ComPAT: A Compiler Principles Course Assistant

Shubin Cai[1,2](✉), Honglong Chen[1], Youyi Huang[1], and Zhong Ming[2,3]

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
shubin@szu.edu.cn

[2] Laboratory of Artificial Intelligence and Digital Economy (Shenzhen), Shenzhen, China
mingz@szu.edu.cn

[3] Shenzhen Technology University, Shenzhen, China

Abstract. Students in compiler principle course often encounter difficulties in understanding abstract programming concepts and need first-aid assistants in the context of independent learning. An innovative Large Language Model-based assistant ComPAT (**Com**piler **P**rinciples course **A**ssis**T**ant) is proposed in this paper. ComPAT uses vector database for text segmentation and similarity queries, meta prompting for answer refinement and challenging questions for heuristic learning. A compiler principles domain-specific dataset with 1520 questions is constructed. Compared to ChatGPT3.5, ERNIE3.5, and ChatGLM4, ComPAT has shown superior performance in handling fill-in-the-blank and question-and-answer. ComPAT demonstrates an innovative and promising use of LLM-based teaching/learning assistant system in computer education.

Keywords: Computer Education · Large Language Model · Teaching/Learning Assistant

1 Introduction

Compiler principles is a core course in the computer science curriculum [1]. Students aspiring to learn how to build, maintain, and execute a compiler for a modern programming language, often encounter difficulties in learning and understanding abstract concepts in this course. It is very helpful to provide teaching/learning assistants to answer students' questions in a timely manner. The emergence of Large Language Models and their applications makes it possible to construct an intelligent chatbot to serve as an always-online teaching/learning assistant.

The compiler construction course is crucial for developing students' programming skills. However, its theoretical components, such as formal language, automata theory, and syntax-directed translation, are very complex and abstract. This complexity makes

Supported by Guangdong Higher Education Teaching Research and Reform Project of China JG2019023, Guangdong Province Key Laboratory of Popular High Performance Computers 2017B030314073 and the National Natural Science Foundation of China No. 62276168

C. Cao et al. (Eds.): KSEM 2024, LNAI 14888, pp. 74–83, 2024.
https://doi.org/10.1007/978-981-97-5489-2_7

the material challenging for students to grasp, especially in the context of independent learning.

Modern technologies like LLMs can significantly enhance compiler principles course teaching. An LLM-powered teaching/learning assistant can provide instant aid, personalized guidance, and help with complex compiler concepts [2]. It can also support programming practice and offer resources to deepen students' understanding. Vector databases can serve as external knowledge databases for large language models, and using appropriate organization methods can enhance the learning assistance capabilities of large models.

An LLM-based learning assistant, ComPAT (*Compiler Principles course AssisTant*) is proposed in this paper. Compared to ChatGPT3.5, ERNIE3.5, and ChatGLM4, ComPAT has shown superior performance in answering 1,520 compiler domain-specific questions. ComPAT can also stimulate students to learn more by asking relevant questions. This demonstrates a promising use of an LLM-based teaching/learning assistant system in computer education.

2 Related Work

2.1 Large Language Models

With the rapid development of cloud computing [3–5] and big data [6–8] technologies, machine learning [9–11] and *Artificial Intelligence* (AI) [12–15] has become a promising approach in many application areas. GPT-like LLMs are pivotal in NLP research, propelling advancements in language understanding and generation. *Pre-trained Language Models* (PLMs) leverage the self-attention mechanism and the Transformer framework to learn language in an unsupervised manner. Larger PLMs with more parameters achieve better performance, which has led to the development of LLMs such as GPT-3 (175 billion parameters), LLaMA (65 billion parameters), and PaLM (540 billion parameters). These models demonstrate advanced capabilities like contextual learning and step-by-step reasoning, further enhanced by the *Chain of Thought* (CoT) approach for few-shot learning scenarios [16–20].

Applying LLMs to specific domain is challenging due to varied data and complex knowledge. Specialization is crucial, with recent progress on domain-specific tech via fine-tuning and targeted incremental pre-training [21]. For examples, LaWGPT, tailored for Chinese legal texts, and BianQue, built on Chinese medical Q&A data and fine-tuned with ChatGLM-6B [22].

LLMs can generate false information (hallucinations), which is risky in critical areas like education. Recent research shows that knowledge-enhanced LMs can improve accuracy by using external databases to curb hallucinations.

