction on context length. As for educational purposes, it is especially important to keep the interaction on track, meanwhile ensuring its accessibility and informativeness.

Reflection. To facilitate interaction, we devise a reflection process to generate high-level insights on the learning progress, which serves as global information [36] input into the system module. It is expected to help adjust system response dynamically based on user preference and behavior to achieve personalized tutoring.

Reaction. Along with reflection, reaction refers to the automatically triggered system behavior afterward, including adjustment of course plan and quiz generation. It differs from the interaction process in that *interaction* is always triggered directly by a new round of response, while *reaction* is performed at the backend from time to time, subsequent to the reflection process.

3.2 Components

The introduced three processes are further realized by separate components that support or execute a single task. There are three kinds of components: *Tools*, *Memories*, and *Meta Agent*.

Tools. Under the principled design, each process is embodied by a set of tasks performed either sequentially or in parallel. For instance, there are diverse ways of engaging with the student, such as providing instructions, addressing questions, administering quizzes, and offering feedback. This variation in approaches complicates the development of a single unified solution. We therefore devise separate modules for each specific task to ensure performance. We term those modules as "tools", and that each tool is a task-specific prompted LLM responsible for generating system output or updating memories, as shown in Figure 2. For example, *interaction* is broken down into four types of response in terms of education function, each hosted by one well-prompted tool. At each round, only one tool is used to generate the response.

Memories. Apart from tools, data storage is required to host information generated by reflection and reaction processes, while also supporting querying and updating. We propose four types of memories to record the progress and current status of learning, each stored in distinct data format and supports different ways of querying and updating. Another critical feature of the memories is that they serve as a linkage between different sets of tools to pass on information to control tool output. The detailed description of each tool and memory can be found in the next section.

Meta Agent. Above all three processes, we introduce meta agent, the single access of the control flow. It is powered by LLM and

prompted to decide what specific tasks to execute next. See Figure 2 for an example prompt for controlling the interaction process. The template contains helpful information retrieved from the memory and asks for an output deciding the type of interaction process. In our implementation, the meta agent only controls the interaction tools, while we set a fixed time interval for the execution of reflection process.

3.3 Overview of Control Flow

Above all, all designs serve for the ultimate goal of better interaction with the users. The system reflects from time to time to update cognition on the overall progress, and in turn refines the interaction production with new insights. At the frontend, the user first inputs what to learn with desired difficulty level. Then the system automatically calls the course design tool to generate the initial course plan, and starts the conversation accordingly. Upon receiving a new round of user input, the meta agent decides which interaction tool to use and the tool executes the task correspondingly to generate a new response with queried information from memories. At the backend, the reflection tools are triggered to reflect on the status quo and update the learning profile and current objective, after which the reaction tools will be triggered immediately to generate new quiz questions and update the course plan.

As shown in Figure 2, the right side represents the interaction process that is presented on the user interface, while the left side demonstrates the backend processes that are responsible for generating and updating memory modules. Practically, throughout each dialogue session, the reflection and reaction processes run alternatively at the backend, where the output result is periodically utilized by the interaction process to produce the final response to the user in each round. Table 1 presents detailed usage of each tool in the three processes, including the input and output memory content, and the condition for tool execution. The learning proceeds until all objectives in the course plan have been completed.

4 Structured Control and Adaptive Reflection

As described in the previous section, the system functions through the combination of memory modules and LLM-powered tools, where memories are extracted as part of prompt in the tools. Table 1 presents detailed usage of each tool in the three processes, including the input and output memory content, and the condition for tool execution. We further describe the key features of ChatTutor along with explaining the functionality of core components below.

4.1 Structured Memory Control

The interactive and cooperative feature of a tutoring system calls for the need to communicate with the users effectively about current and future progress. Meanwhile, it is also important to keep the system itself aware of the progress to ensure better stability. We therefore design various memory modules in different storage format and function to support the mutual communication.

Course Plan. The course plan is stored in a tree structure, with each node representing an atomic topic in the course, and its child nodes representing the sub-topics. The course is expected to be taught and learnt in depth-first traverse order. Current objective is a pointer pointing to the next uncompleted objective node in the tree to denote current progress. Such structure allows for presentation to the users, informing them of the overall status of learning, while enabling mechanistic operation by the system. Specifically, the course design tool is used at the beginning of the learning to generate the initial course plan based on user's desired topic and difficulty level. In each new round of conversation, objective completion tool is called to update status of the current objective based on the recent and relevant learning history. Then, the course design tool is asked to update the current course plan while maintaining the completed objectives.

Learning History. As for learning histories, the recent history is stored as plain text that can be directly fed into the LLM, whereas the relevant history is stored along with their embedding and queried with cosine similarity with embeddings of current objective upon usage. The detailed mechanism can be seen in Figure 3. Meanwhile, the benefit of explicitly collecting completed objectives also extends to more effective quiz offerings, which will be detailed in the next section.

