

Large language models in software engineering: a systematic review and vision

Nguyen Van Viet*

Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam

Nguyen The Vinh

Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam

***Corresponding Author:** nviet@ictu.edu.vn

Keywords

Large Language Models
Software Engineering
PRISMA
Recurrent Neural Network
Artificial Intelligence

Article History

Received 2024-09-25

Accepted 2024-11-19

Copyright © 2024 by Author(s).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

Abstract

Large Language Models (LLMs) are very large deep learning models pre-trained on a vast amount of data. This article aims to provide an overview of the use of major language models in the field of software engineering from January 2021 to February 2024. It surveys the emerging area of Large Language Modeling in Software Engineering but acknowledges that to fully understand the issues, effects, and limitations of LLMs in this field, further research is needed. The article also highlights open research challenges for applying Large Language Models to technical problems faced by software engineers. The exceptional properties of LLMs bring novelty and creativity to applications within Software Engineering activities, including coding, design, requirements, repair, refactoring, performance improvement, documentation, and analytics. Our survey demonstrates the key role of reliable and efficient large language models in the development and deployment of Software Engineering.

INTRODUCTION

Large language models (LLMs) are very large deep learning models, pre-trained on huge amounts of data. The basic convolutional set consists of neural networks that have an encoder and a decoder with self-focusing capabilities. Encoders and decoders extract meaning from a string of text and understand the relationships between words and phrases within it. LLMs are capable of unsupervised training, with the main mechanism being self-learning. Through this process, the model learns to understand grammar, language, and basic knowledge. Unlike previous recurrent neural networks (RNNs) that typically process input data sequentially, the transformer processes the entire sequence in parallel. This allows data scientists to use GPUs to train transformer-based LLMs, significantly reducing training time.

This paper aims to conduct a systematic review of research on large language models (LLMs) in software engineering. Software developers and software testers can utilize LLMs during the development and construction phases. The use of LLMs in software engineering presents an opportunity for software development companies and research groups to deploy quickly, thereby minimizing human and financial resources.

Software engineering (SE) focuses on developing and building computer software and application software. This process includes a series of activities from requirements gathering and analysis to system design, implementation, and testing. Using LLMs in the field of software engineering helps reduce human effort in the 4.0 industrial era. LLMs can be utilized to solve tasks in software engineering, such as analyzing data, code, or text.

GPT-4 is an OpenAI LLM that performs strongly in several software engineering tasks, including coding, comprehension, execution, and reasoning. It handles real-world applications and diverse cryptographic challenges while demonstrating the ability to interpret results in natural language and generate code from descriptions.(Fu et al. 2023a)

To address the above problems, research groups and scientists have highlighted the potential, opportunities, difficulties, and challenges of using LLMs in the field of software engineering. This is demonstrated through numerous general and applied research articles related to LLMs in software engineering. For instance, the research team led by Mark Chen and colleagues, in 2023, conducted studies such as "Evaluating Code-Trained Large Language Models (Chen et al. 2021a)," "Evaluating Code-Trained Large Language Models (Fu et al. 2023a)," "Evaluating the Usability of Code Generation Tools Provided by Large Language Models (Vaithilingam, Zhang, and Glassman 2022a)," and "Automatically Generate Programming Exercises and Code Explanations Using Large Language Models." (Sarsa et al. 2022).

Therefore, this article aims to address the above problem by analyzing bibliographies of scientific articles on the use of LLMs in the field of software engineering:

Question 1: What are the most important research topics in the use of LLMs in the field of software engineering?

Question 2: What are the top 10 most influential articles in this research field?

Question 3: How and from where is the data collected?

Question 4: What methods and algorithms are used in LLMs in the field of software engineering?

Question 5: What are the gaps and areas for future research?

Answering these research questions will help software developers gain basic perspectives on approaching LLMs in the field of software engineering. Additionally, new researchers can identify future research directions through the identified research gaps. Review the key concept you use in the research and provide previous relevant studies/investigations that are relevant to your paper. Review the key concept you use in the research and provide previous relevant studies/investigations that are relevant to your paper. Review the key concept you use in the research and provide previous relevant studies/investigations that are relevant to your paper. Review the key concept you use in the research and provide previous relevant studies/investigations that are relevant to your paper. Review the key concept you use in the research and provide previous relevant studies/investigations that are relevant to your paper.

METHODS

The article employs the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) general research method [5]. PRISMA provides a framework for conducting systematic and transparent analyses of scientific articles, assisting researchers in evaluating the reliability of studies before incorporating them into their own research.

Search sources

This study primarily used Google Scholar as the main search source for data collection. Google Scholar offers flexible search capabilities, allowing access to digital copies of articles online. This flexibility not only facilitates the rapid collection of information but also helps ensure the diversity and richness of data sources. Consequently, the study achieved comprehensiveness in evaluating and synthesizing information from diverse sources available on Google Scholar.

Search criteria

The author selects articles for general analysis from journal and conference databases based on the following criteria:

1. Must have from Large language Models (LLMs) and Software Engineering(SE)
2. Related to Large language models
3. Related to Software Engineering

The article data collection period is from January 2021 to 2024. Using the above criteria, Google Scholar returned results with 4,750 articles included in the review.

Conditions for including articles in analysis

To be included in the final list for analysis and evaluation, articles need to meet some additional requirements as follows:

1. Time: article published from 2021 to present (2024)
2. Language: article is written in English and Vietnamese
3. Accessibility: the article is accessible in full text
4. Articles with one of the following elements will be removed from the list:
 - a. Not an article (book, thesis, poster, introduction page...)
 - b. The article is not written in English or Vietnamese
 - c. Article published before 2021

Figure 1 depicts the flow of information through the different stages of the systematic review process using the PRISMA approach. 4750 records were found in the search data source from Google Scholar, 3010 articles before 2020 were removed, leaving 1740 articles. Next, there were 1250 duplicate articles based on title, records that were not articles, not in English or Vietnamese, and not related to LLMs were also removed. 335 articles whose full text was not accessible due to access restrictions were then also removed. The authors examined the remaining 125 articles and eliminated 82 articles due to inappropriate research content. Finally, 43 articles were included in this study for evaluation and analysis.

Figure 1 depicts the flow of information through the different stages of system assessment and the use of the PRISMA method.

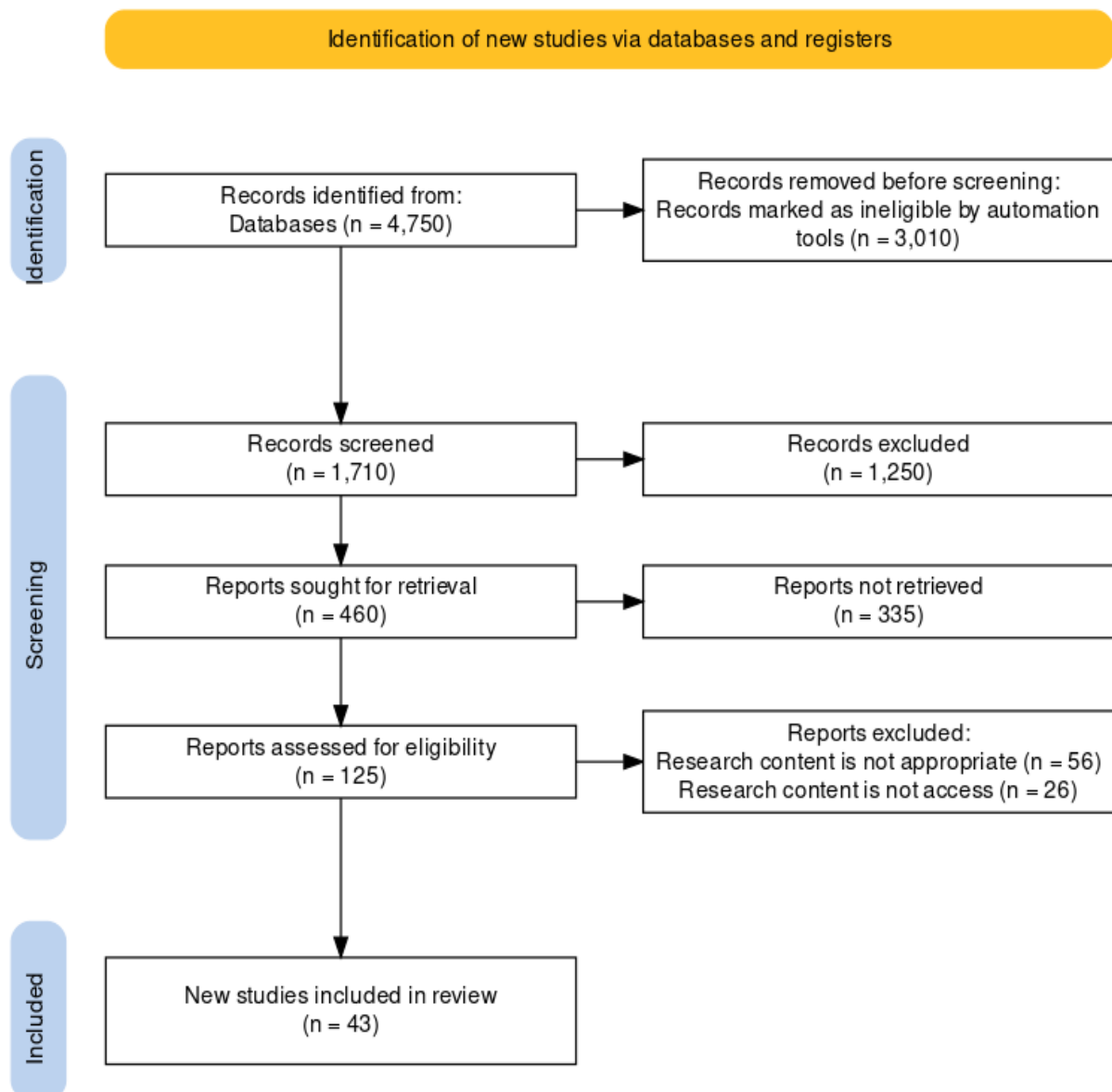



Figure 1. The diagram shows the movement of information through the different stages of a systematic review

RESULTS AND DISCUSSION

What are the most important research topics in the use of LLMs in the field of software engineering?

Figure 2 shows the most frequently occurring keywords such as Large Language Models with a total of 116 times; Software Engineering with 24 times; Challegen, design, GPT... with 20 times; genetic improvement appears 16 times; Some related keywords between LLMs and SE appear in decreasing numbers.

Create Map ×

 **Verify selected keywords**


Selected	Keyword	Occurrences	Total link strength 
<input checked="" type="checkbox"/>	large language models	32	116
<input checked="" type="checkbox"/>	software engineering	4	24
<input checked="" type="checkbox"/>	challenges	2	20
<input checked="" type="checkbox"/>	design	2	20
<input checked="" type="checkbox"/>	gpt	2	20
<input checked="" type="checkbox"/>	llm-assisted	2	20
<input checked="" type="checkbox"/>	requirements	2	20
<input checked="" type="checkbox"/>	state-of-the-art	2	20
<input checked="" type="checkbox"/>	validation	2	20
<input checked="" type="checkbox"/>	verification	2	20
<input checked="" type="checkbox"/>	genetic improvement	2	16
<input checked="" type="checkbox"/>	code generation	5	14
<input checked="" type="checkbox"/>	automated program repair	1	12
<input checked="" type="checkbox"/>	documentation generation;generative ai	1	12
<input checked="" type="checkbox"/>	human computer interaction	1	12
<input checked="" type="checkbox"/>	open-source	2	12
<input checked="" type="checkbox"/>	refactoring	1	12
<input checked="" type="checkbox"/>	requirements engineering	1	12
<input checked="" type="checkbox"/>	search based software engineering (sbse)	1	12
<input checked="" type="checkbox"/>	software analytics	1	12

Figure 2 . Keywords were extracted from abstracts of articles related to Large Language Models and Software Engineering

Top 10 most influential articles in the research field?

From Table 1, it is evident that the research article "Evaluating Large Language Models Trained on Code," published on arXiv preprint by Mark Chen and his colleagues, is the most cited and popular among scientists, with 2,453 citations in the Google Scholar database [1]. This highlights the significance and impact of their work in this field. Other notable articles on LLMs in software engineering, which have also garnered substantial citations, include "A Systematic Evaluation of Large Language Models of Code," "Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models," "Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models," and "Promptchainer: Chaining Large Language Model Prompts Through Visual Programming." In contrast, the remaining articles by other authors have citation counts of 0 or 1, indicating limited interest in their work.

Table 1 . The top articles with the most citations in the database are related to Large language models

Article name	Total	Highly Influential	Background (%)	Methods (%)
Evaluating Large Language Models Trained on Code(Chen et al. 2021)	2453	21,2%	54,2%	42,7%
Emergent abilities of large language models (Wei et al. 2022a)	1262	6,70%	67,40%	10,90%
Extracting training data from large language models	1065	8,90%	73,30%	16,70%
A systematic evaluation of large language models of code(Xu et al. 2022)	326	8,30%	54,60%	32,80%
A survey on evaluation of large language models(Chang et al. 2023)	264	6,40%	71,60%	17,80%
A survey on large language model based autonomous agents(Wang et al. 2023)	252	8,30%	70,20%	20,20%
SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models(Xiao et al. 2023a)	227	17,20%	55,90%	47,60%
Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models(Vaithilingam, Zhang, and Glassman 2022a)	198	6,60%	38,90%	14,60%
Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models(Vaithilingam, Zhang, and Glassman 2022a)	156	4,50%	55,10%	20,50%
Promptchainer: Chaining large language model prompts through visual programming(Wu et al. 2022)	125	6,40%	53,60%	28,80%

How is the data collected and from where?

In research, data plays a crucial role. Aggregated data is collected from various sources to ensure the model can handle diverse situations and scenarios. It also clarifies the training goals and identifies errors during training. Pre-processing the data helps standardize it and improve the quality of the models.

Finally, the data is formatted into appropriate structures for processing, allowing the large language model to understand the features and performance of the data patterns. The data collection process is carried out through the following steps.

Where does the training data of large language models come from?

Data is an important factor in the training of big data language models, it determines the importance, efficiency, and performance of the models. Data completeness, accuracy, and diversity are extremely important for language models to understand the characteristics and patterns of data usage, optimize parameters, and ensure reliability in data usage. implementation and testing process.

First, regarding the synthesis method to obtain data, using analytical methods during the research process, data sources are divided into specific types as follows: first, open source data are data sets. publicly accessible through open source platforms or repositories; Second, the collected data are data sets researched from many different sources through different presentations such as websites, forums, applications...; Third, data are built by asking questions whose answers are appropriate to their research direction; Fourth, industrial data are data sets obtained from commercial or industrial organizations that often contain proprietary business data, user behavior logs, or other confidential information. In summary, the above four data types all have their own advantages and disadvantages, so their use in the author's research needs needs to be flexible and appropriate.

What types of data sets in the field of software engineering are used in large language models?

Data types play a crucial role in shaping the architecture and selection of large language models (LLMs), as they directly influence the definition of modeling tasks and decisions. Therefore, the choice of data type impacts the overall performance and flexibility of LLMs. Various solutions exist to classify data types based on criteria related to LLMs and software engineering (SE). According to researched publications, data types include: code-based, text-based, chart-based, software repository-based, and combined data types.

In practice, over 80% of studies use text data. Collecting text data leverages strengths in training LLMs for SE tasks, enhancing their ability to handle the model's natural language logic. These LLMs develop capabilities in understanding and processing text data, making them ideal for code-related tasks, debugging, code generation, and other SE challenges.

What methods and algorithms are used in LLMs in the field of SE?

Through the process of analyzing the data set of articles, the methods and algorithms used in LLMs in the field of SE are shown in the chart:

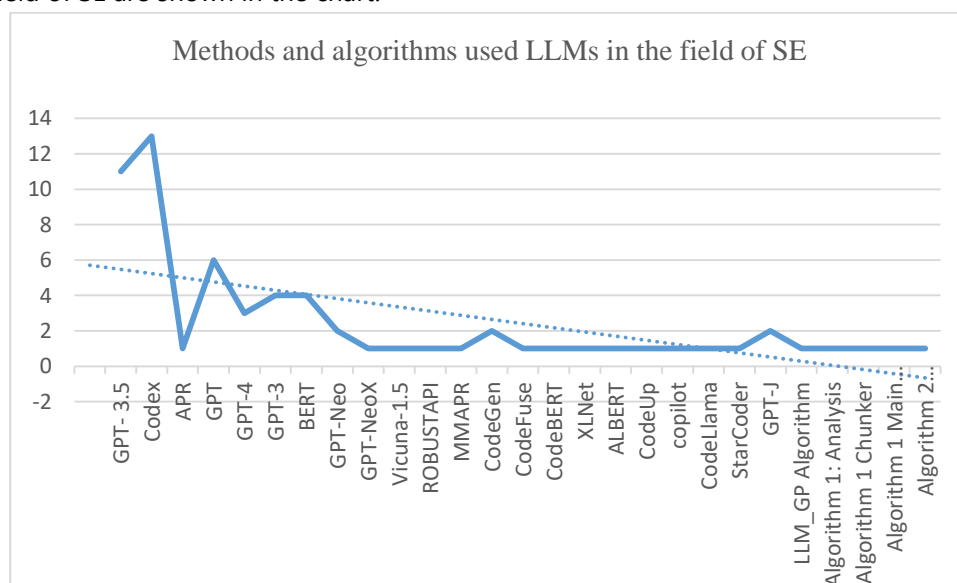


Figure 3 . The methods and algorithms used in the papers are related between Large Language Models and Software Engineering

Quantity by year:

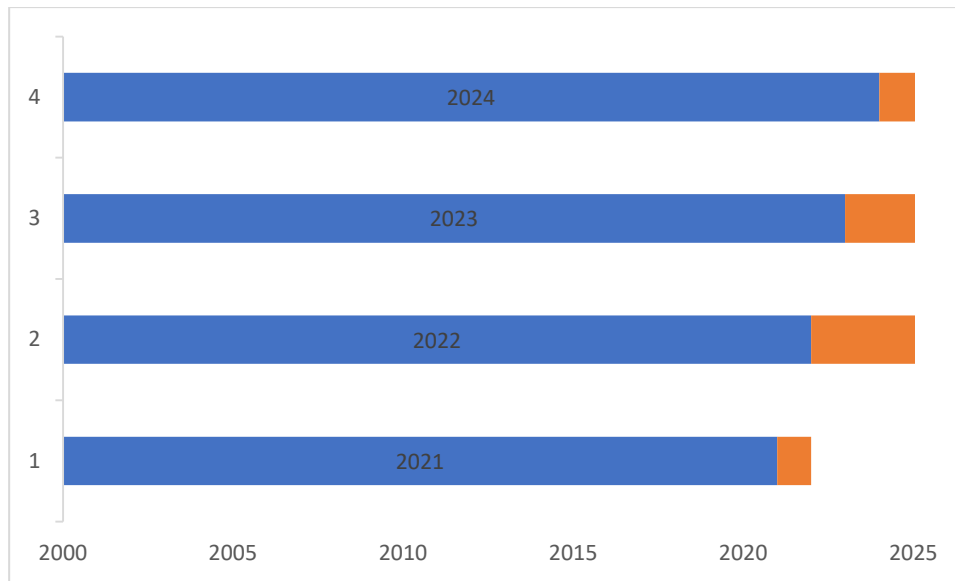


Figure 4 . The methods and algorithms used in the papers are related between Large Language Models and Software Engineering

What are the gaps and areas for future research?

Large language models have been researched at a moderate level from 2021 to 2024 but by 2023 have gradually developed strongly , but there are still some limitations and challenges that need to be overcome. Table 2 shows the frequency with which limitations and challenges of Large language models are stated in publications.

Limitations/challenges	Reference publications
Does not capture certain security	(Sallou, Durieux, and Panichella 2023; Wei et al. 2022b)
Do not generalize and program other languages	(Wei et al. 2022b)
Memory storage	(Xiao et al. 2023b)
Quality evaluation	(Fan et al. 2023; Zheng et al. 2023)
Misleading content	(Kiesler, Lohr, and Keuning 2023; Liu et al. 2023; Vaithilingam, Zhang, and Glassman 2022b)
Approach	(Du et al. 2023; Pan et al. 2023; Thakur et al. 2023)
Generate complex robot behavior remains relatively unexplored.	(Singh et al. 2023)
Issues related to benchmarking, method design and functional extension	(Zhang et al. 2023)
Reproducibility concerns	(Sallou, Durieux, and Panichella 2023)
Other datasets	(Li et al. 2023)

Overreaction to the potential threat of technology	(Makridakis, Petropoulos, and Kang 2023)
Missing data	
Automate data processing and processes	(Fan et al. 2023; Fu et al. 2023)
Test, verify, check data	(Lu et al. 2023)

Based on the provided information, it can be confirmed that the topic "Large Language Models in Software Engineering: A Systematic Review and Vision" is emerging as a new and pressing research direction. This is supported by the substantial rise in the adoption of large language models across various fields of information technology, particularly in software development. A comprehensive understanding of how LLMs can be integrated and utilized in the software development process will bring significant benefits to both the industry and the research community. This paves the way for promising opportunities to optimize software development processes and enhance the quality of final products.