2.2 LLMs in Education

LLMs are transforming education by offering interactive, personalized learning experiences. They foster deeper understanding and knowledge mastery by simulating dialogues and posing questions, aiming to improve education quality [23].

LLMs like GPT-4 are proving valuable in education through applications such as Khan Academy's Khanmigo project [24], which provides personalized AI-assisted teaching experiences and aids teachers with administrative tasks. East China Normal University's EduChat [25], an LLM integrating multidisciplinary resources, aims to deliver intelligent and empathetic education, showing gains in assessments, tutoring, and emotional support.

LLMs in education face challenges including accuracy issues and potential overreliance, which may affect educational quality. GPT-4 and similar models could generate errors, and students might use them as shortcuts, bypassing deep learning. To maximize LLM benefits, educators should address emotional and interactive aspects, and continually refine these tools for better outcomes.

3 System Framework of ComPAT

At the start of the compiler principles course, a dedicated WeChat group was established to facilitate the exchange of inquiries between students and instructors. Given the complexity inherent in compiler principles, the frequency of student inquiries escalated, thereby posing challenges for instructors to provide timely responses. In response to this issue, a WeChat mini-app, named ComPAT, was developed utilizing LLM-based Retrieval-augmented Generation (RAG) technology. Initially, instructors upload domain-specific documents pertinent to compiler principles into ComPAT's backend management system, thereby augmenting the LLM capability to furnish domain-specific responses. Students are afforded the opportunity to consult ComPAT at their convenience. ComPAT generates responses based on its specialized knowledge base supplemented by LLM's broader knowledge spectrum. Instructors subsequently monitor the dialogues between students and ComPAT, addressing inaccuracies in the responses in a timely manner. Furthermore, feedback regarding erroneous queries and responses is treated as domain-specific knowledge and is integrated into the ComPAT backend management system, which refines the accuracy of future responses provided by ComPAT.

The following sections outline ComPAT's operations, detailing how user queries are processed, accurate answers obtained, and corrections made for incorrect outputs, as depicted in Fig. 1.

3.1 Document Segmentation and Indexing

Domain-specific documents are transformed into vector embeddings and stored in a vector database called Milvus. The Milvus vector database is divided into two sections: one for FAQs in (question, answer) format, and another for domain-specific document slices. We use the book *Compilers: Principles, Techniques, and Tools* as the documentation for slicing. We specify that the length of the question and answer library for each slice is 500 characters, and the number of overlapping characters between two slices is 100.

When queries diverge from existing FAQs, the document fragments module is activated. Top 8 relevant text slices are selected from the database by considering both similarity algorithms and content diversity to the vector representation of the query. The LLM then comprehensively summarizes these slices to answer the query.

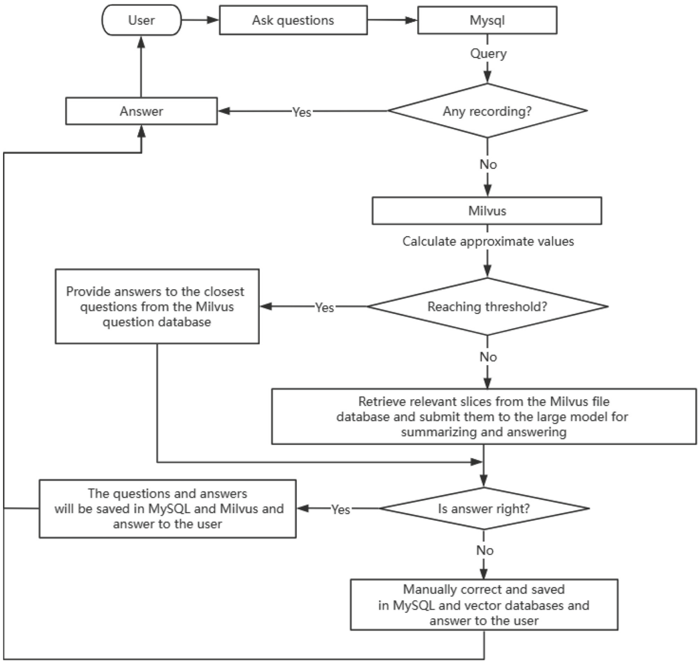

Fig. 1. System Framework of ComPAT

3.2 Establishing Context in Questions and Crafting Effective Prompts

Designing a relevant context is crucial for leveraging LLMs' ability to provide consistent, logical responses that meet students' specific learning requirements.

We craft a rich, detailed context for LLMs by combining background information and professional terms, particularly focusing on fundamental compiler principles concepts like "lexical analysis," "syntax analysis," and "semantic analysis," to enhance understanding and structure knowledge for better question comprehension.

