RESEARCH ON AN UTOMATIC MULTIPLE-CHOICE QUESTIONS GENERATION METHOD

Ngo Tien Duc
Posts and Telecommunications Institute of Technology

Abstract: The multiple-choice test format is becoming widespread for its convenience. However, manual multiple-choice questions (MCQs) generation is timeconsuming and costly. Therefore, automatic MCQs generation from texts has become a popular research area. Along with the growth of artificial intelligence (AI) in general and natural language processing (NLP) in particular, many large language models (LLMs) were developed with the ability of understanding text and processing information in high accuracy. Taking those advantages, this paper proposes a method on automatic MCQs generation using popular LLMs, ChatGPT and Gemini, in combination with a technique that has never been applied to this domain ReAct Agent. We evaluated the effectiveness of the proposed method by generating questions in Vietnamese for Operating System course of Posts and Telecommunications Institute of Technology. The conducted experiments shows that our method achieved the accuracy of 89%, a promising result to apply on other courses.

Keywords: Natural language processing, Multiple-choice questions, Large language models, ReAct Agent

I. INTRODUCTION

Nowadays, the application of multiple-choice test format in exams is becoming more and more popular in Vietnam and world while, thanks to their outstanding advantages compared to traditional essay tests. Instead of asking examiners to read pages of papers while the scoring is still intuitive, the evaluation process will become more consistent and time-saving when the answer can only be right or wrong. In addition, multiple choice tests can also be scored by machines to decrease evaluation time and increase accuracy. However, to create high quality MCQs requires a lot of effort from the examiners if it has to be done manually. The first efforts to find a solution to automatically generate MCQs in English were conducted in 1997 [1], since then many similar systems have been developed, applying in different domains and languages. With the rapid growth of NLP techniques, the perfection of those systems is increasingly enhanced, but there are still shortcomings mainly related to semantic and contextual issues, and the generated questions still need to be re-evaluated by candidates and experts.

In recent years, the emergence of LLMs has marked a new era in Generative AI. Tools that are able to perform linguistic tasks at human-level are no longer strange, which helps us save a lot of time and effort. For example, in the field of information technology, LLMs can handle difficult tasks such as generating programming code according to user requests [2] or explaining user-provided code [3], [4]. The reliability of the generated text can also be improved through techniques such as Chain of Thought Prompting (CoT) [5], which provides models with the ability of reasoning to give more accurate answers. Moreover, the ReAct Agent model [6], inspired by human learning and decision-making capabilities, is promising in leveraging the power of LLMs to an even higher level.

We studied and proposed a method to generate questions for multiple-choice tests automatically using LLMs combined with supporting techniques. First, the model is fed with specialized knowledge using retrieval augmented generation technique (RAG) [7], then ask the model to perform the task of generating MCQs with specific requirements using CoT, we continue to use ReAct Agent to ask the model to perform the evaluation and quality improvement so that each of the generated MCQs would be considered as the most complete version.

The remainder of the paper is organized as follows: In Section II, we review some related studies to the generation of MCQs and LLMs. The details of our proposed method are presented in Section III. Section IV describes the experimental results we conducted and gives some evaluations about the performance of the proposed method. Finally, in Section V, we present the conclusions, limitations, and future directions.

II. RELATED WORK

The techniques used in this paper are related to two main research directions, including automatic MCQs generation and LLMs.

Contact author: Ngo Tien Duc
Email: ducnt@ptit.edu.vn
Manuscript received: 25/9/2024, revised: 24/10/2024, accepted: 2/11/2024.

A. Automatic MCQs generation

A multiple-choice question consists of two main components: the stem that poses the problem to be solved, the correct answer (key) and the wrong answers (distractors). In this article, when referring to the phrase "MCQ", we should understand that it includes both the stem and the answers. Although the structure of an MCQ is simple, creating a multiple-choice test requires a lot of time and effort. A high-quality MCQ should have a clear question and good enough distractors to cause challenges for candidates. This is also the biggest challenge in the process of creating MCQs automatically using AI, requiring machine systems to understand correctly and clearly about the elements in the question in relation to data from the input text. In addition, many schools and educational institutions are using question banks for reusability. This may lead to a trick for students, they only memorize and remember the answers for all the questions instead of understanding knowledge from the lessons. Changing the order of answers is not an effective solution, all the questions need to be updated continuously, to avoid the cases where question banks are publicly announced or leaked, hence affecting the objectivity of the exam.