Quiz Storage. A crucial function of an ITS system is to offer adaptive quiz that helps the learner review and master what has been learned. In ChatTutor, the LLM is instructed to generate quiz questions based on learning materials and formatted as a structured json string. The questions are stored corresponding to each learning objective with explicit status marking, and extracted in order whenever the quiz tool is called. The quiz questions will keep appearing in the next quiz batch until it is answered correctly by the learner.

4.2 Adaptive Reflection and Reaction

Reflection and reaction processes at the backend are closely bound to each other in ChatTutor, whereas reflection process generates high-level insight about the learning progress, reaction process updates the structured memory modules based on the insights. The sequential and dependent design could more accurately infer about the status and enhance the stability and adaptiveness of the system behavior.

Learning Progress Control. One core function of reflection process is to control the learning progress by determining when to move on to next learning objective. Objective completion tool is prompted to judge whether the current objective has been completed based on queried learning histories based on text embedding similarity. Whenever the current objective is considered completed, the status of the course plan and the pointer will automatically be updated. Meanwhile, the reaction process "quiz generation" will be triggered as well. It is prompted to generate several representative quiz questions for the completed current objective, with relevant queried learning history provided, which ensures the stability and relevance of the generated quiz question. The questions are stored in the memory until the meta agent decides it is time for a quiz, where the corresponding quiz questions are retrieved from the storage for the completed objectives and further filtered and organized by the quiz tool in the interaction process to present to the user.

Profile and Course Plan Update. Apart from reflecting on objective status, an important component is user's learning profile. Learning profile summarizes what the user has learned and gives high-level insight on the user's preference based on conversation

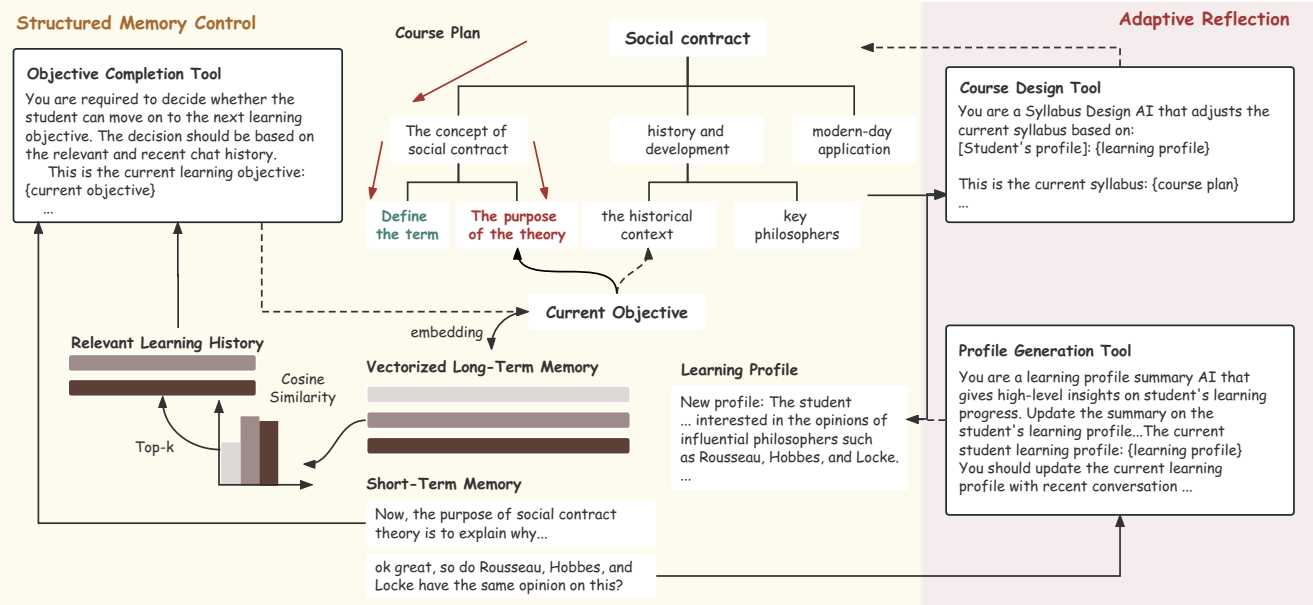

Figure 3: A detailed illustration of how course plan is stored and manipulated structurally and how reflection process helps customize the reaction followed.

Table 1: A summary of the detailed tool usage. "Input" means the memories are part of the prompt.

Process	Tool Name	Execution Condition	Input	Output/Update
Interaction	Teach Tool	Meta agent	Learning history, Current objective, Learning profile	System output
	Answer Tool	Meta agent	Learning history	System output
	Quiz Tool	Meta agent	Quiz storage, Learning profile	System output
	Evaluation Tool	Quiz	Learning history	System output
Reflection	Profile Generation	Each round	Learning history, Learning profile	Memory: Learning profile
	Objective Completion	Each round	Learning history, Current objective	Memory: Current objective
Reaction	Course Design	Profile generation	Course plan, Learning profile	Memory: Course plan
	Quiz Generation	Objective completion	Learning history, Current objective	Memory: Quiz storage

history. Though not directly presented to users, it is crucial to the stability of system's memory update and overall understanding of the learning process. It is especially useful as part of the input to course design tool to provide direction for course plan adjustment.

