AI Applied to Coding

Alexia Jennings, Jenan Qasem, Sanskriti Gyawali, Diwashna Poudel, Arushi Chaturvedi

March 30, 2025

Table of contents

Introduction

Artificial Intelligence, also known as AI, is starting to change the way people code. The purpose of AI coding tools is to accelerate the software development process, simplify the process, and create efficiencies when coding software. Tools like GitHub Copilot, OpenAI Codex, and TabNine enable programmers to autocomplete a line of code, recommend a solution, or even write a function based on a short description. These coding tools are particularly useful during repetitive tasks and even more useful when a programmer is attempting to decide how to write something. AI can also benefit new programmers by explaining code and helping with the syntax or structure of programming. The first fundamental principle behind how these tools work is machine learning. Machine learning is the process of training the AI on large amounts of code from various sources, like GitHub. The pattern, style, and conventions the AI is able to learn will allow it to intelligently suggest things based on how the programmer is coding.

Natural Language

The second major principle is natural language, which is the ability for the AI to learn the human language and translate that into code. So essentially, if a developer types something like, “create a login form,” the AI is able to return the actual HTML and JavaScript to do it. There are also some other types of coding AI supports. Some of the coding AI supports involve reinforcement learning, where, as the name suggests, the AI improves over time by learning from user feedback, so the more you use the tool, the more it is able to contribute suggestions. AI can be found in code review tools to automatically find bugs or security issues, and help remove messy code. While these AI tools still leave much to be desired and require human oversight, they are already becoming a large part of development practice today, and will only take over further in the future. Artificial Intelligence (AI) is already changing the field of software development because it handles mundane tasks, helps improve code quality, and even automatically generates complete programs for you. In just the past several years, we have seen developments in AI bring in useful tools and techniques that assist developers through the whole software development process, from code writing to testing and debugging.

Large Language Models

One of the most important advances is the use of large language models (LLMs) such as OpenAI's Codex and Meta's Code Llama for code generation. Trained on large volumes of public code and natural language, tools such as GitHub Copilot and Amazon CodeWhisperer are able to make suggested completions to code inside the editor, suggesting functions or even files in real time. This can enhance productivity, but also help developers discover and work with languages they are not familiar with or with frameworks they have not used before.

Code Review, Documentation, & Summarization

AI is also improving bug detection and code review. Machine learning algorithms now analyze codebases and pinpoint common bugs, vulnerabilities, and inefficiencies. Tools like DeepCode, Snyk, and CodeGuru, for example, are tools that will continually learn from open-source codebases and generate intelligent, rapid feedback. This reduced the manual review needed for code and contributed to overall software reliability. Translating natural language to code is now a new area, where developers describe in words in plain English what they need, and the AI translates that into code. This lowers the barrier to software development for non-developers, as well as improves the agility of professionals who can now concisely develop prototypes. AI technology can also be harnessed to automatically generate unit testing suites to lessen the workload for quality checks. Products such as Diffblue Cover and Ponicode, for example, automatically generate comprehensive testing suites to not only augment testing coverage but also catch bugs earlier in the development process. Finally, AI-assisted code documentation and summarization tools are helping teams to keep readable, documented codebases. These tools can run through functions and create documentation or summaries, which support new hire onboarding and collaborations. Today, there is a trend towards collaborative AI, where AI is more of a coding partner, rather than a tool. Research has shown autonomous AI agents can plan, code and debug with little human interaction through projects such as SWE-Agent. In addition, open-source AI generators such as StarCoder are becoming popular due to their open and editable nature.

Conclusion

AI is changing the creation of software. Gone are the days of needing extensive manpower and time. Now, with software development assistants such as GitHub Copilot and OpenAI Codex providing direct assistance, there is the opportunity for streamlined generation, debugging, and testing of code. With advancements in AI, its function is bound to change from assistant to co-creator in the hands of qualified experts and novices alike, enabling the construction of smarter and more reliable software systems. This transformation is powered by a combination of human intelligence and AI efficiency.

References

Alarcón-Zendejas, A. P., Scavuzzo, A., Jiménez-Ríos, M. A., Álvarez-Gómez, R. M., Montiel-Manríquez, R., Castro-Hernández, C., Jiménez-Dávila, M. A., Pérez-Montiel, D., González-Barrios, R., Jiménez-Trejo, F., Arriaga-Canon, C., & Herrera, L. A. (2022). The promising role of new molecular biomarkers in prostate cancer: From coding and non-coding genes to artificial intelligence approaches. Prostate Cancer and Prostatic Diseases, 25(3), 431–443. https://doi.org/10.1038/s41391-022-00537-2

Amodei, D., Ananthanarayanan, S., Anubhai, R., et al. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. Proceedings of the 33rd International Conference on Machine Learning, 48(1), 173-182. https://dl.acm.org/doi/10.5555/3045390.3045410

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https://doi.org/10.1145/3442188.3445922

Hendler, J. (2023). Understanding the limits of AI coding. Science (American Association for the Advancement of Science), 379(6632), 548–548. https://doi.org/10.1126/science.adg4246

Kakhiani, D. (2024). Code at the speed of thought: Exploring the impact of AI on coding practices. ResearchGate. https://www.researchgate.net/publication/379994921_Code_at_the_Speed_of_Thought_Exploring_the_Impact_of_AI_on_Coding_Practices

Merow, C., Serra-Diaz, J. M., Enquist, B. J., & Wilson, A. M. (2023). AI chatbots can boost scientific coding. Nature Ecology & Evolution, 7(1), 960–962. https://doi.org/10.1038/s41559-023-02063-3

Tongshuang, W., Terry, M., & Jun Cai, C. (2022). AI chains: Transparent and controllable Human-AI interaction by chaining large language model prompts. Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 385(1), 1–22. https://doi.org/10.1145/3491102.3517582