

Como as interações entre Inteligência Artificial e seres humanos estão impactando a programação?

Carlos Kauã Berghahn, Thyago Augusto Martins Silveira, Orientador: Giuliano Araujo Bertoti

FATEC Professor Jessen Vidal

carlos.berghahn@fatec.sp.gov.br, giuliano.bertoti@fatec.sp.gov.br

1. Introdução

A programação é uma disciplina em constante evolução e, nos últimos anos, a Inteligência Artificial (IA) emergiu como uma força transformadora nesse campo. A interação entre IA e seres humanos na programação tem o potencial de revolucionar a forma como desenvolvemos software, automatizamos tarefas e resolvemos problemas complexos. Neste artigo, exploraremos as diversas formas pelas quais a IA e os seres humanos podem se beneficiar mutuamente, impulsionando o progresso e abrindo novas possibilidades para o futuro da programação.

2. Automação de tarefas e códigos Autopilots

A IA tem o poder de automatizar tarefas repetitivas e mundanas, permitindo que os programadores se concentrem em atividades mais criativas e desafiadoras. Por exemplo, ferramentas de geração automática de código podem produzir trechos de código com base em requisitos específicos, acelerando o processo de desenvolvimento.

Outro exemplo são os chamados códigos Autopilot – termo utilizado para códigos que por si só já fazem todas as tarefas instruídas, sem a necessidade de um ajuda externa (uma pessoa). Um exemplo de autopilot seria o sistema de alguns aviões, onde estes possuem a função de Piloto Automático, que permite manter o voo e sua trajetória sem ter necessariamente uma pessoa no comando. Há diversos casos onde existem códigos autopilots, abrangendo várias áreas, como na área da saúde e na medicina, em processos industriais, transportes autônomos, entre outros.

3. A IA no processo de depuração e testes

Outra área em que a IA está desempenhando um papel importante é no processo de depuração e testes de software. Através da análise de grandes conjuntos de dados e do reconhecimento de padrões, a IA pode identificar possíveis erros e falhas no código. Isso ajuda os programadores a identificar e corrigir problemas mais rapidamente, melhorando a qualidade do software.

Além disso, a IA pode ser usada para automatizar tarefas de teste, permitindo que os programadores se concentrem em aspectos mais complexos do desenvolvimento. Por exemplo, algoritmos de IA podem ser treinados para realizar testes de regressão e identificar possíveis problemas de compatibilidade em diferentes ambientes. Isso aumenta a eficiência e a confiabilidade do processo de teste, proporcionando uma base sólida para o lançamento de software de alta qualidade.

4. Colaboração homem-máquina

A colaboração entre humanos e máquinas é uma área de pesquisa cada vez mais importante. Os programadores podem se beneficiar da assistência da IA durante o processo de desenvolvimento, com sugestões de código, verificação de erros e até mesmo sugestões criativas para melhorar a solução de um problema. Essa colaboração pode ampliar a capacidade dos programadores, permitindo que eles explorem soluções mais complexas e inovadoras.

5. Desafios e ética

Apesar de todas as promessas, a interação entre IA e seres humanos na programação também levanta questões éticas e desafios a serem enfrentados. É necessário considerar a privacidade e segurança dos dados, a transparência dos algoritmos de IA e o impacto potencial no mercado de trabalho.

6. Conclusões

A interação entre a Inteligência Artificial e os seres humanos na programação oferece um futuro empolgante e promissor. Ao automatizar tarefas repetitivas, otimizar o desenvolvimento de software, criar interfaces intuitivas e promover a colaboração entre humanos e máquinas, a IA pode impulsionar a eficiência, a criatividade e a inovação no campo da programação. No entanto, é importante abordar os desafios éticos e garantir que o progresso tecnológico seja orientado para o benefício da sociedade como um todo. Com uma abordagem cuidadosa e responsável, a interação entre IA e seres humanos na programação pode moldar um futuro empolgante e transformador para a indústria do software.

7. Referências

- [1] UFSCAR. Repositório Institucional da UFSCAR: Um processo dirigido a modelos para geração de código <https://repositorio.ufscar.br/bitstream/handle/ufscar/528/5180.pdf?sequence=1>
- [2] SHIN, J.; NAM, J. A Survey of Automatic Code Generation from Natural Language. [s.d.]
- [3] KEMPF, D. et al. Automatic Code Generation for High-Performance Discontinuous Galerkin Methods on Modern Architectures. Disponível em: <https://arxiv.org/abs/1812.08075>
- [4] BALOG, M. et al. DeepCoder: Learning to Write Programs. arXiv:1611.01989 [cs], 8 mar. 2017.