LLMs require command-following and step-by-step reasoning to accurately address complex queries. Effective prompts and clear contexts are essential for guiding LLMs to produce precise and useful answers. Utilizing Chain of Thought and few-shot learning, we help LLMs dissect complex Compiler Principles questions by identifying key terms and data, ensuring precise answers. We also create tailored prompts with diverse problems and cases to align closely with educational goals, as shown in Fig. 2, improving teaching and learning outcomes.

```
question_prompt_template = """You are a compiler principles teacher, proficient in compiler principles.
Use the following portion of a long document to see if any of the text is relevant to answer the question.
Return any relevant text verbatim.Must response in Chinese.
{context}
Question: {question}
Relevant text, if any:"""
```

```
combine_prompt_template = """
Given the following extracted parts of a long document and a question, create a final answer with references ("来源").
If you don't know the answer, just say that you don't know. Don't try to make up an answer.Must response in Chinese.
ALWAYS return a "来源" part in your answer.

QUESTION: Noun explanation: prefix
=========
Content: Prefixes and suffixes are very important concepts in data structures and compilation principles. In data structures,
prefixes and suffixes are a common algorithm used to quickly calculate the sum of a certain interval in an array. In the principle
of compilation, prefixes and suffixes refer to the prefixes and suffixes used in lexical analysis to identify words.
Source: 28-pl
Content: In data structures, prefix and suffix are common algorithms, where prefix and refer to the sum of the first i elements in
an array, while suffix and refer to the sum of the last elements in an array. These two algorithms can be implemented through dynamic
programming with a time complexity of O (n). The specific implementation method is to first calculate the prefix and array, as well as
the suffix and array, and then subtract to calculate the sum of any interval.
Source: 30-pl
Content: Appearing in a combined form, connected to a sound or consecutive sounds at the beginning of a word, root, or phrase, or
used in writing to produce a new word or variation. In Chinese, it refers to the word forming element before the root of a word.
Source: 4-pl
=========
FINAL ANSWER: Prefix - refers to any beginning of a symbol string.
```

Fig. 2. Context and Prompt

3.3 Meta-Prompting

Inspired by Meta-Prompting [26], we've adopted a method to enhance the large model's answer accuracy by assigning question types, revising responses, and standardizing answers in a three-step process. Initially, the model generates an answer from text; then it reviews this answer without the text for accuracy; finally, it standardizes the answer, omitting extra explanations. Figure 3 shows the prompt.

```
content = """
            You are a Meta Expert, an advanced expert in compilation principles. Other experts may not be able to
            determine the answer or the answer may be incorrect. You need to review the answers and corresponding
            explanations provided by other experts on compilation principles, make corrections to incorrect answers
            and explanations, and provide answers to questions that cannot be determined to ensure that all questions
            are effectively answered. Please carefully consider and review step by step, and provide the final
            certification answer.
            """
content = """
            You are an advanced expert in compilation principles, and you need to review the answers to compilation
            principles questions to ensure that all questions can be effectively answered and that the answer format
            is accurate. That is, the answer to multiple-choice questions should only contain one option, and the answer
            to multiple-choice questions should only be true or false. Please think carefully and carefully, conduct a
            review, and provide the final certification answer.
            """
```

Fig. 3. The second and third step of Meta-Prompting

3.4 Feedback and Adjustment

To address the inherent limitations of the model and gaps in database information, our system incorporates both preprocessing and postprocessing steps, along with an expanded feedback mechanism to enhance performance and accuracy.

In the preprocessing step, when a user poses a question, the system first searches the MySQL database, which contains accurate, preset questions and answers. This step ensures quick responses and avoids uncertainties associated with the model.

The postprocessing focuses on correcting inaccurate answers generated by the LLM. Instructors can intervene based on student feedback, modifying and updating questions and answers. These revised contents are stored in both MySQL and Milvus, improving response quality and continuously optimizing database accuracy.

Additionally, instructors can input new materials into the vector database's text block section. These materials are segmented into logical paragraphs or thematic blocks around core concepts, ensuring knowledge updates and dynamic.

These strategies ensure that the system remains accurate and efficient in handling queries while continuously self-improving to meet evolving educational needs.

3.5 User Interface

The WeChat mini-app interface of ComPAT and its core features are shown in Fig. 4. Students can ask questions and await responses from the system. ComPAT provide a probable correct answer or indicates its lack of knowledge for students (threshold is set by instructor) and generate a pending review for instructor. Pending review is manual inspected and updated by instructors in the "Review Center".