To address the above difficulties, several studies on automatic MCQs generation have been conducted. Starting from simple methods such as utilizing available templates [8]. In 2020, Dhawaleswar and Sujan [9] surveyed and listed strategies and techniques for sentence analysis, including machine learning techniques, applied in automatic MCQs generation and proposed a general strategy for this problem. After that, in 2021, more modern NLP techniques were published by Nwafor and Onyenwe [10]. In general, this automation process still faces challenges mainly related to processing speed and performance when the complexity of the input text is too large or the quality of the answers is noisy. In addition, there is no standard evaluation method or dataset to compare the effectiveness of those methods.

Building a complete automated MCQs generation system would certainly be extremely expensive, along with the investment requirements for training dataset. Recently, LLMs have demonstrated their superior performance, so we came up with the idea of leveraging their strength in creating MCQs for exams.

B. Large language models

LLMs are neural networks trained on huge textual datasets, allowing computers to perform human-like NLP tasks. GPT-3 [11] can be considered the first LLM as it outperformed other models at that period in both size and performance. To enable LLMs to learn from human instructions, InstructGPT [12] was created to collect feedback and prompts from users. This is the predecessor of GPT-3.5, the core of ChatGPT - LLM that has attracted a lot of attention since its release in 2022. One year later, GPT-4 [13] was introduced as the new and most powerful multi-modal LLM of the GPT family, capable of processing both text and multi-media inputs.

While recent GPT models, since GPT-3, have become closed source and only accessible via API, Meta has released LLaMA [14] as an open source platform for LLMs. These models are pre-trained on trillions of tokens, collected from publicly available datasets. In July 2023, Meta and Microsoft jointly released LLaMA-2 [15]. Due to its open source nature, the LLaMA family is widely used by research groups and is growing rapidly in number, including Giraffe [16], Long LLaMA [17], Code LLaMA [18], etc.

Another multi-modal language model, Gemini [19], is also famous for its performance in understanding text, image, audio, and video. Gemini was introduced by Google Deepmind in 2023, with three versions: Ultra for highly complex tasks, Pro for large-scale deployment with high performance, and Nano for on-device apps.

There have also been numerous research on applying LLM to create multiple-choice tests. For example, Andrew Tran and colleagues [20] used GPT-3 and GPT-4 for C programming exam. A research group at universities in the US [21] studied a combination of several LLMs with few-sample learning methods and CoT to generate MCQ for their machine learning course, based on question banks and course materials, resulting in questions with content that is completely different from the old ones, and very difficult to distinguish from questions that are written by humans.

However, most of the popular LLMs work efficiently in English only, there are fews studies that applied LLMs for other languages including Vietnamese, due to concerns about the fluency and naturalness of the questions, depending on the translation ability of the model. While [22] only tested and compared the ability to find answers to MCQs in Physics between ChatGPT and Microsoft Bing AI Chat with accuracy of 61% and 66% respectively, or [23] fine-tuned the ChatGPT model to create MCQ banks for universities with an average accuracy only about 58.6%. Both studies only limited at the level of using chatbots and their performance was normal. The method we proposed will be based on requiring the model to inference and act sequentially, not only automatically generating questions but also carefully evaluating, checking, and modifying stems and answers to create the most complete questions.

III. PROPOSED METHOD

A. ReAct Agent

Our main objective when working with LLMs is to make the models give the best answers that align with specific requirements. Currently, there are many methods to support users to interact with LLMs quickly and efficiently, including CoT - a technique that provides a way to reason in small steps as same as human-ways to solve problems, then asks the model to follow, so that the results are more accurate and reliable. However, with the suggestion of CoT, a model without a foundation of external knowledge will sometimes have difficulty if it only uses pre-trained knowledge to create arguments, limiting the ability to explore and update knowledge like humans. And the ReAct Agent architecture was born to solve this problem. First introduced in [6], ReAct is a technique that combines "Reasoning" and "Acting", enhances the understanding and information processing capabilities of LLMs by generating reasoning traces and task-specific actions, allowing the model to interact with external data sources to gain additional information for question answering. ReAct can be used in conjunction with other methods to improve reasoning and action capabilities, supporting contextual decision making rather than strictly following pre-determined scenarios. The components of ReAct Agent include LLMs, information gathering tools, CoT for reasoning, and ReAct Prompting - a technique that guides LLMs to generate reasoning and acting traces. To our knowledge, there is no research applying ReAct Agent to generate MCQs until now.

B. Applying LLMs and ReAct Agent in automatic MCQs generation

To generate MCQs for a specific domain, i.e. a subject or a course in education, the model needs to be familiar with specialized knowledge about that subject, hence the input includes a document D as study materials. Generally, when an agent receives a request P at time step t, it will take an action $a_t \in A$ following policy $\pi(a_t|c_t)$, where A is the action space, c_t is the context that is trained to find a way to map to the corresponding a_t. Lacking specialized data from A may lead to inaccurate mapping. ReAct performs an augmentation of the action space to $A' = A \cup L$ where L is the language space. In our method, L is the thought or reasoning trace to generate MCQs from the knowledge in D, specifically in the form of instructions to find the topic from the request and create suitable questions, evaluate and update if the question is not good enough, even delete and recreate bad questions, etc. The way to perform a new action a'_t will be inferred based on the current context c_t, this process is supported by the large language model M. After performing a'_t, the context will be updated with $c_{t+1} = (c_t, a'_t)$ to support future reasoning or acting.

$$a'_t = M(c_t, D, P) \text{ with } a'_t \in A \cup L$$

We applied RAG to augment specialized data for the model. RAG is a technique to improve the accuracy of language models by retrieving information from external sources and documents. Although LLMs have strong capabilities to answer most questions posed by humans, for information that requires high accuracy, especially in scientific domains, the model needs to rely on specialized knowledge, and RAG acts as an assistant for the model's reinforcement learning process to be able to give more reliable answers. After feeding the input dataset to the model, we use CoT to ask the model to perform the task of generating MCQs. The initial question generation may not be good enough, e.g. duplicate answers, wrong key identification or unreasonable questions. By leveraging the question answering capabilities of LLMs demonstrated in [24], we continue to use ReAct Agent to require the model to evaluate and improve the quality of questions.

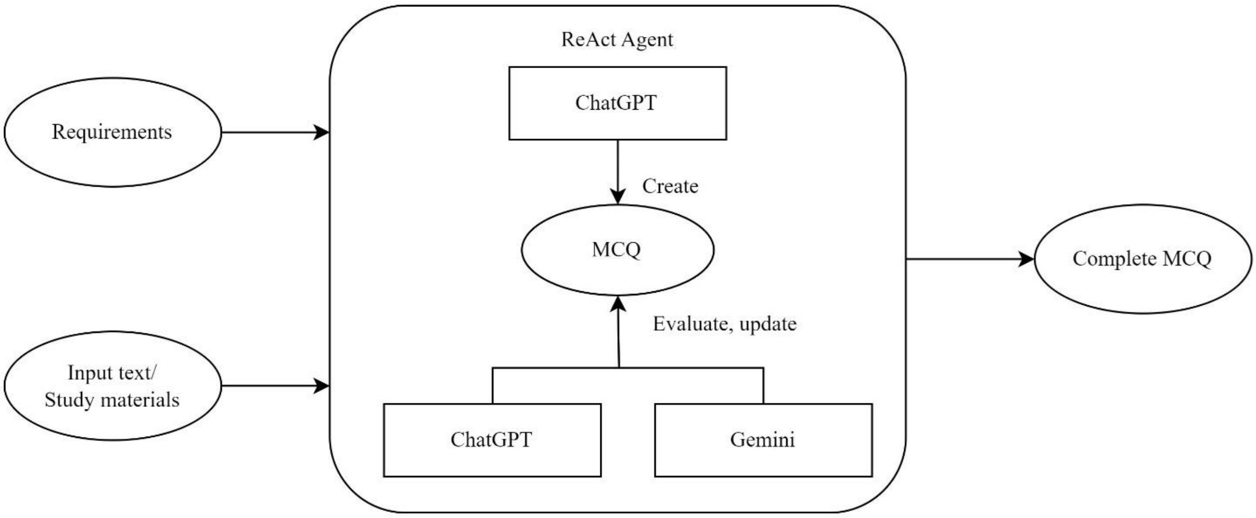

Fig. 1: Proposed automatic MCQ generation model with ReAct Agent

Figure 1 describes the process of ReAct Agent receiving and processing actions in order, input contains user requirements and study materials for augmentation, output is the most complete version of the generated MCQ. The components of ReAct Agent include a question generation tool using ChatGPT, two question evaluation tools using ChatGPT and Gemini, in which we use CoT to build thinking logic for Agent based on the basic idea of creating MCQs and then evaluating them twice, at each evaluation, modifying the question if necessary, in case evaluators think the question is bad and can not be updated, then create another question. Specifically, the process of creating a complete MCQ is as follows:

- User enters a request to create a question, and can add instructions to help the model better define the purpose of the task. For example: "You are an expert

in creating MCQs, from the input text, create MCQs", or you can ask more specifically "based on the content of section 3, chapter 4, create an MCQ about pagination", etc.
- ReAct Agent receives the request and proceeds to feed the request to the ChatGPT model to generate the first version of the question.
- The generated question is sent to the second ChatGPT model to perform an evaluation, e.g. checking the correct answer, if there is a problem with the question or answer, performs modifications to make them reasonable. If there is no problem, then moves to the next step.
- The generated question is sent to the Gemini model to perform evaluation one more time as same as above. If quality of the generated question is too low, ask the model to re-create from the beginning then repeats above steps until the result is good enough.
- Return the most complete MCQ to user.

Through the above steps, we believe that the model will be able to generate questions with high quality by taking advantage of the questionanswering capabilities of LLMs. Besides, using multiple models to evaluate each other will be more objective than using only a single model and reduce the incidence of "hallucination".

IV. EXPERIMENT

A. Dataset and evaluation metrics

In this paper, we use dataset taken from the textbook of "Operating Systems" courses, which is currently used in the university education program of Posts and Telecommunications Institute of Technology. This is a fundamental course for information technology students, providing detailed knowledge about the concepts and components of operating systems. The textbook has 177 pages of content in total, divided into 4 chapters including: general introduction (32 pages) - providing basic concepts about operating systems and computer system hardware, process management (52 pages) - including knowledge related to processes and process scheduling, memory management (42 pages) - introducing knowledge about paging and page changing strategies, and file system (51 pages) - including concepts about files, directories and ways to organize and allocate space for files. With diverse and theoretical knowledge, applying multiple-choice exam to this subjects is reasonable to assess the learning process of students, and also suitable for us to consider the quality of Vietnamese questions generated by LLMs from the proposed method. Since there is currently no standard metric to evaluate the quality of automatic generated MCQs, we manually collected assessments from lecturers and students. The criteria we set include:

- Accuracy: The model generates questions with different answers and only one of those answers is the key. Besides, the key answer must be indicated correctly.
- Distinction: There is no semantic overlap or duplication between generated answers (key and distractors).
- Alignment: The generated question align with the user's input requirements, e.g. match specified topic, number of answers,... And the question must be answerable, i.e. the key must be able to find from the input document.
- Difficulty: Ratio of difficult questions - questions with distractors which confuse students, or finding the answer requires inference and knowledge synthesis from candidates.

B. Experimental results

Our experiments were conducted on a Windows 11 computer, Intel Core I5 10300H CPU, 16GB RAM, NVIDIA GTX 1650 GPU. We applied the proposed method to generate 200 MCQs from the initial dataset. The approximate time for generating a complete MCQ is about 5 to 10 seconds. Figure TABLE I: Quality assessment results of automatic MCQs generations

Criteria	ReAct Agent	ChatGPT	Gemini
Accuracy	89%	80%	77%
Distinction	72%	64%	59%
Alignment	66%	57%	61%
Difficulty	40%	37%	36%

2 is an example of the reasoning and acting process of ReAct Agent to generate an MCQ in Vietnamese. It can be seen that the reasoning and acting process is carried out in detail, which help to increase the reliability of the answer. In terms of quality, the stem and answers are quite similar to those written by humans, in which the final results have also been automatically truncated to become more concise and natural. We also experimented by asking ChatGPT and Gemini to generate questions independently with the same input dataset, in order to compare with our proposed method. We calculated the average rate for each evaluation criteria and obtained the results as shown in Table 1.

C. Discussion

The results show that the proposed method with ReAct Agent is capable of generating MCQs better than dependent component models in all criteria, especially in terms of accuracy. There are many questions that cannot be searched directly on the Internet to find the answer. However, the accuracy rate is not absolute, generated questions should be double-checked by lecturers. In the process of selecting questions to include in the exam, the lecturer can perform further updates depending on the purpose of increasing the ease or difficulty of the questions. Obviously, using this method can save a lot of time and effort. In addition, noisy answers are being generated based on the input dataset, from which there are opinions that we should take advantage of other sources of knowledge outside the textbook to increase the richness of the answers. We will consider this issue and find a solution to be able to build the most complete tool in supporting the creation of quality multiple-choice tests.

V. CONCLUSION

The paper proposes a method of applying LLMs for automatic MCQs generation. We combine popular LLMs including ChatGPT and Gemini with RAG, CoT, ReAct Agent techniques to improve the efficiency of generating questions in the most optimal way and ensure objectivity in mutual evaluation between models. Experimental results on the Operating System course at the Posts and Telecommunications Institute of Technology show that our method achieves an accuracy up to 89%. Although double-check should be performed to ensure the quality of the questions, with the above performance, our method will be helpful to save time and effort in the process of creating multiple-choice exams.

For the future work, we plan to experiment on other LLMs to find the model that may work more effectively. In addition, expanding the application to documents of other courses is also necessary to evaluate in more detail the reliability of the proposed method. Another limitation is that our model has not handled questions that containing complex calculation formulas. Our research only works with the generation of individual MCQ, while creating a complete multiple-choice test is more complicated in terms of selecting questions with appropriate difficulty. Through preliminary survey, we found that LLMs are not yet able to accurately assess the difficulty of questions, but we can train them on Bloom's taxonomy [25] to solve this problem. This is also a promising research direction that we are interested in and plan to investigate further studies in the future.

```
Prompt: Tạo 1 câu hỏi trắc nghiệm có nội dung liên quan đến "Tiến trình" trong hệ điều hành. Sau đó sử dụng công cụ kiểm tra lại.
Thought: Tôi cần sử dụng công cụ để giúp tôi tạo câu hỏi trắc nghiệm.
Action: Create
Action Input: {'input': "Tạo câu hỏi trắc nghiệm về 'Tiến trình' trong hệ điều hành."}
Observation: Câu hỏi: Tiến trình là gì trong quản lý tiến trình của hệ điều hành?
A. Tiến trình là một chương trình đang trong quá trình thực hiện
B. Tiến trình là một chương trình tĩnh không thay đổi theo thời gian
C. Tiến trình là một chương trình chỉ có thể thực hiện một tác vụ duy nhất cùng một lúc
D. Tiến trình là một chương trình không thể chạy song song với các chương trình khác
Đáp án đúng: A. Tiến trình là một chương trình đang trong quá trình thực hiện
Thought: Tôi cần sử dụng công cụ để kiểm tra lại câu hỏi trắc nghiệm vừa tạo.
Action: Check
Action Input: {'input': 'Câu hỏi: Tiến trình là gì trong quản lý tiến trình của hệ điều hành?\nA. Tiến trình là một chương trình đang trong quá trình thực hiện\nB. Tiến trình là một chương trình tĩnh không thay đổi theo thời gian\nC. Tiến trình là một chương trình chỉ có thể thực hiện một tác vụ duy nhất cùng một lúc\nD. Tiến trình là một chương trình không thể chạy song song với các chương trình khác\nĐáp án đúng: A. Tiến trình là một chương trình đang trong quá trình thực hiện'}
Observation: Lời đánh giá: Câu hỏi đã được cung cấp đầy đủ thông tin và chính xác về khái niệm tiến trình trong quản lý tiến trình của hệ điều hành. Câu hỏi có thể được viết lại rõ ràng và ngắn gọn hơn: Trong quản lý tiến trình của hệ điều hành, tiến trình được định nghĩa như thế nào? A. Một chương trình đang trong quá trình thực hiện B. Một chương trình tĩnh không thay đổi theo thời gian C. Một chương trình chỉ có thể thực hiện một tác vụ duy nhất cùng một lúc D. Một chương trình không thể chạy song song với các chương trình khác Đáp án đúng: A. Một chương trình đang trong quá trình thực hiện.
Thought: Tôi có thể đưa ra câu hỏi trắc nghiệm mà không cần sử dụng thêm bất kỳ công cụ nào.
Answer: Câu hỏi: Trong quản lý tiến trình của hệ điều hành, tiến trình được định nghĩa như thế nào?
A. Một chương trình đang trong quá trình thực hiện
B. Một chương trình tĩnh không thay đổi theo thời gian
C. Một chương trình chỉ có thể thực hiện một tác vụ duy nhất cùng một lúc
D. Một chương trình không thể chạy song song với các chương trình khác
Đáp án đúng: A. Một chương trình đang trong quá trình thực hiện.
```

Fig. 2: Example process of ReAct Agent to generate an MCQ in Vietnamese

REFERENCES

[1] D. Coniam, "A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests," *CALICO Journal*, vol. 14, 01 1997.

[2] B. A. Becker, P. Denny, J. Finnie-Ansley, A. LuxtonReilly, J. Prather, and E. A. Santos, "Programming is hard – or at least it used to be: Educational opportunities and challenges of ai code generation," 2022. [Online]. Available: https://arxiv.org/abs/2212.01020

[3] S. MacNeil, A. Tran, A. Hellas, J. Kim, S. Sarsa, P. Denny, S. Bernstein, and J. Leinonen, "Experiences from using code explanations generated by large language models in a web software development e-book," 2022. [Online]. Available: https://arxiv.org/abs/2211.02265

[4] J. Leinonen, P. Denny, S. MacNeil, S. Sarsa, S. Bernstein, J. Kim, A. Tran, and A. Hellas, "Comparing code explanations created by students and large language models," in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE 2023. ACM, Jun. 2023. [Online]. Available:

http://dx.doi.org/10.1145/3587102.3588785

[5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-ofthought prompting elicits reasoning in large language models," 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023. [Online]. Available: https://arxiv.org/abs/2210.03629

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler,¨ M. Lewis, W. tau Yih, T. Rocktaschel,¨ S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledgeintensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[8] A. Ezz and M. Dahab, "Automatic generation of question bank based on pre-defined templates," 01 2014.

[9] D. R. CH and S. K. Saha, "Automatic multiple choice question generation from text: A survey," *IEEE Transactions on Learning Technologies*, vol. 13, no. 1, pp. 14–25, 2020.

[10] C. A. Nwafor and I. E. Onyenwe, "An automated multiple-choice question generation using natural language processing techniques," *International Journal on Natural Language Computing*, vol. 10, no. 02, p. 1–10, Apr. 2021. [Online]. Available: http://dx.doi.org/10.5121/ijnlc.2021.10201

[11] T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[12] L. Ouyang, J. Wu, X. Jiang *et al.*, "Training language models to follow instructions with human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[13] OpenAI, J. Achiam, S. Adler *et al.*, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[15] H. Touvron, L. Martin, K. Stone *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[16] A. Pal, D. Karkhanis, M. Roberts, S. Dooley, A. Sundararajan, and S. Naidu, "Giraffe: Adventures in expanding context lengths in llms," 2023. [Online]. Available: https://arxiv.org/abs/2308.10882