At each round of conversation, the system automatically reflects on the recent dialogue and updates learning profile with profile generation tool. The tool is a prompted LLM that takes recent dialogues and current profile summary as input and outputs a new version of learning profile, summarizing the learned knowledge, the user's reaction and preference mainly. Then it is fed into the course design tool for a new version of course plan generation. Figure 3 provides an example of profile generation tool generating high-level insight of *"the student seeks to expand their knowledge on the different perspectives and purposes of the social contract as discussed by the philosophers."* after the user asks a follow-up question about different philosophers' opinion. This further leads to an updated course plan that enhances deeper understanding of the theory.

5 Experiments

To demonstrate and analyze the features of our tutoring system, the experiments are conducted in two folds. We invite a number of users to learn a series of pre-defined topics using the system. During interaction, we collect critical statistics and record the conversation for future analysis. After learning completes, the users are required to answer a questionnaire to rate their experience with the system from multiple perspectives. We also develop ablation systems to better understand the effect of each process and module.

5.1 Experimental Design

System Setup. In addition to the main system, we implement two ablation systems with only partial functions. Specifically, we have one system without reflection process (w/o Reflection) and another with both reflection and reaction processes removed (w/o Reflection & Reaction). For ChatTutor w/o Reflection, the reflection process is removed so no learning profile is generated throughout the whole process, and the reaction process is triggered at a fixed time interval with limited input information. For ChatTutor w/o Reflection &

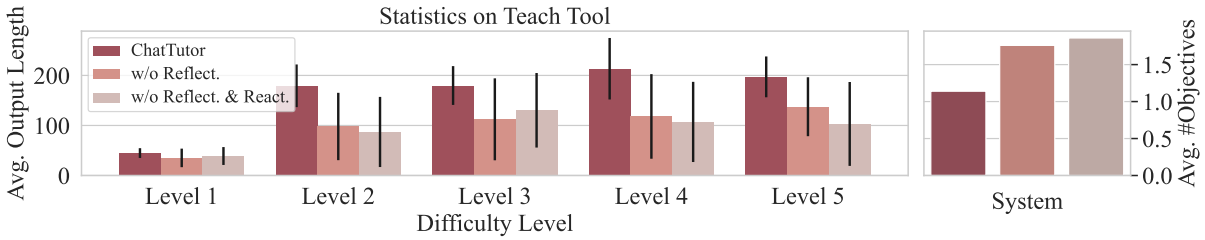

Figure 4: Average output length (calculated by the number of words) and the number of objectives covered in each output for different systems. Average number of objectives are manually annotated with 50 randomly sampled response from each system.

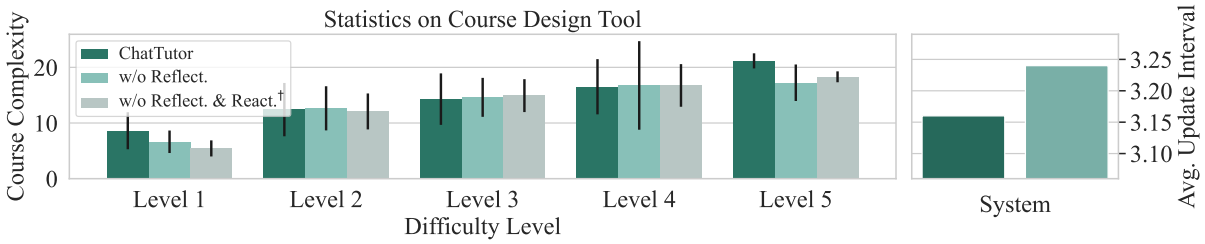

Figure 5: Average course plan complexity (calculated by the number of objectives) and update interval (calculated by the number of conversation rounds in between) by course design tool for different systems.† means this is the baseline statistics as the system without reflection or reaction processes has a fixed course plan throughout learning.

Reaction, only recent learning history and the initial course plan are available for tools.

Main Learning Objectives. For system evaluation, we collect learning objectives that cover a wide range of academic subjects and some daily life phenomena, varying in granularity and language format. We first ask GPT-4 to generate a set of general academic domain. Then we ask for generation of more fine-grained subjects under each domain and the related classic concepts. Besides, we also include some daily phenomena that may inspire people's wondering. We encourage GPT-4 to generate a typical list of them in diverse language style. To demonstrate the system's robustness in dealing with various types of learning objectives, we randomly sample 80 topics from generated fine-grained subjects, atomic concepts, and daily wondering. To make the learning process more diverse and controllable, we also design 5 difficulty levels according to Bloom's taxonomy [5] and randomly assign them to each topic. In evaluation, each topic is learned independently with three systems, making up altogether 240 courses. Table 2 shows a sample of learning objectives we adopt.

