BMT23
5th International Conference
Business Meets Technology.
Valencia, 13th-15th July 2023

GENERATIVE AGENTS TO SUPPORT STUDENTS LEARNING PROGRESS

Schacht, Sigurd [a1]; Kamath Barkur, Sudarshan [a2]; and Lanquillon, Carsten [b]

[a] *University of Applied Science Ansbach, Germany*
([a1] sigurd.schacht@hs-ansbach.de, [a2] s.kamath-barkur@hs-ansbach.de)
[b] *University of Applied Science Heilbronn, Germany (carsten.lanquillon@hs-heilbronn.de)*

ABSTRACT: Ongoing assessments in a course are crucial for tracking student performance and progress. However, generating and evaluating tests for each lesson and student can be time-consuming. Existing models for generating and evaluating question-answer pairs have had limited success. In recent years, large language models (LLMs) have become available as a service, offering more intelligent answering and evaluation capabilities. This research aims to leverage LLMs for generating questions, model answers, and evaluations while providing valuable feedback to students and decentralizing the dependency on faculty. Our approach is based on the development and interaction of advanced AI-powered generative agents, built on large language models like ChatGPT, GPT-4, and Vicuna, and designed to emulate human activities such as information abstraction, context refinement, and query rating. These agents interact autonomously in a network, employing techniques like Zero-Shot and Few-Shot Prompting to generate responses and adapt to various roles and contexts. The setup includes three key agents for question generation, refinement, and quality assurance, which leverage text vectorization, document selection and filtering, and cutting-edge language models to generate, refine, and evaluate questions and answers based on specific learning objectives. In conclusion, this paper demonstrates the versatility of LLMs for various learning tasks, including question generation, model answer generation, and evaluation, all while providing personalized feedback to students. By identifying and addressing knowledge gaps, LLMs can support continuous assessment and help students improve their understanding before semester exams.

KEY WORDS: *LLM; Question Generation; Generative Agents; Personalized Assessments.*

1. INTRODUCTION

In the quest for educational excellence, a crucial aspect is ensuring students consistently meet the learning objectives (Bulut & Wongvorachan, 2022). To achieve this, it is vital to measure and track the students' state of knowledge throughout the course continuously.

How to cite: Schacht, S., Kamath Barkur, S., and Lanquillon, C. 2023. Generative Agents to Support Students Learning Progress. In Proc.: *5th International Conference Business Meets Technology*. Valencia, 13th-15th July 2023. 179-197. https://doi.org/10.4995/BMT2023.2023.16750

However, this proves to be a significant challenge for educators, especially when dealing with hundreds of students and different courses. The generation and correction of questions, though necessary for personalized feedback and assured learning progress, is a time-consuming process. To overcome these problems, we propose an automatic assessment generation and feedback system based on generative ai agents, which generates learning goal-based assessments and performs correction after each lesson, with a primary focus on enhancing learning progress.

One innovative concept this paper will explore is automatic question generation from documents, slides, transcriptions, or recommended reading materials. This raises the critical question of how to select essential paragraphs for question generation. Further, to ensure the quality of learning, we need to generate thought-provoking questions that go beyond simple multiple-choice queries and stimulate deeper thinking in students. This, however, poses the problem of assessing the quality of such questions. In addition, the question which is generated should not only focus on one paragraph or section rather should also be more focused on questions that force the students to use their knowledge to answer it.

In addition to question generation, automatic correction and guidance form a significant part of our proposed solution. While multiple-choice questions are relatively easy to grade automatically, open-ended answers present a more complex problem for automated correction. Beyond just identifying the right or wrong responses, the system should also be capable of providing constructive feedback and guiding the students towards improved responses, and showing ways how questions could be answered in a more familiar way (Kulshreshtha et al., 2022).

By addressing these issues, this paper aims to lay the foundation for a dynamic and effective automated learning tool, capable of handling large-scale education with personalization and efficiency.

The contribution of this paper is to show how generative agents which autonomously identify the important section of the given supporting material, extract questions, refine and evaluate the question toward the learning goal, and support the correction process could be set up by using prompt engineering and different large language models.

Delving further into the concept of automatic question generation from documents and recommended books, the first issue that arises is the selection of important paragraphs. Generating high-quality questions from these important paragraphs is another challenge. The idea is not to simply produce factual recall questions but to design queries that foster critical thinking and deep comprehension. However, the notion of quality in question generation raises another concern. Quality can be measured based on relevance, difficulty level, clarity, and the type of cognitive skills they target, such as recall, analysis, or evaluation (Gnanasekaran et al., 2021; Sen Gupta et al., 2019; Xie et al., 2020). Developing an assessment model incorporating these factors would be necessary for a successful automatic question-generation system.

Automatic correction and feedback for non-multiple-choice questions is another crucial component of our proposed system. As open-ended responses cannot be evaluated against a set of predefined answers, novel solutions are needed.