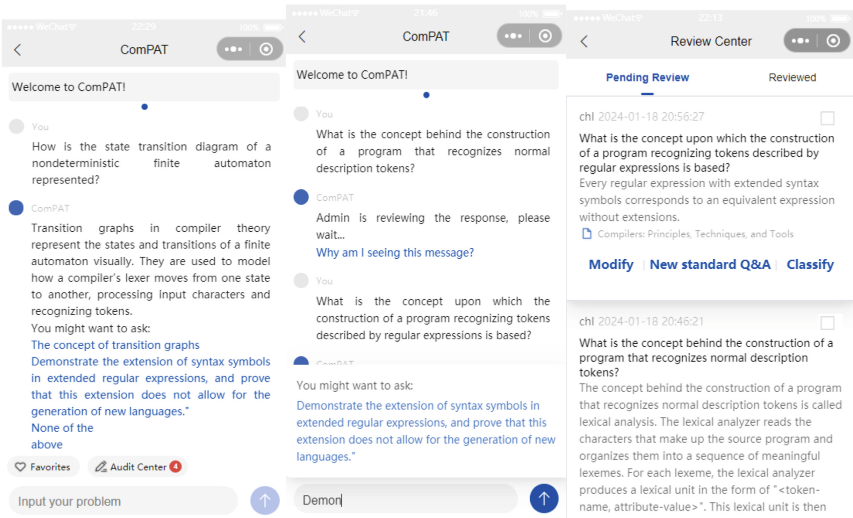

Fig. 4. Q&A interface

Automatically matched Q&A pairs are considered correct and used to respond to users directly. For questions not matched to existing pairs, the system uses the LLM to search the vector database and provide a summary answer, marked as unreviewed for instructor verification. Instructors can then decide whether to save these Q&A pairs in

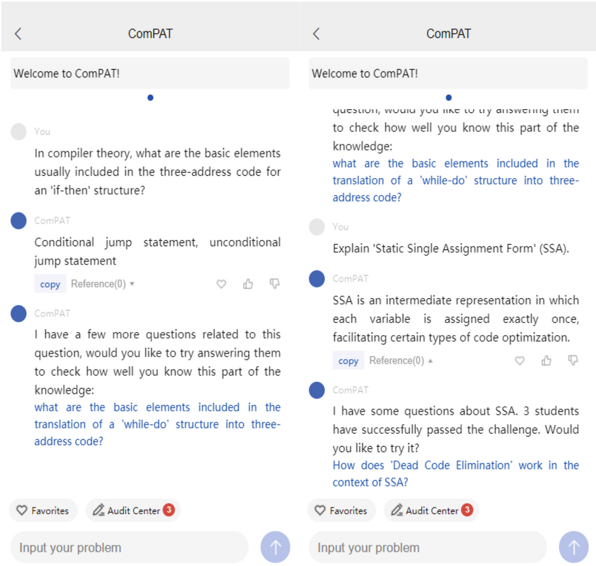

Fig. 5. Challenge and Inspection

the database or just answer the user query, maintaining flexibility. They also have access to the source of the answers for further credibility assurance.

The system also includes two heuristic learning modes as shown in Fig. 5. The inspection mode focuses on specific themes and raise relevant questions for students to answer. The challenge mode presents tougher questions to challenge students to answer. Figure 6 shows ComPAT's backend management system. These include the ability to

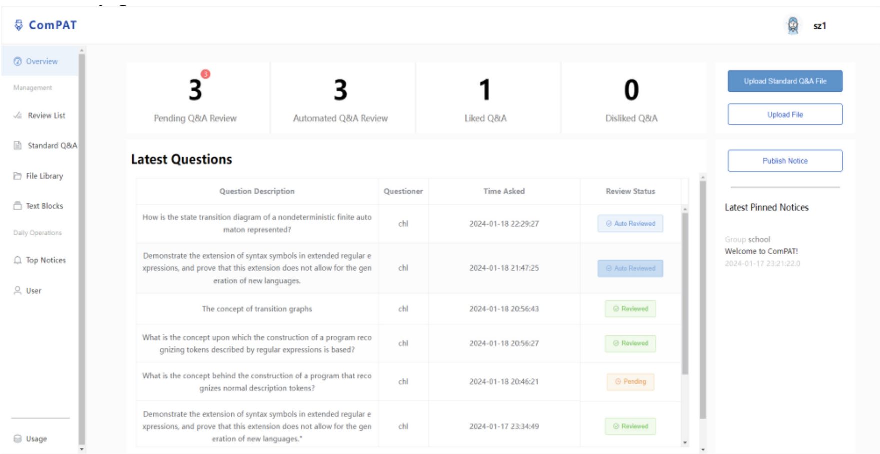

Fig. 6. Overview of ComPAT's backend management system.

input new question-answer pairs, upload domain-specific files and process inaccuracy generated answers.

4 Experiments and Results

In this section, a compiler principle domain-specific dataset is constructed and results from ComPAT, ERNIE-3.5, ChatGPT-3.5 and ChatGLM-4 is presented.

4.1 Dataset Construction

To verify whether ComPAT can improve the model's ability to answer compile principle questions, we construct a diverse dataset of 1520 questions, categorized into four types: 520 true/false, 520 single-choice, 160 fill-in-the-blank, and 320 short-answer questions. The data in this dataset is mutually exclusive with the data stored in the vector database.

The dataset was created using public documents collected from the internet, transformed into questions, and further enhanced with GPT-4 generated items. This approach ensures a comprehensive coverage of essential concepts and technologies in compiler principles. The required outputs for these questions varied: true/false responses for the respective questions, complete options for single-choice, comma-separated answers for fill-in-the-blanks, and answers with over 80% cosine similarity to the dataset answers for short-answer questions. Especially for fill-in-the-blank and short-answer questions, the evaluation criteria are set to at least 80% cosine similarity between the model's response and the data set's answers. This metric was chosen to ensure a rigorous assessment of the models' ability to generate detailed and contextually accurate responses.

4.2 Comparative Analysis with Other Leading LLMs

To comprehensively evaluate ComPAT's performance in the compiler principles domain, we conducted an extensive comparative analysis with other mainstream LLMs, including ERNIE 3.5, ChatGPT-3.5, and ChatGLM-4. The comparison used the same dataset and consistent testing methodologies to ensure fairness and comparability in experimental conditions (Table 1).

Table 1. Performance comparison of LLMs in compiler principles domain

Model/Type	Judgment	Choice	Fill-in-the-Blank	Short Answer
ERNIE-3.5	0.4000	0.5385	0.1687	0.0531
ChatGPT-3.5	0.5634	0.6500	0.1689	0.0500
ChatGLM-4	**0.6346**	**0.6673**	0.2875	0.1094
ComPAT	0.6327	0.5656	**0.3500**	**0.2750**

The experimental results demonstrate that ComPAT has higher accuracy and stability in processing Compiler Principles-related datasets compared to base model ChatGPT.

Specifically, ComPAT improved accuracy by 6.93%, 18.11%, and 22.50% in true/false, fill-in-the-blank, and Q&A questions respectively. This indicates ComPAT's superior performance in the compiler principles domain. However, it performed less effectively in multiple-choice questions, likely due to interference from options when retrieving text blocks via VDB, leading to selection errors.

It's important to consider that variations in performance across question types can be attributed to the unique challenges each type presents. For instance, short-answer questions require concise, comprehensive responses, aligning well with ComPAT's design. ComPAT's strengths may lie in its precision and approach to specific compiler principles issues, crucial in real-world applications where output quality is prioritized.

Comparative analysis revealed that ChatGPT also exhibited errors in handling compiler principles datasets, possibly due to insufficient understanding of the subject or dataset imperfections. In contrast, ComPAT showed greater accuracy and stability, benefiting from optimizations and adjustments made during training.

In summary, ComPAT shows potential for compiler principles education and self-study, highlighting the benefits of targeted AI training in niche fields. Future versions will build on these insights to enhance its efficacy as the go-to model for compiler principles and similar areas. We will also continue to monitor the role of ComPAT in practical teaching environments, observe how it improves students' problem-solving abilities, and whether it will cause dependency.

5 Conclusion

Following an extensive study and empirical assessment, insights into ComPAT, an innovative teaching/leaning assistant for compiler principles course have been expanded. ComPAT leverages vector database for text segmentation and indexing, and finely tuned large language models, enhancing content delivery and the learning experience. Experiments show that ComPAT excels in domain-specific tasks, particularly in understanding and applying professional knowledge.