Table 2: Examples of learning objectives used for evaluation.

Category	Main Learning Objective
Subjects	Developmental psychology Impressionism Computer architecture
Atomic Concepts	Stream of consciousness Earth's mantle
Daily Wondering	How do bees communicate and find their way back to the hive? How do rainbows form and why do they have different colors?

Participants. We invite 13 average adult users who are proficient in English to participate in learning. Every single course is randomly assigned to one user, while we make sure that each participant does not get repeated course topics.

Statistical Analysis. We collect various statistics for analysis, including (1) *Complexity of course plan* reflects the ability to design adaptive course plan; (2) *Average length of system response* and *Average number of objectives per response* are indicators of instruction informativeness; (3) *Frequency of course plan update* shows the reflective feature of the system; and (4) *Frequency of in-course quiz* explores the pattern of quiz offerings.

Survey Design. After completing the course, the learner is required to answer a survey composed of 8 questions targeting different aspects of the system. Each question is a statement to be rated on a 1~5 scale, where higher scores mean better agreement with the statement. Table 5 presents the statements by category.

Stability Analysis. To further demonstrate the benefit of our system design, intentional disruption to the learning process is conducted to test the stability of the system. We take 15 most difficult topics (difficulty 4~5) in our list and for each learning process apply 3 consecutive rounds of attacks with ChatGPT generating a random question. The system is expected to answer robustly to the question and resume the original learning course after the disruption. we manually annotate the quality of resumed learning after disruption and the quality of response for the attack questions. In our experiments, each topic is learned independently with three systems. We evaluate each learning process in terms of 1) *Repeat*: the degree of repetition in course material, 2) *Omit*: whether there is omission

Table 3: Survey results for learning courses at difficulty level 1∼3. † means the score evaluates the initial course plan only, as no changes in course plan happen throughout the learning process. It could be viewed as the static quality evaluation of course plan generated from scratch. * means p-value < 0.1 using t-test.

System	Course Plan		Instruction		Question Answering		Quiz	
	Relevance	Coherence	Consistency	Accessibility	Timeliness	Consistency	Relevance	Judgment
ChatTutor	**4.72**	4.51	4.32	**4.77***	4.41	4.82	**4.88**	4.24
ChatTutor w/o Reflect.	4.71	**4.62**	**4.46**	4.66	4.64	4.85	4.75	**4.65**
ChatTutor w/o Reflect. & React.	4.97†	4.77†	4.34	**4.77**	**4.75**	**4.95**	4.86	4.36

Table 4: Survey results for learning courses at difficulty level 4∼5. † means the score evaluates the initial course plan only, as no changes in course plan happen throughout the learning process. It could be viewed as the static quality evaluation of course plan generated from scratch. * and ** means p-value < 0.1 and < 0.05 using t-test.

System	Course Plan		Instruction		Question Answering		Quiz	
	Relevance	Coherence	Consistency	Accessibility	Timeliness	Consistency	Relevance	Judgment
ChatTutor	**4.87***	**4.87****	**4.26****	4.53	3.67	**5.00**	4.87	**4.20**
ChatTutor w/o Reflect.	4.67	4.60	4.14	4.87	4.80	4.67	4.40	4.13
ChatTutor w/o Reflect. & React	4.93†	4.73†	3.73	**5.00**	**4.33**	4.93	**4.93**	4.00

Table 5: The complete survey questions. Learners are asked to rate the compatibility of each statement on a scale of 1∼5.

Course Plan
1. Relevance: The course plan is relevant to the main objective.
2. Coherence: The course plan is coherent and logical.
Instruction
3. Consistency: The instruction content strictly follows the course plan.
4. Accessibility: The language used is easy to understand.
Question Answering
5. Timeliness: The learner's questions always get immediate response.
6. Consistency: The response is consistent with learning material.
Quiz
7. Relevance: The quiz questions match what has been covered.
8. Judgment: The quiz evaluation is accurate in parsing and scoring.

of sub-topics while instructing, and 3) *Response*: whether the system responds robustly to user's random questions. Each learning is scored with 3 aspects respectively and the score can be 0, 0.5 or 1. Specifically, for response robustness, direct ignorance of the question or repeated template answer like "Let's stay focused on the course material." will be considered a sign of lack of robustness.

5.2 Results

Statistical Results. Figure 4 presents the statistical characteristics related to teach tool, including average length of output and the average number of objectives covered in each generation. Overall all systems can generate tailored output according to difficulty level. Higher difficulty comes along with longer and more informative output. It means the teach tool is successfully aware of the dynamic prompting controlled by difficulty. What is worth noting is that ChatTutor generates significantly longer output with the smallest variation. It demonstrates that ChatTutor is able to consistently generate informative content on the given topic, which is further testified by the number of objectives covered in each output. This phenomenon shows the benefits of structured memory control, where the objective completion tool reflects on and updates the current objective so that the teach tool is prompted to give new materials on one specific atomic objective.