Providing automated recommendations for better answers is equally important. Here, the system would not only indicate the errors but also guide the student to the correct understanding. This could involve the generation of partial answers or hints, suggesting relevant resources, or pinpointing the parts of the original text that the student should revisit, like it is proposed also in Kulshreshtha et al. (Kulshreshtha et al., 2022).

All these aspects aim to foster a more interactive, engaging, and effective learning experience, ensuring that the students aren't simply memorizing information, but truly understanding and assimilating it.

2. RELATED WORK

The role of assessments in education has been significantly automated using technology, with numerous studies employing natural language processing (NLP) and machine learning (ML) for test creation. Malinova and Rahneva (Malinova & Rahneva, 2017) and Botelho et al. (Botelho et al., 2023) proposed methods for automated question generation in English and mathematics respectively. Emerson et al. (Emerson et al., 2022) introduced the use of fine-grained content for student performance prediction during tests. Transformer-based models for reading comprehension test generation were studied by Bulut and Yildirim-Erbasli (Bulut & Yildirim-Erbasli, 2022) and Runge et al. (Runge et al., 2022). Zou et. al. (Zou et al., 2022) proposed a method to generate true/false questions for reading comprehension tests based on unsupervised learning methods by leveraging different NLP techniques and a generative framework by a novel masking-and-infilling strategy. Yet, most automatically generated questions, especially those based on manually generated rules, lack linguistic diversity or unusable content. Therefore a shift towards generated questions based on neural networks occurred (Zou et al., 2022).

The importance of inclusive assessments was highlighted by Johnson and Monroe (Johnson & Monroe, 2004), while Mislevy et al. (Mislevy et al., 2002) discussed task-based language assessment models. Freeman et al. (Freeman et al., 2015) emphasized the importance of providing feedback to students by listening to what students produce and then asking questions or modeling a desired response. Improvement in automatic scoring of student responses through pre-trained language models was investigated by Liu et al. (Z. Liu et al., 2023).

Kulshreshtha et al. (Kulshreshtha et al., 2022) utilized the transformer architecture for the process of generating individualized feedback during tutoring. They take the answer from the students and a manual gold standard drafted answer to generate individual feedback for the student.

Finally, Sarsa (Sarsa, 2022) delved into the capabilities of large language models in producing programming course resources. Niraula (Niraula & Rus, 2015) suggests that an automated method for judging question quality would make the question generation process much more efficient. Therefore, developing an assessment model that incorporates these factors would be necessary for a successful automatic question-generation system.

Despite these advancements, none of these studies explored the use of generative agents for automatic assessment generation and quality assurance. Generative agents to simulate human behavior are introduced by Park (Park et al., 2023). The combination of agents and prompt engineering is presenting a unique direction for future research in automatic assessment creation and is the subject of the investigation of this paper.

3. GENERATIVE AGENTS FOR ASSESSMENT GENERATION AND EVALUATION

Introduction to Generative Agents

Generative agents are advanced computational software agents constructed on large language models. These AI-powered constructs are capable of mimicking complex human activities, such as abstracting key information from voluminous texts, refining context-specific queries, or rating queries based on their relevance (Park et al., 2023).

These software agents are designed using a state-of-the-art architecture that incorporates a large language model. A notable feature of this architecture is its ability to record a comprehensive history of interactions, all encoded in natural language. Over time, these agents can convert these accumulated experiences into higher-order reflections, effectively assimilating them into their 'memory'. This repository of information can be dynamically accessed to guide future responses, akin to the decision-making process in humans (Park et al., 2023).

The underpinning of these generative agents are large language models capable of producing high-quality text in an autoregressive manner. Prime examples include the generative pretrained transformers, such as ChatGPT and GPT-4, and models like Llama and Vicuna (Chiang et al., 2023; Y. Liu, Han, et al., 2023; OpenAI, 2023).

These agents are defined and interacted with through a technique known as 'prompting'. Prompting involves presenting the agent with a specific task or query, which it then processes to generate a suitable output. This process can be further divided into zero-shot and few-shot prompting (P. Liu et al., 2021).

Zero-shot prompting is where the agent generates a response to a query without having any prior exposure or examples of the task. On the other hand, few-shot prompting provides the model with a limited number of examples, which it uses to understand and process the task (Brown et al., 2020; Wei et al., 2022).

Interestingly, the few-shot prompting technique plays a key role in defining the agent. The agent is prompted to play various roles and situations, with the context of the task feeding into the description of the agent itself. This design simulates the way humans adapt to different roles and situations based on prior experiences.

Intriguingly, each generative agent can interact autonomously with others. This interaction manifests as one agent accepting the output of another as its input, thereby fostering a collaborative intelligence network (Park et al., 2023). The development and use of these generative agents herald a significant stride forward in the field of artificial intelligence, pushing the boundaries of machine learning and natural language understanding.