Future work may aim to improve ComPAT by advancing its structure and training, particularly for compiler multiple-choice questions and complex compiler principles. Exploring new retrieval-augmented generation, optimization tactics and integrating ComPAT with models like ERNIE 3.5 and ChatGPT may yield a superior educational resource, merging multimodal learning and nuanced language comprehension with deeper architectural and algorithmic enhancements.

Our research offers insights for compiler principles teaching reform and showcases the potential of advanced tech in education. We aim to evolve ComPAT into a smarter, more efficient teaching aid, poised to boost student learning in compiler principles and drive educational tech innovation.

References

1. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools, 2nd edn. Pearson Education, New Delhi (2006)

2. Lixiang, Y., Lele, S., et al.: Practical and ethical challenges of large language models in education: a systematic literature review, CoRR, abs/2303.13379 (2023)
3. Qiu, M., Dai, W., Vasilakos, A.: Loop parallelism maximization for multimedia data processing in mobile vehicular clouds. IEEE Trans. Cloud Comput. **7**(1), 250–258 (2016)
4. Gai, K., Xu, K., et al.: Fusion of cognitive wireless networks and edge computing. IEEE Wirel. Commun. **26**(3), 69–75 (2019)
5. Zeng, Y., Qiu, H., Memmi, Gd., Qiu, M.: A data augmentation-based defense method against adversarial attacks in neural networks. In: Qiu, M. (ed.) ICA3PP 2020, pp. 274–289. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60239-0_19
6. Zhang, Y., Qiu, M., et al.: Health-CPS: healthcare cyber-physical system assisted by cloud and big data. IEEE Syst. J. **11**(1), 88–95 (2015)
7. Wei, X., Guo, H., et al.: Reliable data collection techniques in underwater wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **24**(1), 404–431 (2021)
8. Qiu, M., Qiu, H.: Review on image processing based adversarial example defenses in computer vision. In: IEEE 6th BigDataSecurity (2020)
9. Qiu, H., Zheng, Q., et al.: Toward secure and efficient deep learning inference in dependable IoT systems. IEEE Internet Things J. **8**(5), 3180–3188 (2020)
10. Cui, Y., Cao, K., et al.: Client scheduling and resource management for efficient training in heterogeneous IoT-edge federated learning. In: IEEE TCAD (2021)
11. Song, Y., Li, Y., et al.: Retraining strategy-based domain adaption network for intelligent fault diagnosis. IEEE Trans. Ind. Inform. **16**(9), 6163–6171 (2019)
12. Zhang, Y., Qiu, M., Gao, H.: Communication-efficient stochastic gradient descent ascent with momentum algorithms. In: IJCAI (2023)
13. Ling, C., Jiang, J., et al.: Deep graph representation learning and optimization for influence maximization. In: ICML (2023)
14. Zeng, Y., Pan, M., et al.: Narcissus: a practical clean-label backdoor attack with limited information. In: ACM CCS (2023)
15. Li, C., Qiu, M.: Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies. CRC Press, Boca Raton (2019)
16. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
17. Touvron, H., Lavril, T., Izacard, G., et al.: LLaMA: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
18. Chowdhery, A., Narang, S., Devlin, J., et al.: Palm: scaling language modeling with pathways. J. Mach. Learn. Res. **24**(240), 1–113 (2023)
19. Kojima, T., Gu, S.S., et al.: Large language models are zero-shot reasoners. In: NeurIPS (2022). http://papers.nips.cc/paper_files/paper/2022/hash/
20. Wei, J., Wang, X., Schuurmans, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Adv. Neural Info. Proc. Syst. **35**, 24824–24837 (2022)
21. Fu, Y., Peng, H., et al.: Complexity-based prompting for multi-step reasoning. In: ICLR 2023. OpenReview.net (2023). https://openreview.net/pdf?id=yf1icZHC-l9
22. Guu, K., Lee, K., et al.: Retrieval augmented language model pre-training. In: ICML 2020, vol. 119 of Proceedings of Machine Learning Research, pp. 3929–3938. PMLR (2020)
23. Extance, A.: ChatGPT has entered the classroom: how LLMs could transform education. Nature **623**, 474–477 (2023)
24. Sahlman, W.A., Ciechanover, A.M., Grandjean, E.: Khanmigo: Revolutionizing Learning with GenAI, pp. 824–059. Harvard Business School Case, November 2023
25. Dan, Y., Lei, Z., Gu, Y., et al.: EduChat: a large-scale language model-based Chatbot system for intelligent education. arXiv (2023)
26. Suzgun, M., Kalai, A.T.: Meta-prompting: enhancing language models with task-agnostic scaffolding (2024)