Figure 5 showcases the behavior of course design tool. Note that all systems start with the same initial course plan. The system without reflection relies on learning history in course plan updates, while ChatTutor takes advantage of the learning profile. All systems can generate difficulty-aware course plans, seen in a positive correlation between complexity and difficulty. Note that ChatTutor shows more stability in complexity control compared to ablation systems without intermediate learning profile generation step, demonstrating the effectiveness of reflection process stabilizing the reaction process. In terms of frequency of updates, we observe that ChatTutor seems more prone to making updates, meaning that it is easier to detect helpful information in a compressed learning profile compared to verbose learning history.

As for in-course quizzes, the average interval between quizzes for ChatTutor and ablation system are 6.91 and 10.20, respectively. Without the backend processes, the meta agent has little information available to suggest effective quiz generation.

Survey Results. Table 3 and Table 4 present results on survey questions. Overall, with powerful ChatGPT, all systems demonstrate promising usability and quality. The designed reflection and reaction processes offer advantages in complex learning settings. For intricate course plan updates, the main system achieves better coherence. Through reflection and reaction, the system maintains better control with much more consistent instructions and more faithful quiz evaluation during longer conversations. While all three systems seem to generate highly relevant quiz questions, ChatTutor actually generates much more fine-grained questions querying about detail information in the learning material, while the questions from ablation systems are largely answerable based on the course plan, as demonstrated in Table 8. This advantage is attributed to online quiz generation and storage unique to reflection and reaction processes. However, it should also be noted that the

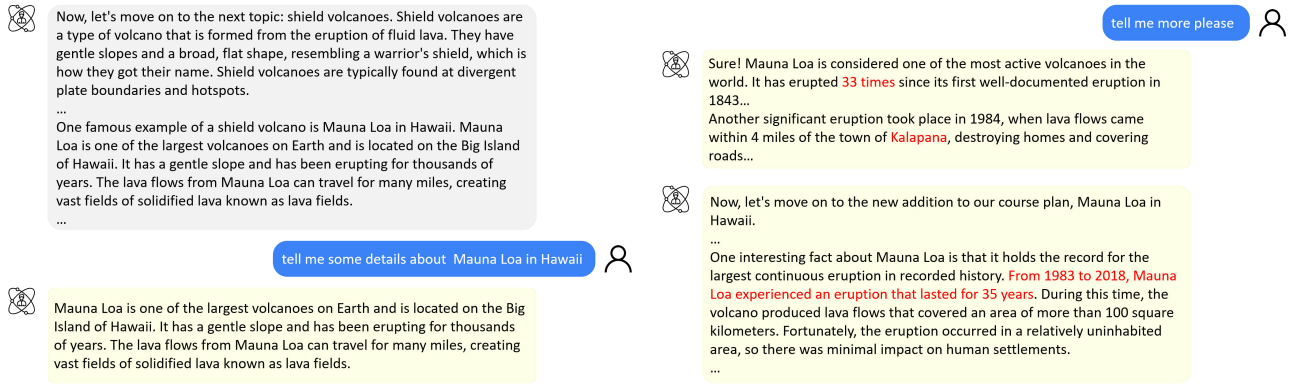

Figure 6: An example of hallucination. Hallucinated generation as been colored red.

Table 6: Results of stability evaluation. "Repeat" means the repetition of learning materials and "omit" means some topics are skipped. Higher score indicates better stability.

System	Repeat	Omit	Response	Overall
ChatTutor	0.50	0.93	0.60	**0.68**
w/o Reflect.	0.60	0.47	0.40	0.49
w/o Reflect. & React.	0.33	0.67	0.67	0.56

timeliness in response of ChatTutor is compromised by backend processes. It indicates that prompts with global information may interfere with the LLM's ability to focus locally and generate timely and coherent response.

Stability Analysis. As shown in Table 6, thanks to the reflection and reaction processes, ChatTutor has the overall best performance in terms of stability and robustness, with up to 20% increase compared to ablation systems. Whereas there is a clear trade-off between the repetition and omission of course material in the two ablation systems, while both signify instability. The fact that ChatTutor tends to ignore user's irrelevant questions more often also echoes finding in user study and highlights the reconciling effect between robustness and stability and controllability.

5.3 Case Study

In this section, we further demonstrate how ChatTutor behaves with detailed case studies. To compare with real-world education scenario, we adopt the machine learning course on Coursera platform. As show in Table 8, it can be seen that ChatTutor can satisfactorily cover the major topics of the course, meanwhile also maintaining a logical dependence between crucial concepts. On the other hand, it should be noted that while ChatTutor tends to propose a wide range of concepts, real world teaching pays more attention to technical problems, including how to solve a specific machine learning problem and what practical tricks are commonly used. This down-side could be compensated by the adaptiveness and timely responsiveness of ChatTutor, where users can motivate more in-depth discussion with impromptu questions.