Technical Parts of the Proposed System

The foundation of the proposed system rests on three large language models (LLMs): ChatGPT (specifically, gpt-3.5-turbo), GPT-4, and Vicuna 13B.

ChatGPT gpt-3.5-turbo, hosted by OpenAI and accessed through its API, is central to the refine and answer generation agent. This model is a variant of the GPT-3 model, with an architecture that includes a transformer network, featuring an attention mechanism for comprehending input context and generating sophisticated responses. The structure leverages layers of self-attention and feed-forward neural networks, allowing it to understand and generate human-like text by predicting the probability of a word given its previous words (Brown et al., 2020; Ouyang et al., 2022).

GPT-4, also hosted by OpenAI and accessible via its API, is the backbone of the rating and control agent. GPT-4 extends its predecessor by increasing its parameters, enhancing its understanding and response generation abilities. This improvement enhances its ability to manipulate and understand complex language constructs, making it more adept at providing accurate ratings and control mechanisms (OpenAI, 2023).

The Vicuna LLM 13B model, a locally hosted model based on Facebook's Llama and fine-tuned using shared GPT instruction datasets, is instrumental for question generation. As a 13 billion parameter model, it uses its expansive architecture to generate creative and diverse questions from given inputs (Chiang et al., 2023; Touvron et al., 2023).

Embeddings are another essential aspect of the proposed system. In essence, embeddings transform words, sentences, or documents into numerical vectors that capture their semantic meaning. The OpenAI embeddings are used as a foundation for this transformation (Neelakantan et al., 2022). All documents are split into chunks of approximately 1000 characters, with an overlap of 50 characters. Each chunk generates a corresponding embedding vector. These embeddings play a critical role in the similarity search, identifying the necessary text snippets in alignment with the learning goals.

The final significant component is the vector store, specifically ChromaDB.[1] A vector store is a system for storing and retrieving vectors, used in this case to store the embeddings for each document chunk. The vector store serves as a knowledge base, allowing for fast, efficient retrieval of information when the system requires access to the support material. This feature allows the proposed system to extract, analyze, and respond to inputs quickly, making it both effective and efficient in achieving its objectives.

Proposed Agent Setup, System Design, and Proposed Workflow for Generation Question and Answers

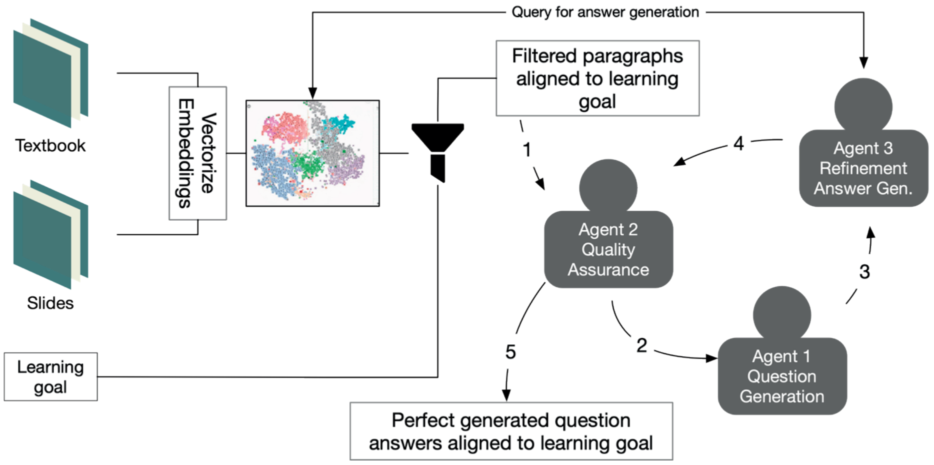
Query for answer generation
Textbook
Slides
Vectorize Embeddings
Filtered paragraphs aligned to learning goal
1
4
Agent 3 Refinement Answer Gen.
Agent 2 Quality Assurance
3
2
5
Agent 1 Question Generation
Learning goal
Perfect generated question answers aligned to learning goal

Figure 1. Illustrated representation of the interaction between agents.

Our prototype focuses on a tool engineered to function as a command line utility, programmed to accept an array of parameters. These parameters encompass documents that serve as the foundational data, the established learning objective, and a number of document sections that can be located in accordance with the learning objective.

The internal **preprocessing** of the provided documents is the base for the vector similarity search to facilitate a semantic search that selects and provides sections of the given documents from the vantage point of the learning objective (Reimers et al., 2019).

The mechanism of text vectorization is instrumental in transforming the input data. For this task, the Open AI Embeddings model is utilized (Neelakantan et al., 2022). Initially, the documents are subdivided into manageable fragments, subsequently feeding these portions into the vectorization model. The segmentation process does not rely on paragraph divisions but instead determines the chunks based on a character length of 1000 and overlapping of 50 characters with adjacent chunks.

