

THE PROBLEM OF USING ARTIFICIAL INTELLIGENCE IN IT INTERVIEWS

Pleskatsevich V.A. Panteleev N.A

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Perevyshko A.I. – Senior Lecturer at the Department of Foreign Languages

Annotation. This paper examines the problem in the selection of IT staff caused by the use of artificial intelligence by candidates during online interviews and when performing test tasks. The article analyzes a situation when a person looks like an expert at the hiring stage, but without the help of neural networks he cannot cope with basic tasks. The article suggests practical ways to change the selection process that will help companies test the real knowledge of specialists.

Keywords: IT industry, artificial intelligence, job interviews, AI, fundamental skills, IT recruiter

Introduction. The IT job market is facing a completely new challenge. The software development industry is undergoing massive changes due to the introduction of artificial intelligence. The widespread use of tools like ChatGPT and GitHub Copilot has significantly accelerated the execution of everyday tasks. For example, a study conducted by Microsoft and GitHub experts found that programmers who use AI assistants complete tasks 55.8% faster on average than their counterparts who rely on traditional methods. It would seem that such an increase in productivity should bring a lot of benefits to the industry. However, in practice, this technological breakthrough has caused a deep management crisis in the hiring process. While a few years ago, interviewers were trying to figure out what a candidate knew, now their main task is to determine whether the skills they claim to have are real. As artificial intelligence becomes more widely available, there is an alarming trend.

Main part. Candidates most often use artificial intelligence in interviews in two ways. Some secretly peek at neural network prompts on the second monitor right during an online interview. Others simply instruct the algorithms to fully generate the code for the test tasks. As a result, the company hires a new employee based on his ideal knowledge, which does not correspond to reality. This situation hides the gap between what the candidate has shown and his skills in practice. When a novice developer comes to his first internship at an IT company, he gets tasks that cannot be solved by simple copying. Real-world development always requires immersion in the specific context of the project.

To understand the essence of the problem, we need to figure out how beginners cheat the system using neural networks. The traditional methods of candidate screening are now «broken» at all stages.

The biggest impact was on homework assignments. Historically, this format has been considered the gold standard for evaluating the Junior and Middle levels of professionals. Companies would give a large assignment to see how a person wrote code. However, this no longer works. Neural networks like ChatGPT can write perfect code in just 10 minutes. As a result, someone with no experience brings in a completed project that was actually created by AI.

Even video calls no longer guarantee honesty. There are programs that listen to an interviewer's question, send it to a neural network, and display the answer on a nearby screen. A company hires a person because they are confident in their expertise, but the gap between generated code and real engineering thinking becomes apparent in the first few days of the internship.

Let's take database design, for example. When completing a test assignment at home, a candidate can use AI to write an impeccable SQL query full of complex views and triggers. The employer looks at the code and believes they have found an excellent specialist. However, when asked to design a simple data structure from scratch in the office, the same trainee becomes lost. They struggle to properly normalize tables and achieve at least the third normal form (3NF), and become confused about simple join operations (JOIN). As a result, the intern is unable to explain the logic behind the code that he supposedly wrote a week ago.

Exactly the same pattern can be observed in programming. The person sends the employer a beautiful structured code. Then a non-standard error occurs in the production environment. Neural

networks often cannot find such specific bugs. And since the developer doesn't really understand the algorithms of his own code, he can't fix it manually.

Table 1 clearly shows what companies expect and what they actually get when hiring candidates who abuse the prompts of algorithms.

Table 1 – Comparison of demonstrated and actual competencies of candidates

Tested skill	Demonstration using AI (selection stage)	Reality without AI (probably period)
Code writing	Impeccable syntax, modern design patterns	Complete misunderstanding of the logic of the algorithm; inability to find a bug without a development environment
Database	Generated complex SQL queries	Misunderstanding of the logic of relationships between tables; inability to design a basic schema
Problem solving	Instant, optimized solutions to test tasks	Work stoppage when faced with a bug; inability to find solutions on their own

The cost of this problem is huge. Companies spend months searching for a candidate, paying salaries, and wasting the time of experienced employees who act as mentors. All of this becomes meaningless when a new employee eventually has to be fired.

The psychological side of this problem is just as alarming as the technical side, as excessive reliance on neural networks during the learning process sets up a time bomb for the future career. Researchers from the ACM Association warn that reliance on artificial intelligence critically reduces students' ability to solve problems independently. This is supported by the work of Yilmaz & Yilmaz, who found that blindly copying AI code deprives programmers of self-confidence, as they do not gain personal experience in overcoming errors. As a result, he can't explain the logic behind the code he submitted for review, and any follow-up question from an experienced colleague causes him to panic.

Solving this problem requires a complete overhaul of how the industry conducts interviews. Companies must accept the fact that coding is no longer an exclusive human skill. Artificial intelligence is now doing it. Therefore, the process of evaluating candidates must shift towards testing their architectural and algorithmic thinking.

First, companies need to completely abandon test homework assignments. Their results no longer reflect a person's real abilities. Second, classic algorithm writing tasks should be replaced with the Code Review format. A candidate is given a ready-made but obviously vulnerable or confusing piece of code generated by a neural network. The engineer's task is to find errors, explain them, and suggest solutions. This directly tests fundamental skills and the ability to read someone else's code.

Conclusion. This article shows how neural networks are disrupting the traditional hiring process. The rapid development of large language models has irreversibly transformed the IT market, breaking down the traditional, time-tested mechanisms of hiring specialists. The uncontrolled use of neural networks by candidates without work experience creates a dangerous illusion of qualifications. In order to maintain the high quality of the selection and minimize financial losses, companies should urgently modernize the formats of technical interviews. In the era of victorious artificial intelligence, the focus of recruiting should shift from evaluating written code to analyzing fundamental knowledge. Only by testing systems thinking and the ability to solve non-standard engineering problems will it be possible to identify specialists who will manage neural networks, and not just copy their answers.

References

1. Peng, S. *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot* / S. Peng, E. Kalliamvakou, P. Cihon, M. Demirer // *arXiv preprint arXiv:2302.06590*. – 2023. – P. 1–18.
2. *The economic potential of generative AI: The next productivity frontier* / McKinsey & Company // *McKinsey Global Institute Report*. – 2023. – P. 15–28.
3. Becker, B. A. *Programming Is Hard-Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation* / B. A. Becker [et al.] // *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*. – 2023. – Vol. 1. – P. 500–506.
4. Yilmaz, R. *The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self efficacy and motivation* / R. Yilmaz, F. G. K. Yilmaz // *Computers and Education: Artificial Intelligence*. – 2023. – Vol. 4, № 100147. – P. 1–12.