While ChatTutor largely provides accurate information on "machine learning", in another case featuring "volcanoes", we find that the system stumbles when users ask for more details. For example, as show in Figure 6, the system makes typical hallucination due to the knowledge cutoff of training data, and also confuses two towns in Hawaii in a specific eruption. When being pushed to provide more information, it hallucinates the eruption duration as well. Problems like this could be mitigated with retrieval-augmented generation technique given a relevant knowledge base.

6 Conclusion and Future Work

This work is a pioneering exploration of an LLM-powered intelligent tutoring system, with an emphasis on the possibility of employing LLMs to complete complex and dynamic long-term interactions. The proposed system, ChatTutor, can satisfactorily complete the core functions of an intelligent tutoring system. As ablation study shows, the three-process system design provides unique benefit in ensuring the stability and consistency of the system behavior, meanwhile maintaining flexibility and adaptiveness with the designed mechanism. Although our evaluation reveals the advantage of memory mechanism and process design in long-term interaction, we acknowledge that comprehensively evaluating an intelligent tutoring system is far more tricky [8, 20]. It is also important to design more standard metrics for interactive systems in the era of LLMs. The system also faces concerns unique to LLMS, such as the validity of generated education content and the potential bias from training data [17, 33], which might be mitigated by domain-specific fine-tuning and retrieval-augmented fact-checking. Despite that, this work proposes a meaningful application of chaining LLMs in the educational process, which might inspire future efforts in employing LLMs to build more interactive and reflective systems.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 62276154 & No. 62236004), Research Center for Computer Network (Shenzhen) Ministry of Education, the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and GJHZ202402183000101), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), and the National Key R&D Program of China (No.2022ZD0116312).

Table 7: Example quiz questions generated by different systems on the topic of "Gravity Waves". "Fine-grained" means the questions are more detailed so that the answer is not obvious from the course plan.

Course Plan	Quiz Questions	Fine-grained?
1. Introduction to Gravity Waves a. Definition and Key Concepts i. Differentiation between Gravity Waves and Gravitational Waves ii. Causes and Influences of Gravity Waves b. Factors Affecting Gravity Wave Formation i. Atmospheric Stability ii. Wind Shear iii. Topography and Surface Obstacles 2. Characteristics and Properties of Gravity Waves a. Wave Amplitude and Wavelength i. criteria for wave breaking ii. energy transfer and propagation b. Wave Speed and Frequency i. dispersion relation ii. role of buoyancy and ambient fluid properties 3. Observation and Detection of Gravity Waves a. Remote Sensing Techniques i. satellite imagery ii. lidar and radar ...	**ChatTutor** 1. What is amplitude in the context of gravity waves? a) The distance between successive crests or troughs of a wave b) The maximum displacement of particles within a wave c) The rate at which wave energy is transferred vertically d) The disrupted, turbulent state of a wave 2. How does wavelength influence the behavior of gravity waves? a) It determines the spatial scale of the wave b) It represents the distance over which the wave repeats itself c) It influences the rate at which wave energy is transferred vertically d) It determines the amplitude of the wave **ChatTutor w/o Reflect. & React.** 1. What is the main focus of the course "Introduction to Gravity Waves"? a) Gravitational Waves b) Atmospheric Stability c) Causes and Influences of Gravity Waves d) Topography and Surface Obstacles 2. What are the factors affecting gravity wave formation? a) Wave Speed and Frequency b) Wind Shear c) Wave Amplitude and Wavelength d) Dispersion Relation	✓ ✓ ✗ ✗

Table 8: Course plan comparison for "Machine Learning".

Coursera	ChatTutor
1. introduction a. welcome to machine learning b. supervised learning c. unsupervised learning 2. linear regression with one variable a. model representation b. cost function c. gradient descent d. gradient descent for linear regression 3. linear algebra review a. matrices and vectors b. addition and scalar multiplication c. matrix vector multiplication d. matrix matrix multiplication e. matrix multiplication properties f. inverse and transpose 4. linear regression with multiple variables a. multiple features b. gradient descent for multiple variables c. gradient descent in practice i feature scaling d. gradient descent in practice ii learning rate e. features and polynomial regression f. normal equation g. normal equation noninvertibility h. working on and submitting programming assignments 5. octave matlab tutorial a. basic operations b. moving data around c. computing on data d. plotting data e. control statements for while if statement ...	1. Introduction to Machine Learning a. Definition of Machine Learning b. Importance and Applications of Machine Learning c. Types of Machine Learning 2. Supervised Learning a. Definition and Explanation b. Classification i. Binary Classification ii. Multiclass Classification c. Regression i. Linear Regression ii. Polynomial Regression 3. Unsupervised Learning a. Definition and Explanation b. Clustering i. K-Means Clustering ii. Hierarchical Clustering c. Dimensionality Reduction i. Principal Component Analysis (PCA) ii. t-Distributed Stochastic Neighbor Embedding (t-SNE) 4. Evaluation and Validation a. Training, Testing, and Validation Data b. Accuracy, Precision, Recall, and F1-Score c. Cross-Validation 5. Model Selection and Regularization a. Bias-Variance Tradeoff b. Overfitting and Underfitting c. Regularization Techniques 6. Introduction to Neural Networks ...

