

# 인공지능과 프로그래밍의 미래

백인규

2023.11.22

## Abstract

현재 대중에 공개된 인공지능(AI) 기반 프로그래밍 도구에는 명확한 한계가 존재한다. 하지만 인공지능 기반 도구들은 가지고 있는 장점 또한 많으며, 장점을 극대화하도록 도구를 이용한다면 단점을 상쇄할 수 있다. 인공지능 기반 도구들의 장점은 크게 두 가지가 있다. 첫째로, 프로그래머의 사고와 결합하여 사용한다면 사람이 만들어내는 실수를 줄일 수 있다. 단순하거나 패턴화가 이루어진 작업은 인공지능에게 맡겨 오류를 줄일 수 있기 때문이다. 둘째로, 상식적인 시각 밖에서의 프로그램 합성이라는 창의적인 측면을 제공해 줄 수 있다. 막대한 양의 데이터를 기반으로 하는 인공지능은 복잡한 문제에 대해 새로운 아이디어를 제시할 수 있기 때문이다. 따라서 인공지능 기반 프로그래밍의 적절한 도움이 이루어진다면 프로그래머는 고차원적인 사고를 하는 데에 더욱 많은 자원을 투자할 수 있게 될 것이며, 이것이 프로그래밍이 발전해 나가야 할 미래이다.

현재 대중에 공개된 인공지능(AI) 기반 프로그래밍 도구에는 명확한 한계가 존재한다. 최근 인공지능 기술의 발전으로 인해 많은 인공지능 기반 프로그래밍 자동화 도구들이 공개되고 있다. 특히 Copilot과 Codex와 같은 코드 완성 도구들은 거의 모든 자연어 명령에 대해 대응되는 코드를 생성해 준다. 하지만 인공지능 기반 도구들이 생성하는 방식은 정확성과는 거리가 멀다. 이러한 도구들은 사용자의 세부 요구사항을 만족하는 프로그램 합성에 집중하는 것보다 사용자의 요구사항의 방향성과 최대한 유사하도록 프로그램을 만들어 주는 것에 중점을 둔다. 그 결과는 특정한 경우에 대한 정확도의 부재로 나타나는데, Pearce(2022)에 의하면 특정한 취약점이 발생할 수 있는 상황에 대한 코드를 Copilot으로 작성한 결과 약 40퍼센트가 취약한(vulnerable) 것으로 나타났다 [1].

하지만 인공지능 기반 프로그래밍은 가지고 있는 장점 또한 많다. 아래에서 서술할 인공지능 기반 프로그래밍의 장점은, 프로그래머가 적재적소에 사용한다면 단점을 상쇄시키면서 생산성까지 보장할 수 있다는 것이다. 일반적으로, 어떠한 도구에 단점이 존재한다고 해서 그 도구를 쓰지 못할 이유는 없다. 도구의 장점을 부각시킬 수 있는 상황에서만 도구를 사용하면 효율을 극대화할 수 있기 때문이다.

첫째로, 프로그래머의 사고와 결합하여 사용한다면 사람이 만들어내는 실수를 줄일 수 있다. 인공지능은 사람의 뇌와 비교할 수 없는 크기의 연산을 통해 패턴인식과 추론을 제공한다. 이는 프로그래머의 결정을 보완할 수 있다. 하나의 예를 들자면, 프로그래머의 입장에서 스스로가 만들어낸 프로그램이 오류를 범할 가능성이 있는지 검증하려면 많은 반복과 재검사를 통해 반례를 찾아내야 하지만, 잘 구성된 인공지능 도구의 경우는 프로그래머에게 즉시 반례를 제공할 수 있다. 또한 단순한 오타 혹은 논리적인 오류 또한 미연에 방지할 수 있다. 이처럼 인공지능 기반 도구를 사용한다면 프로그래머가 저단계의 오류를 찾기 위해 낭비하는 자원을 아낄 수 있으며 결론적으로 실수를 줄일 수 있다.

둘째로, 상식적인 시각 밖에서의 프로그램 합성이라는 창의적인 측면을 제공해 줄 수 있다. 인공지능 기반 프로그래밍 도구는 인간이 원하는 프로그램을 합성하면서도, 인간이 빠질 수 있는 논리의 고착화, 혹은 사고의 일률성에서 더욱 자유로울 수 있다. 이는 인공지능이 막대한 양의 데이터를 기반으로 하기 때문에 가능하다. 복잡한 문제에 대해 새로운 아이디어를 제시할 가능성이 있는 인공지능 기반 도구들은 프로그래밍의 접근 방식을 혁신적으로 바꿀 수 있으며, 더 나은 해법을 찾는 데 창의적인 지혜를 줄 수 있다.

따라서 인공지능 기반 프로그래밍의 적절한 도움이 이루어진다면 프로그래머는 고차원적인 사고를 하는 데에 더욱 많은 자원을 투자할 수 있게 될 것이다. 단순하거나 패턴화가 이루어진 작업은 인공지능에게 전적으로 맡기고, 고차원적인 문제에 대해서는 인공지능 기반 프로그래밍 도구의 창의적인 의견을 참고할 수 있다면 똑같은 인적 자원으로 더욱 높은 생산성을 보장할 수 있을 것이다. 이것이 프로그래밍이 발전해 나가야 할 미래이다.

## References

- [1] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. Asleep at the keyboard? assessing the security of github copilot’s code contributions. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 754–768. IEEE, 2022.