CONCLUSION

This article focuses on research on Large Language Models (LLMs) in the field of software engineering, aiming to analyze and evaluate articles and documents related to the use of LLMs from 2021 to 2024. The research results indicate that the number of studies on LLMs was low from 2021 to 2022, comprising 17% of the total. However, this number surged to 32 articles in 2023, representing 74%, and the first three months of 2024 saw four articles, accounting for 9% in areas such as software engineering, testing, and ChatGPT. The most prominent publications are from "arXiv preprint," followed by "IEEE," with the highly influential article "Evaluating Large Language Models Trained on Code" being cited 2,453 times in the field of software engineering. In terms of LLMs research in various fields, including engineering, ChatGPT, and education, the USA leads in the number of publications and conferences. However, overall assessments of LLMs research in fields like software engineering and ChatGPT have not been extensively analyzed by authors. This review serves as a foundation for identifying future research gaps and directions in the development of LLMs research in software engineering.

REFERENCES

- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., et al. (2023). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Ponde de Oliveira Pinto, H., Kaplan, J., Edwards, H., et al. (2021). Evaluating large language models trained on code. *arXiv preprint*, arXiv:2107.03374.
- Du, X., Liu, M., Wang, K., Wang, H., Liu, J., Chen, Y., Feng, J., et al. (2023). Classeval: A manually-crafted benchmark for evaluating LLMs on class-level code generation. *arXiv preprint*, arXiv:2308.01861.
- Fan, Z., Gao, X., Mirchev, M., Roychoudhury, A., & Tan, S. H. (2023). Automated repair of programs from large language models. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)* (pp. 1469–1481). IEEE.

- Fu, Y., Zhang, Y., Yu, Z., Li, S., Ye, Z., Li, C., Wan, C., & Lin, Y. C. (2023). GPT4AiGChip: Towards next-generation AI accelerator design automation via large language models. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)* (pp. 1–9). IEEE.
- Kiesler, N., Lohr, D., & Keuning, H. (2023). Exploring the potential of large language models to generate formative programming feedback. In *2023 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE.
- Li, J., Li, G., Tao, C., Zhang, H., Liu, F., & Jin, Z. (2023). Large language model-aware in-context learning for code generation. *arXiv preprint*, arXiv:2310.09748.
- Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., et al. (2023). Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 100017.
- Lu, J., Yu, L., Li, X., Yang, L., & Zuo, C. (2023). LLaMA-Reviewer: Advancing code review automation with large language models through parameter-efficient fine-tuning. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 647–658). IEEE.
- Makridakis, S., Petropoulos, F., & Kang, Y. (2023). Large language models: Their success and impact. *Forecasting*, 5(3), 536–549.
- Pan, J. Z., Razniewski, S., Kalo, J.-C., Singhania, S., Chen, J., Dietze, S., Jabeen, H., et al. (2023). Large language models and knowledge graphs: Opportunities and challenges. *arXiv preprint*, arXiv:2308.06374.
- Sallou, J., Durieux, T., & Panichella, A. (2023). Breaking the silence: The threats of using LLMs in software engineering. *arXiv preprint*, arXiv:2312.08055.
- Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1* (pp. 27–43).
- Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., & Garg, A. (2023). ProgPrompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 11523–11530). IEEE.
- Thakur, S., Ahmad, B., Pearce, H., Tan, B., Dolan-Gavitt, B., Karri, R., & Garg, S. (2023). VeriGen: A large language model for Verilog code generation. *arXiv preprint*, arXiv:2308.00708.
- Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022a). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts* (pp. 1–7).
- Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022b). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts* (pp. 1–7).
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., et al. (2023). A survey on large language model-based autonomous agents. *arXiv preprint*, arXiv:2308.11432.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., et al. (2022a). Emergent abilities of large language models. *arXiv preprint*, arXiv:2206.07682.

- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., et al. (2022b). Emergent abilities of large language models. *arXiv preprint*, arXiv:2206.07682.
- Wu, T., Jiang, E., Donsbach, A., Gray, J., Molina, A., Terry, M., & Cai, C. J. (2022). PromptChainer: Chaining large language model prompts through visual programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts* (pp. 1–10).
- Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023a). SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning* (pp. 38087–38099). PMLR.
- Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023b). SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning* (pp. 38087–38099). PMLR.
- Xu, F. F., Alon, U., Neubig, G., & Hellendoorn, V. J. (2022). A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming* (pp. 1–10).
- Zhang, Z., Zhang, X., Xie, W., & Lu, Y. (2023). Responsible task automation: Empowering large language models as responsible task automators. *arXiv preprint*, arXiv:2306.01242.
- Zheng, Z., Ning, K., Chen, J., Wang, Y., Chen, W., Guo, L., & Wang, W. (2023). Towards an understanding of large language models in software engineering tasks. *arXiv preprint*, arXiv:2308.11396.