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. arXiv:2302.04023 [cs.CL]

[3] Aditi Bhutoria. 2022. Personalized education and Artificial Intelligence in the United States, China, and India: A systematic review using a Human-In-The-Loop model. *Computers and Education: Artificial Intelligence* 3 (2022), 100068. https://doi.org/10.1016/j.caeai.2022.100068

[4] Benjamin S Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher* 13, 6 (1984), 4–16.

[5] Benjamin S Bloom and David R Krathwohl. 2020. *Taxonomy of educational objectives: The classification of educational goals. Book 1, Cognitive domain.* longman.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* (2023).

[8] Peter A Cohen, James A. Kulik, and Chen-Lin C. Kulik. 1982. Educational Outcomes of Tutoring: A Meta-analysis of Findings. *American Educational Research Journal* 19, 2 (1982), 237–248. https://doi.org/10.3102/00028312019002237 arXiv:https://doi.org/10.3102/00028312019002237

[9] R. Costello and D.P. Mundy. 2009. The Adaptive Intelligent Personalised Learning Environment. In *2009 Ninth IEEE International Conference on Advanced Learning Technologies*. 606–610. https://doi.org/10.1109/ICALT.2009.38

[10] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377* (2023).

[11] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. *arXiv preprint arXiv:2305.14233* (2023).

[12] Sidney D'mello and Art Graesser. 2012. AutoTutor and Affective Autotutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers That Talk Back. 2, 4 (2012), 1–39. https://doi.org/10.1145/2395123.2395128

[13] Myroslava O. Dzikovska, Natalie B. Steinhauser, Elaine Farrow, Johanna D. Moore, and Gwendolyn E. Campbell. 2014. BEETLE II: Deep Natural Language Understanding and Automatic Feedback Generation for Intelligent Tutoring in Basic Electricity and Electronics. *International Journal of Artificial Intelligence in Education* 24 (2014), 284–332. https://api.semanticscholar.org/CorpusID:15442631

[14] Vanessa Echeverria, Bruno Guamán, and Katherine Chiluiza. 2015. Mirroring Teachers' Assessment of Novice Students' Presentations through an Intelligent Tutor System. In *2015 Asia-Pacific Conference on Computer Aided System Engineering*. IEEE, 264–269.

[15] Gilan M El Saadawi, Eugene Tseytlin, Elizabeth Legowski, Drazen Jukic, Melissa Castine, Jeffrey Fine, Robert Gormley, and Rebecca S Crowley. 2008. A natural language intelligent tutoring system for training pathologists: Implementation and evaluation. *Advances in health sciences education* 13 (2008), 709–722.

[16] Mustafa Al Emran and Khaled Shaalan. 2014. A Survey of Intelligent Language Tutoring Systems. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2014-09). 393–399. https://doi.org/10.1109/ICACCI.2014.6968503

[17] Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From Pretraining Data to Language Models to Downstream Tasks: Tracking the Trails of Political Biases Leading to Unfair NLP Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 11737–11762. https://doi.org/10.18653/v1/2023.acl-long.656

[18] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. 2016. Affective Personalization of a Social Robot Tutor for Children's Second Language Skills. 30, 1 (2016). https://doi.org/10.1609/aaai.v30i1.9914

[19] Arthur C. Graesser, G. Tanner Jackson, Eric Mathews, Heather H. Mitchell, Andrew M. Olney, Matthew Ventura, Patrick Chipman, Donald R. Franceschetti, Xiangen Hu, Max M. Louwerse, and Natalie K. Person. 2003. Why/AutoTutor: A Test of Learning Gains from a Physics Tutor with Natural Language Dialog. https://api.semanticscholar.org/CorpusID:5539246

[20] Arthur C. Graesser, Natalie K. Person, and Joseph P. Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology* 9 (1995), 495–522. https://api.semanticscholar.org/CorpusID:143574878

[21] Arthur C. Graesser, Kurt VanLehn, Carolyn P. Rose, Pamela W. Jordan, and Derek Harter. 2001. Intelligent Tutoring Systems with Conversational Dialogue. 22, 4 (2001), 39. https://doi.org/10.1609/aimag.v22i4.1591

[22] Beate Grawemeyer, Manolis Mavrikis, Wayne Holmes, Sergio Gutierrez-Santos, Michael Wiedmann, and Nikol Rummel. 2016. Affecting Off-Task Behaviour: How Affect-Aware Feedback Can Improve Student Learning. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (Edinburgh, United Kingdom) *(LAK '16)*. Association for Computing Machinery, New York, NY, USA, 104–113. https://doi.org/10.1145/2883851.2883936

[23] Foteini Grivokostopoulou, Isidoros Perikos, and Ioannis Hatzilygeroudis. 2013. An intelligent tutoring system for teaching FOL equivalence. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, Vol. 20.