[17] S. Tworkowski, K. Staniszewski, M. Pacek, Y. Wu, H. Michalewski, and P. Miłos, "Focused transformer:´ Contrastive training for context scaling," 2023. [Online]. Available: https://arxiv.org/abs/2307.03170

[18] B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, R. Sauvestre, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. C. Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve, "Code llama: Open foundation models for code," 2024. [Online]. Available: https://arxiv.org/abs/2308.12950

[19] G. Team, R. Anil, S. Borgeaud *et al.*, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[20] A. Tran, K. Angelikas, E. Rama, C. Okechukwu, D. H. Smith, and S. MacNeil, "Generating multiple choice questions for computing courses using large language models," in *2023 IEEE Frontiers in Education Conference (FIE)*, 2023, pp. 1–8.

[21] I. Drori, S. J. Zhang, R. Shuttleworth, S. Zhang, K. Tyser, Z. Chin, P. Lantigua, S. Surbehera, G. Hunter, D. Austin, L. Tang, Y. Hicke, S. Simhon, S. Karnik, D. Granberry, and M. Udell, "From human days to machine seconds: Automatically answering and generating machine learning final exams," 2023. [Online]. Available: https://arxiv.org/abs/2206.05442

[22] D. Xuan-Quy, L. Ngoc-Bich, P. Xuan-Dung, N. Bac-Bien, and V. The-Duy, "Evaluation of chatgpt and microsoft bing ai chat performances on physics exams of vietnamese national high school graduation examination," 2023. [Online]. Available: https://arxiv.org/abs/2306.04538

[23] D. H. Quan, N. T. Thuc, and B. M. Hiep, "A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 10, no. 3, pp. 1646–1649, 01 2024.

[24] E. Kamalloo, N. Dziri, C. L. A. Clarke, and D. Rafiei, "Evaluating open-domain question answering in the era of large language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.06984

[25] S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban, "How teachers can use large language models and bloom's taxonomy to create educational quizzes," 2024. [Online]. Available: https://arxiv.org/abs/2401.05914

NGHIÊN CỨU PHƯƠNG PHÁP SINH CÂU HỎI TRẮC NGHIỆM TỰ ĐỘNG

Tóm tắt — Hình thức trắc nghiệm đang được áp dụng rộng rãi vì tính tiện lợi của nó. Tuy nhiên, quá trình tạo ra các câu hỏi trắc nghiệm theo cách thủ công đòi hỏi không ít thời gian và chi phí. Vì vậy, tự động sinh các câu hỏi trắc nghiệm từ văn bản đã trở thành một lĩnh vực nghiên cứu phổ biến. Cùng với sự phát triển của trí tuệ nhân tạo nói chung và xử lý ngôn ngữ tự nhiên nói riêng, nhiều mô hình ngôn ngữ lớn (LLM) đã được phát triển với khả năng đọc hiểu văn bản và xử lý thông tin với độ chính xác cao. Tận dụng những ưu điểm đó, bài báo này đề xuất một phương pháp sinh câu hỏi trắc nghiệm tự động sử dụng các LLM quen thuộc là ChatGPT và Gemini, kết hợp với một kỹ thuật mới chưa từng được áp dụng trong lĩnh vực này - ReAct Agent. Nhóm nghiên cứu thực hiện đánh giá độ hiệu quả của phương pháp đề xuất bằng cách áp dụng cho tạo các câu hỏi trắc nghiệm tiếng Việt cho môn Hệ điều hành của Học viện Công nghệ Bưu chính Viễn thông. Kết quả thực nghiệm tiến hành cho thấy phương pháp đạt độ chính xác 89%, một kết quả đầy hứa hẹn để áp dụng cho các môn học khác.

Từ khóa— Xử lý ngôn ngữ tự nhiên, câu hỏi trắc nghiệm, mô hình ngôn ngữ lớn, ReAct Agent

Ngo Tien Duc MSc., lecturer in the Faculty of Information Technology, Posts and Telecommunications Institute of Technology.