1 https://www.trychroma.com

The phase of document selection is employed to identify the most pertinent sections in relation to the stipulated learning objective. This is accomplished through a similarity search using cosine similarity, yielding the top n-documents, with n representing a parameter defined by the user (Mishra et al., 2020). This process employs the learning goal sentence, which is vectorized using embeddings. This resultant embedding vector is used in a similarity search against the embedding vectors of each chunk in the vector store.

Next, the document filtering phase further refines the selection by only opting for those documents with a similarity score greater than 0.8. This score is derived from the comparison between the learning goal vector and the chunk vector. This ensures that the paragraphs, which will be used as a basis for question generation, are highly focused on the learning goal. With this methodology, the input documents' size becomes irrelevant, making it capable of processing millions of paragraphs.

Quality Assurance Agent, labeled as **Agent 2**, is tasked with the evaluation of each paragraph of the selected documents based on their relevance to the learning objective. This is carried out using one of the LLM models along with an evaluation prompt. The GPT-4 model is employed for rating each paragraph on a scale from 1 to 10 (Y. Liu, Iter, et al., 2023). After this, Agent 2 delivers its output to Agent 1. (See Figure 1: Step 1)

Agent 1, the question generation agent is responsible for generating at least two questions for each highly-rated paragraph (those with a rating of 9 or above). This is accomplished with the intention of verifying the learning objective, employing a self-hosted Vicuna model for this generation task. The task is formulated by using a few-shot-prompt, enriched with the learning goal and one paragraph, repeated to as often as paragraphs rated by Agent 2 exists (see Appendix for the prompt and Figure 1: Step 2).

Subsequently, **Agent 3, the refinement and answer generation agent** based on ChatGPT, refines the generated questions to enhance their precision and alignment with the learning objective. This agent discards questions referring to figures, other chapters, code, or irrelevant aspects, replacing them with 'NaN'. Moreover, this agent generates short, precise answers for each refined question using a combination of the vector store search and few shot prompting (see Appendix for the prompt and Figure 1 Step 3).

Finally, **Agent 2 (quality assurance agent)** based on GPT-4, rates the overall quality of the generated questions and answers to ascertain if they can sufficiently achieve the learning objective again by using few-shot-prompting (see Appendix for the prompt). All generated answers along with the questions are used in a single prompt, which in turn stimulates the model in two ways. First, it rates the fulfillment of the learning goals on a scale from 1 to 10, and second, it describes the reasons for the rating (Y. Liu, Iter, et al., 2023). If the learning goal isn't fully met, the model is prompted to add more questions. (See Figure 1: Step 4)

The concluding output consists of a list of refined questions along with their answers. Lastly, during the assessment evaluation, **Agent 2 (quality assurance)** is employed to evaluate the student's answers by comparing them to the previously generated answers from **Agent 3**. Also, here a specialized prompt to force this behavior is used (see Appendix for the prompt). The output is a rating if the question was answered correctly by the student, not correct, or if a rating is not possible. In addition, Agent 3 provides recommendations in natural language, on how the answer could be improved to guide the students in their learning progress. (See Figure 1: Step 5)

The use of GPT-4 in the quality assurance agent and the Vicuna model in the question generation agent, both cutting-edge technologies, ensures that the generated questions and evaluations align precisely with the defined learning goals. This approach highlights the system's potential for tailoring the learning process according to specific learning objectives.

Furthermore, the inclusion of the refinement and answer generation agent adds depth to the system's capabilities, enabling it to produce concise answers and refine questions for increased precision.

The use of similarity search and document filtering ensures that the focus remains on the learning objectives. This approach ensures that the content, both in the generation of questions and the evaluation of responses, remains consistently relevant and beneficial to the learning process.

4. EVALUATION

Findings

Our research findings suggest that the quality of the questions generated automatically for learning assessments is strongly dependent on the quality of the input text. We observed that the generation process consistently produces high-quality question candidates when the input text provides sufficient contextual information. Conversely, if the input document lacks the necessary information, the system refrains from generating questions.

In terms of language model performance, our observations indicate that large language models can occasionally hallucinate or generate new, irrelevant information. In our research, however, we found that the model performed exceptionally well without manifesting this behavior. We attribute this result not only to the utilization of a temperature setting of zero but also to the use of a vector store and few-shot prompting, which helped control the system's generation process.

We noted that the back-and-forth interaction between the agent was time-consuming. For a 30-page document, with a goal to generate 10 questions, the process took approximately 3 minutes for both question generation and quality assurance.

Our study consisted of 10 runs using different documents and learning objectives. All runs were evaluated by a Quality Agent, and each received a score greater than 8. This score was further corroborated by manual evaluation, where the perceived quality was in agreement with the automated rating. Despite the positive outcomes, we recommend further tests to confirm the system's ongoing quality and effectiveness for different contexts and types of documents.

Research limitations/implications

While the research demonstrates the potential of generative agents based on language models and prompting in creating assessment questions and evaluating test results, there are significant limitations and implications that need to be acknowledged.