[24] Foteini Grivokostopoulou, Isidoros Perikos, and Ioannis Hatzilygeroudis. 2017. An Educational System for Learning Search Algorithms and Automatically Assessing Student Performance. *International Journal of Artificial Intelligence in Education* 27 (2017), 207–240. https://api.semanticscholar.org/CorpusID:2871394

[25] Hsieh S. J. and Cheng Y. T. 2014. Algorithm and intelligent tutoring system design for programmable controller programming. *International Journal of Advanced Manufacturing Technology* 71 (2014), 1099.

[26] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651 [cs.CL]

[27] Ankit Malpani, Balaraman Ravindran, and Hema A Murthy. 2011. Personalized Intelligent Tutoring System Using Reinforcement Learning.. In *FLAIRS Conference*. 561–562.

[28] Philip M McCarthy, Vasile Rus, Scott A Crossley, Arthur C Graesser, and Danielle S McNamara. 2008. Assessing Forward-, Reverse-, and Average-Entailer Indices on Natural Language Input from the Intelligent Tutoring System, iSTART.. In *FLAIRS Conference*. Citeseer, 165–170.

[29] Jenny McDonald, Alistair Knott, Sarah J. Stein, and Richard Zeng. 2013. An empirically-based, tutorial dialogue system: design, implementation and evaluation in a first year health sciences course. https://api.semanticscholar.org/CorpusID:207826233

[30] Danielle S McNamara, Tenaha P O'Reilly, Rachel M Best, and Yasuhiro Ozuru. 2006. Improving adolescent students' reading comprehension with iSTART. *Journal of Educational Computing Research* 34, 2 (2006), 147–171.

[31] Phaedra Mohammed and Permanand Mohan. 2015. Dynamic cultural contextualisation of educational content in intelligent learning environments using ICON. *International Journal of Artificial Intelligence in Education* 25 (2015), 249–270.

[32] Elghouch Nihad, Yassine Zaoui Seghroucheni, et al. 2017. Analysing the outcome of a learning process conducted within the system ALS_CORR [LP]. *International Journal of Emerging Technologies in Learning (Online)* 12, 3 (2017), 43.

[33] Debora Nozza, Federico Bianchi, and Dirk Hovy. 2022. Pipelines for Social Bias Testing of Large Language Models. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*. Association for Computational Linguistics, virtual+Dublin, 68–74. https://doi.org/10.18653/v1/2022.bigscience-1.6

[34] TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI* (2022).

[35] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[36] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442* (2023).

[37] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shi Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bo Li, Ziwei Tang, Jing Yi, Yu Zhu, Zhenning Dai, Lan Yan, Xin Cong, Ya-Ting Lu, Weilin Zhao, Yuxiang Huang, Jun-Han Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Tool Learning with Foundation Models. *ArXiv* abs/2304.08354 (2023). https://api.semanticscholar.org/CorpusID:258179336

[38] Rod D Roscoe and Danielle S McNamara. 2013. Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology* 105, 4 (2013), 1010.

[39] Vasile Rus, Sidney D'Mello, Xiangen Hu, and Arthur Graesser. 2013. Recent Advances in Conversational Intelligent Tutoring Systems. 34, 3 (2013), 42–54. https://doi.org/10.1609/aimag.v34i3.2485

[40] Vasile Rus and Arthur C. Graesser. 2006. Deeper Natural Language Processing for Evaluating Student Answers in Intelligent Tutoring Systems. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2* (Boston,

Massachusetts) *(AAAI'06)*. AAAI Press, 1495–1500.

[41] Vasile Rus, Dan Stefanescu, Nobal Niraula, and Arthur C. Graesser. 2014. DeepTutor: Towards Macro- and Micro-Adaptive Conversational Intelligent Tutoring at Scale. In *Proceedings of the First ACM Conference on Learning @ Scale Conference* (Atlanta Georgia USA, 2014-03-04). ACM, 209–210. https://doi.org/10.1145/2556325.2567885

[42] Gómez S., Zervas P., Sampson D. G., and Fabregat R. 2014. Context-aware adaptive and personalized mobile learning delivery supported by UoLmP. *Journal of King Saud University – Computer and Information Sciences* 26 (2014), 47.

[43] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366 [cs.AI]

[44] Ying Tang, Joleen Liang, Ryan Hare, and Fei-Yue Wang. 2020. A Personalized Learning System for Parallel Intelligent Education. *IEEE Transactions on Computational Social Systems* 7 (Apr 2020), 352–361. https://doi.org/10.1109/TCSS.2020.2965198

[45] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[47] Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist* 46, 4 (2011), 197–221.

[48] Kurt VanLehn, Pamela Jordan, and Diane Litman. 2007. Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In *Workshop on Speech and Language Technology in Education*. Citeseer.

[49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL]