

# Contents

## Contents 1

### 1.0 Introduction 1

### 2.0 Literature Review 2

#### 2.1 Rise of the Large Language Model (LLM) 3

### 3.0 Methods 4

### 4.0 Results 5

### 5.0 Summary 6

## References 7

## 1.0 Introduction

This page is intentionally blank.

## 2.0 Literature Review

The following literature review presents the necessary background to better understand Large Language Models (LLMs) and their impacts on how novice programmers learn to code. We begin by exploring key research around LLMs and how they have impacted disciplines since their inception. Specifically, we provide evidence of LLMs as a disruptive innovation in the field of computer programming. LLMs have transformed not only how people code, but the ways in which programmers think about programming itself. This has implications for not only how code will be written in the future, but what skills are necessary for becoming a programmer and the methods by which those skills are taught to novices.

In addition, this literature review will explore the research of LLM use by teachers, students and administration within higher education. The research embodying the strengths, drawbacks, and unique challenges of LLMs within this context is discussed. This is necessary to differentiate between the overarching issues of LLMs in education and those specific to the topic of computer programming education.

Research will show LLMs are transforming how skilled professionals write code. LLMs are being used predominantly by teachers and students in computer programming education as well, which has forced academics to rethink not only how to teach computer programming to novices, but what should be taught in the first place. The literature will identify this through the positive and negative impacts LLMs are having programming education.

The academic literature on computer science education discusses the well-known challenges of novice users learning to program. These challenges are the impacts of cognitive load on learning, getting learners to focus on computational literacy over syntax and technology, the role of self-efficacy and motivation on success, and the importance of metacognition for building problem-solving skills. This literature review will revisit each of these challenges through the lens of how LLMs are impacting them both positively and negatively.

## **2.1 Rise of the Large Language Model (LLM)**

Natural Language Processing (NLP) has undergone significant changes in the past few years. With the introduction of transformer-based neural network architecture, Natural Language Processing took a giant step forward in performance and accuracy (Gillioz et al., 2020).

The Generative Pre-Trained Transformer (GPT) reasonably predicts the next word in a sequence using the input and previously generated output. While predicting the next token in the sequence is the extent of their ability (Shanahan, 2024), transformer-based language models trained on large data have demonstrated highly effective reasoning capabilities.

Open AI's foundational paper, "Language Models are Few-Shot Learners", demonstrated these transformer-based language models can produce human-level performance when they are trained on large corpora (Brown et al., 2020). This was a pivotal discovery because at the time since prior to this paper transformer-based models were trained to be relatively task specific (Zhao et al., 2023).

The paper from Brown et al., 2020 led to significant advancements in research with respect to understanding the capabilities of LLMs. Wei et al. (2022) discovered reasoning capabilities of LLM's can be improved through a technique called chain-of-thought prompting. By including few-shot examples that break down complex reasoning into steps, the LLM can use those shots provided as an example of how to explain a complex process.

Halevy et al.'s (2009) seminal paper, "The unreasonable effectiveness of data", explains as the training data set size increases, so does the model accuracy. In addition, the specific selected model algorithm becomes less relevant as training data set size increases. (Wei, Bosma, et al., 2022) documented a similar effect with large language models. Larger-sized models exhibited emergent abilities not found in their smaller counterparts. Examples of emergent abilities seen in the larger models include complex arithmetic and reading comprehension.

## **2.2 LLMs as a Disruptive Innovation for software development**

Christensen et al., (2018) divide technological innovations into two distinct types. Sustaining innovations improve existing products and services, while disruptive innovations provide a unique set of features to an initial set of customers. From the perspective of companies that offer generative AI such as Google,

Anthropic, Open AI, and Microsoft, Horn considers generative AI to be a sustaining innovation (Horn, 2024). In their comprehensive literature review of AI as a disruptive innovation, Păvăloaia & Necula, (2023) consider the application of Generative AI to be a disruptive innovation across different sectors such as healthcare, agriculture business and education.

The rise of AI-assistant programming tools in industry is evidence of this disruption. There are a growing set of tools available in the cloud: Github Copilot, Amazon CodeWhisperer, Gemini Code assist, Claude Code, Open AI Codex v2, Tabnine, and Codeium, in addition to self-hosted options like FauxPilot, Tabby, and CodeLLama. While each provides a unique set of features for differentiation, they all have primary functions like code completion, code generation, code explanations, and discussion. The primary value-add touted by these tools is developers will be able to write code in less time, improving productivity. Talk about “vibe coding” and Lovable, Bolt, Replit and Cursor.

### 3.0 Methods

TODO

### 4.0 Results

TODO

### 5.0 Summary

This page is intentionally blank.

### References

- 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. (2020). *Language Models are Few-Shot Learners*.
- Christensen, C. M., McDonald, R., Altman, E. J., & Palmer, J. E. (2018). Disruptive Innovation: An Intellectual History and Directions for Future Research. *Journal of Management Studies*, 55(7), 1043–1078. <https://doi.org/10.1111/joms.12349>
- Gillioz, A., Casas, J., Mugellini, E., & Khaled, O. A. (2020). *Overview of the Transformer-based Models for NLP Tasks*. 179–183. <https://doi.org/10.15439/2020F20>

- Halevy, A., Norvig, P., & Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2), 8–12. <https://doi.org/10.1109/MIS.2009.36>
- Horn, M. B. (2024, June 3). *What does Disruptive Innovation Theory have to say about AI?* - Christensen Institute. <https://www.christenseninstitute.org/blog/what-does-disruptive-innovation-say-about-ai/>
- Păvăloaia, V.-D., & Necula, S.-C. (2023). Artificial Intelligence as a Disruptive Technology—A Systematic Literature Review. *Electronics*, 12(5), 1102. <https://doi.org/10.3390/electronics12051102>
- Shanahan, M. (2024). Talking about Large Language Models. *Communications of the ACM*, 67(2), 68–79. <https://doi.org/10.1145/3624724>
- 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* (No. arXiv:2109.01652). arXiv. <http://arxiv.org/abs/2109.01652>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*.
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., . . . Wen, J.-R. (2023). *A Survey of Large Language Models* (No. arXiv:2303.18223). arXiv. <http://arxiv.org/abs/2303.18223>