Programming in the Age of Intelligent Machines

Breandan Considine

August 9, 2021

1 Introduction

Since the invention of modern computers in the mid 20th century, computer programming has undergone a number of paradigm shifts. From the rise of functional programming, to its popular adoption in languages like LISP, ML, and Kotlin, to the emergence of myriad tools and frameworks – its practitioners have witnessed a veritable Renaissance in the art of computer programming. With each of these paradigm shifts, programmers have realized new avenues for reasoning and expressing their ideas more clearly and concisely.

Over the last few years, another paradigm shift has arrived, with significant implications for how we think about and write programs in the next century. By most historical measures, computers have grown steadily more intelligent and capable of assisting programmers with common tasks. For example, intelligent programming tools (IPTs) powered by neural networks have this year helped over 10 million human beings program computers on a monthly basis. As IPTs help digitally illiterate communities to discover their innate aptitude for computer programming, this population will continue to expand.

The art of computer programming is a uniquely creative exercise among the range of human activities. It engages our innate linguistic, logical, imaginative, social and sensorimotor abilities to summon abstract ideas into reality, and ultimately, gives human beings the freedom to design their own realities. In collaboration with fellow humans and the increasing participation of IPTs, vast and intricate virtual worlds have been developed, where the majority of humankind now chooses to spend their lives. As these worlds offer increasingly compelling opportunities, their population too will continue to grow.

Arguably IPTs now share an equal role in shaping many aspects of computer programming, from knowledge discovery to API design, to program synthesis and automatic testing. However, this balance is slowly shifting. Once the creators, programmers are increasingly consumers of information provided by the IPTs, which exert a growing influence in our development process. With the unique creative opportunities their expanding powers of intellect afford, what role should we expect humans and machines to play in our creative partnership? This is the question we set out to understand in the following report.

2 Neural Language Models for Source Code

3 Knowledge Discovery and Neural-Guided Search

4 Computer-Aided Reasoning Tools

5 Automatic and Synthetic Programming

6 Future Directions of Computer Programming

7 Conclusion