One of the significant hurdles is the vast size of the large language models, specifically those with 13 billion parameters, equivalent to around 38 GB of data. This size exceeds the capacity of most consumer-grade GPUs, making the application of these models expensive due to the necessity for advanced hardware or cloud resources. Another obstacle is the processing speed, with models like GPT-4 being highly competent but also quite slow, which increases both computational time and costs. This factor may limit the broad usage of these models in real-time educational settings, where rapid generation and evaluation of questions and answers are crucial.

Furthermore, the present state of these models struggles to interpret and generate questions from information like formulas and figures. Their current architecture only allows to process ongoing text, where context must be included. At the moment, the content of the figures is derived by using the additional text in textbooks, which explains the figure. If this is missing the figure would not be used at all. Formular could be processed if they are provided in LaTeX code. Otherwise, they will be ignored. This makes it difficult to use these approaches in mathematics, physics, or other domains where formulas are utilized heavily.

Moreover, the effectiveness of these models depends heavily on the quality of the input data. If the source material, such as lecture slides, contains only bullet points with minimal context, the models may struggle to match them against learning goals, resulting in a failure to generate question-answer pairs.

Finally, the models don't always react as intended. The prompts need then to be refined to ensure that the models produce the expected output, adding an extra layer of difficulty and requiring a deep understanding of how to guide these models effectively. This limitation underlines the importance of continued research in this field, which affects prompt engineering and model alignment.

The current design of our generative agents primarily focuses on individual sessions rather than the entire course. This restrictive focus on session-specific learning goals, as opposed to the broader objectives of the course, might limit a comprehensive understanding of the student's learning journey. To address this, we could extend the agent's functionality by incorporating memory. This feature would allow the agent to track both the course-wide and session-specific learning objectives and retain information about students' performance. Following a model similar to Park et. al (Park et al., 2023), this memory would help the agent determine what information is relevant for long-term and short-term recall, thus better-aligning assessments with the intended educational outcomes.

Practical implications

The practical implications of this research, despite the identified limitations, are vast and transformative for educational settings.

A major benefit lies in the automation of assessments, which saves substantial time for teachers and educators. By generating questions and answers automatically, the model frees educators to focus on higher-level aspects of teaching and learning, such as the design of lesson plans, individual student mentoring, and professional development.

Additionally, the flexibility of these models allows for their application across a wide range of educational institutions and scenarios, such as elementary schools, high schools, and universities. The models could also find use in corporate environments, aiding in employee training and development programs.

Furthermore, the models empower students to take a more active role in their learning process. They can generate as many questions as they wish, enabling self-testing and reinforcing their understanding of the material. This is especially useful for remote and self-paced learning scenarios, which are common in today's digital education landscape.

Perhaps one of the most exciting prospects of these models is their potential to facilitate individualized learning. By generating focused questions and evaluating responses, they can provide personalized feedback and recommendations. This adaptability could help in meeting diverse learning styles and needs, paving the way for more personalized and effective teaching methods.

Originality/Value of the paper

The originality and value of this research lie in its innovative application of agent-based systems for educational assessment generation and evaluation, an area that, to the authors' knowledge, remains largely unexplored.

The paper demonstrates that the setup of such agents, using large language models like Vicuna, ChatGPT, GPT-4, and prompting techniques, is a feasible and promising approach. It sheds light on a novel path of employing advanced AI technologies to assist educators and learners, providing a blueprint for future developments in this field.

In conclusion, the innovative design of this system is highly promising for the future of automated learning and assessment tools. The combination of a comprehensive document processing approach, and advanced AI Agents including automatic refinement and quality assurance tasks, makes this utility a formidable tool in educational settings. It is worth noting that this advanced system also demonstrates the potential for adaptation and further improvements to meet evolving educational needs and challenges in the future.

REFERENCES

Botelho, A., Baral, S., Erickson, J. A., Benachamardi, P., & Heffernan, N. T. (2023). Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning*, *39*(3), 823-840 https://doi.org/10.1111/jcal.12793

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv:2005.14165 [Cs]*. http://arxiv.org/abs/2005.14165

Bulut, O., & Wongvorachan, T. (2022). Feedback generation through artificial intelligence. *The Open/Technology in education, society, and scholarship association conference*, *2*, Article 1.

Bulut, O., & Yildirim-Erba, S. N. (2022). Automatic story and item generation for reading comprehension assessments with transformers. *International Journal of Assessment Tools in Education*, *9*(Special Issue), 72 - 87. https://doi.org/10.21449/ijate.1124382

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., & Xing, E. P. (2023). *Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality*. https://lmsys.org/blog/2023-03-30-vicuna/

Emerson, A., Min, W., Azevedo, R., & Lester, J. (2022). Early prediction of student knowledge in Game-based learning with distributed representations of assessment questions. *British Journal of Educational Technology, 54*(1), 40-57. https://doi.org/10.1111/bjet.13281

Freeman, D., Katz, A., Gomez, P., & Burns, A. (2015). English-for-teaching: Rethinking teacher proficiency in the classroom. *Elt Journal*, *69*(2), 129-139. https://doi.org/10.1093/elt/ccu074

Gnanasekaran, D., Kothandaraman, R., & Kaliyan, K. (2021). An automatic question generation system using rule-based approach in bloom's taxonomy. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, *14*(5), 1477–1487.

Johnson, E. S., & Monroe, B. W. (2004). Simplified language as an accommodation on math tests. *Assessment for Effective Intervention*. https://doi.org/10.1177/073724770402900303

Kulshreshtha, D., Shayan, M., Belfer, R., Reddy, S., Serban, I., & Kochmar, E. (2022). Few-shot question generation for personalized feedback in intelligent tutoring systems. *ArXiv*, *abs/2206.04187*. https://doi.org/10.3233/FAIA220062

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1-35.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., & others. (2023). Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*. https://doi.org/10.1016/j.metrad.2023.100017

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint,* arXiv:2303.16634. https://doi.org/10.18653/v1/2023.emnlp-main.153

Liu, Z., He, X., Liu, L., Liu, T., & Zhai, X. (2023). Context matters: A strategy to pre-train language model for science education. *arXiv preprint,* arXiv:2301.12031. https://doi.org/10.48550/arxiv.2301.12031

Malinova, A., & Rahneva, O. (2017). Automatic generation of english language test questions using mathematica. *Cbu International Conference Proceedings*, *4*, 906-909. https://doi.org/10.12955/cbup.v4.794

Mishra, A. R., Panchal, V., & Kumar, P. (2020). Similarity search based on text embedding model for detection of near duplicates. *International Journal of Grid and Distributed Computing*, *13*(2), 1871–1881.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2002). Design and analysis in task-based language assessment. *Language Testing*, *19*(4), 477-496. https://doi.org/10.1191/0265532202lt241oa

Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy, C., Heidecke, J., Shyam, P., Power, B., Nekoul, T. E., Sastry, G., Krueger, G., Schnurr, D., Such, F. P., Hsu, K., … Weng, L. (2022). *Text and code embeddings by contrastive pre-training*. *arXiv preprint* arXiv:2201.10005. https://doi.org/10.48550/arXiv.2201.10005

Niraula, N. B., & Rus, V. (2015). Judging the quality of automatically generated gap-fill question using active learning. *Proceedings of the tenth workshop on innovative use of NLP for building educational applications*, 196–206. https://doi.org/10.3115/v1/W15-0623

OpenAI. (2023). GPT-4 technical report. *ArXiv*, *abs/2303.08774*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). *Training language models to follow instructions with human feedback*.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *arXiv preprint* arXiv:2304.03442. https://doi.org/10.1145/3586183.3606763

Runge, A., LaFlair, G. T., Yancey, K. P., Goodwin, S., Park, Y., & von Davier, A. A. (2022). The interactive reading task: Transformer-based automatic item generation. *Frontiers in Artificial Intelligence*, *5,* 903077. https://doi.org/10.3389/frai.2022.903077

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint* arXiv:1908.10084. https://doi.org/10.18653/v1/D19-1410

Sarsa, S. (2022). Automatic generation of programming exercises and code explanations with large language models. *ICER '22: Proceedings of the 2022 ACM Conference on International Computing Education Research, 1*, 27–43. https://doi.org/10.48550/arxiv.2206.11861

Sen Gupta, C., Datta, A. K., Datta, R., & Das, A. (2019). E-question paper generation system: A review. *International Journal of Electronics and Communication Engineering and Technology*, *10*(5). https://doi.org/10.34218/IJECET.10.6.2019.001

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., & others. (2023). Llama: Open and efficient foundation language models. *arXiv preprint* arXiv:2302.13971.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). Finetuned Language Models Are Zero-Shot Learners. *ArXiv:2109.01652 [Cs]*. https://doi.org/10.48550/arXiv.2109.01652

Xie, Y., Pan, L., Wang, D., Kan, M.-Y., & Feng, Y. (2020). Exploring question-specific rewards for generating deep questions. *arXiv preprint* arXiv:2011.01102. https://doi.org/10.18653/v1/2020.coling-main.228

Zou, B., Li, P., Pan, L., & Aw, A. T. (2022). Automatic True/False question generation for educational purpose. *Proceedings of the 17th workshop on innovative use of NLP for building educational applications (BEA 2022)*, 61–70. https://doi.org/10.18653/v1/2022.bea-1.10

APPENDIX

Prompts

Rate Paragraphs regarding the learning goal

> Please rate the following paragraph on a scale from 1 to 10, where 1 means that the paragraph is not relevant to the learning goal and 10 means that the paragraph is very relevant to the learning goal. Be precise in your rating. If you are not sure, please rate the paragraph with lower than 5. Give only integer values as rating, no text or explanation.
>
> Learning Goal: {{goal}}
>
> Paragraph: {{paragraph}}

Generate Questions:

> In order to check whether students have achieved the learning goal below, at least two questions must be generated for the given section with the aim of verifying the learning goal.
>
> Learning Goal: {{goal}}
>
> Section: {{section}}
>
> Questions:

Refine Questions:

> Refine the following questions to make them more precise and aligned with the learning goal. Ensure that the original meaning of the question is not changed. Make sure that the questions are not too broad and are focused on the learning goal. Ensure that no reference to figures is asked. If so, remove the question and respond with “NaN”. If the question is not relevant to the learning goal, also remove the question and response with “NaN”.: Make sure that no references to other chapters or code are made. If so, remove the question and respond with “NaN”.
>
> Learning Goal: {{goal}}
>
> Questions: {{question}}
>
> Refined Question:

Overall Rating for all Questions and Answers toward the learning goal

```
Given a dict of questions and answers, please rate the overall quality of this
set of questions and answers as sufficient to achieve the learning goal. Rate
this by a number from 1 to 10, where 1 means that the set of questions and
answers is not sufficient to achieve the learning goal, and 10 means that the
set of questions and answers is sufficient to achieve the learning goal. Be
precise in your rating. If you are not sure, please rate the set of questions
and answers lower than 5.

Describe the reasons for your rating in a few sentences. If the rating is
lower than 9 add 2 questions to close the gap.

Learning Goal: {{goal}}

Questions and Answers: {{qa}}
```

Prompt Evaluation of Student Answers:

```
'I give you the context, question and answer. Rate the answer by giving out
the probabilities for the classes in a new line, with the probablities between
0 and 1. The classes are - Contradiction, Neutral and Entailment. Also give
the reason for the rating.

Give me the output in a json string consisting of {"contradiction":value,
"neutral":value,"entailment": value, "reasoning_behind_grading": value}.

The reasoning must be in German containing the reason for the score given
and how the answer can be improved. Only json string output will be
accepted, no additional text. The sum of the probabilities of all the classes
should be 1.'
```

Example Output

```
############### File to be loadded ###############

data/example_ai.txt

Loaded data from file size: 43

############### Generate VectorStore ###############

############### Documents are Vectorized ###############

############### Search relevant Parts for LG #############
```

```
Learning Goal: As a Student, I would like to understand the interconnection between Agents and Environments!

Selected Documents: 10

------ Document ------

2.1:    Agents and Environments

An agent is anything that can be viewed as perceiving its environment t.... Score: 0.5231937766075134

------ Document ------

In Figure 2.5, we have sketched the basic PEAS elements for a number of additional agent types. Furt.... Score: 0.7983673214912415

------ Document ------

Chapter 1 identified the concept of rational agents as central to our approach to artificial intelli.... Score: 0.8482365608215332

------ Document ------

The code repository associated with this book (aima.cs.berkeley.edu) includes implementations of a.... Score: 0.8700109720230103

------ Document ------

Single agent vs. multiagent: The distinction between single-agent and multiagent environments may se.... Score: 0.8834455609321594

------ Document ------

We can imagine tabulating the agent function that describes any given agent; for most agents, this w.... Score: 0.9547677040100098

------ Document ------

2.3.2 Properties of task environments

The range of task environments that might arise in AI is obvio.... Score: 0.9721614122390747

------ Document ------

Now that we have a definition of rationality, we are almost ready to think about building rational a.... Score: 0.9741522073745728

------ Document ------

To illustrate these ideas, we use a very simple example—the vacuum-cleaner world shown in Figure 2.2.... Score: 0.9888737201690674
```

```
------ Document ------

Consider the simple vacuum-cleaner agent that cleans a square if it is dirty and moves to the other .... Score: 0.9888757467269897

############### Documents are selected ###############

Filtered Documents: 8

############### Documents are filtered ###############

############### Rate Paragraphs regards the learning Goal ###############

Paragraph: Chapter 1 identified the concept of rational agents as central to our approach to artificial intelli.... Score: 0.8482365608215332.... Rating GPT35: 9 Rating GPT4: 10

Paragraph: The code repository associated with this book (aima.cs.berkeley.edu) includes implementations of a.... Score: 0.8700109720230103.... Rating GPT35: 9 Rating GPT4: 9

Paragraph: Single agent vs. multiagent: The distinction between single-agent and multiagent environments may se.... Score: 0.8834455609321594.... Rating GPT35: 8 Rating GPT4: 8

Paragraph: We can imagine tabulating the agent function that describes any given agent; for most agents, this w.... Score: 0.9547677040100098.... Rating GPT35: 7 Rating GPT4: 9

Paragraph: 2.3.2 Properties of task environments

The range of task environments that might arise in AI is obvio.... Score: 0.9721614122390747.... Rating GPT35: 9 Rating GPT4: 8

Paragraph: Now that we have a definition of rationality, we are almost ready to think about building rational a.... Score: 0.9741522073745728.... Rating GPT35: 9 Rating GPT4: 9

Paragraph: To illustrate these ideas, we use a very simple example—the vacuum-cleaner world shown in Figure 2.2.... Score: 0.9888737201690674.... Rating GPT35: 9 Rating GPT4: 9

Paragraph: Consider the simple vacuum-cleaner agent that cleans a square if it is dirty and moves to the other .... Score: 0.9888757467269897.... Rating GPT35: 9 Rating GPT4: 8

############### Paragraphs are rated ###############

Filtered Documents: 5

############### Generate Questions ###############
```

Paragraph: Chapter 1 identified the concept of rational agent.... Score: 0.8482365608215332.... Rating GPT35: 9 Rating GPT4: 10 Questions GPT35: 1. What is the concept of rationality and how is it applied to agents operating in different environments?

2. How do properties of an environment influence the design of suitable agents for that environment?

Paragraph: The code repository associated with this book (aim.... Score: 0.8700109720230103.... Rating GPT35: 9 Rating GPT4: 9 Questions GPT35: 1. What is the purpose of the general-purpose environment simulator in the code repository associated with the book?

2. Why is it important to evaluate agents in multiple environments drawn from an environment class rather than just a single environment?

Paragraph: We can imagine tabulating the agent function that .... Score: 0.9547677040100098.... Rating GPT35: 7 Rating GPT4: 9 Questions GPT35: 1. What is the difference between the agent function and the agent program?

2. How can we construct the table that describes the agent function for a given agent?

Paragraph: Now that we have a definition of rationality, we a.... Score: 0.9741522073745728.... Rating GPT35: 9 Rating GPT4: 9 Questions GPT35: 1. What is a task environment and why is it important for building rational agents?

2. How does the flavor of a task environment affect the design of an agent program?

Paragraph: To illustrate these ideas, we use a very simple ex.... Score: 0.9888737201690674.... Rating GPT35: 9 Rating GPT4: 9 Questions GPT35: 1. What is the vacuum-cleaner world and why is it useful in illustrating the interconnection between agents and environments?

2. Can you explain the agent function described in Figure 2.3 and how it relates to the vacuum agent's actions in the environment?

############## Refine Questions ##############

Question: What is the concept of rationality and how is it applied to agents operating in different environments?.... Refined Question GPT35: How is the concept of rationality applied to agents in different environments and what is its significance in understanding the interconnection between agents and environments?

--

Question: How do properties of an environment influence the design of suitable agents for that environment?.... Refined Question GPT35: What factors of an environment should be considered when designing an agent to operate within it?

--

Question: What is the purpose of the general-purpose environment simulator in the code repository associated with the book?.... Refined Question GPT35: How does the general-purpose environment simulator in the code repository associated with the book facilitate the interaction between agents and environments?

--

Question: Why is it important to evaluate agents in multiple environments drawn from an environment class rather than just a single environment?.... Refined Question GPT35: How does evaluating agents in multiple environments drawn from an environment class help in understanding the interconnection between agents and environments?

--

Question: What is the difference between the agent function and the agent program?.... Refined Question GPT35: What is the distinction between the agent function and the agent program in terms of their roles in the interaction between agents and environments?

--

Question: How can we construct the table that describes the agent function for a given agent?.... Refined Question GPT35: What are the components of the agent function and how do they interact with the environment?

--

Question: What is a task environment and why is it important for building rational agents?.... Refined Question GPT35: What is the role of a task environment in the design and implementation of rational agents?

--

Question: How does the flavor of a task environment affect the design of an agent program?.... Refined Question GPT35: How does the type of task environment influence the development of an agent program?

--

Question: What is the vacuum-cleaner world and why is it useful in illustrating the interconnection between agents and environments?.... Refined Question GPT35: How does the vacuum-cleaner world illustrate the interconnection between agents and environments?

--

Question: Can you explain the agent function described in Figure 2.3 and how it relates to the vacuum agent's actions in the environment?.... Refined Question GPT35: NaN

--

############### Generate Answers ###############

[{‘question’: ‘How is the concept of rationality applied to agents in different environments and what is its significance in understanding the interconnection between agents and environments?’, …

############### Answers are generated ###############

############### Overall Rating if Learning Goal is achievable ###############

Overall Rating GPT4: Rating: 8

Reasons: The set of questions and answers provided covers a good range of topics related to the interconnection between agents and environments. It discusses the concept of rationality, factors to consider when designing an agent, the role of task environments, and the distinction between agent functions and agent programs. However, there are a few areas that could be improved to achieve a more comprehensive understanding of the topic.

Additional Questions:

1. What are some common types of environments, and how do their properties affect the design of agents?

2. Can you provide examples of different agent architectures and how they are suited for specific types of environments?