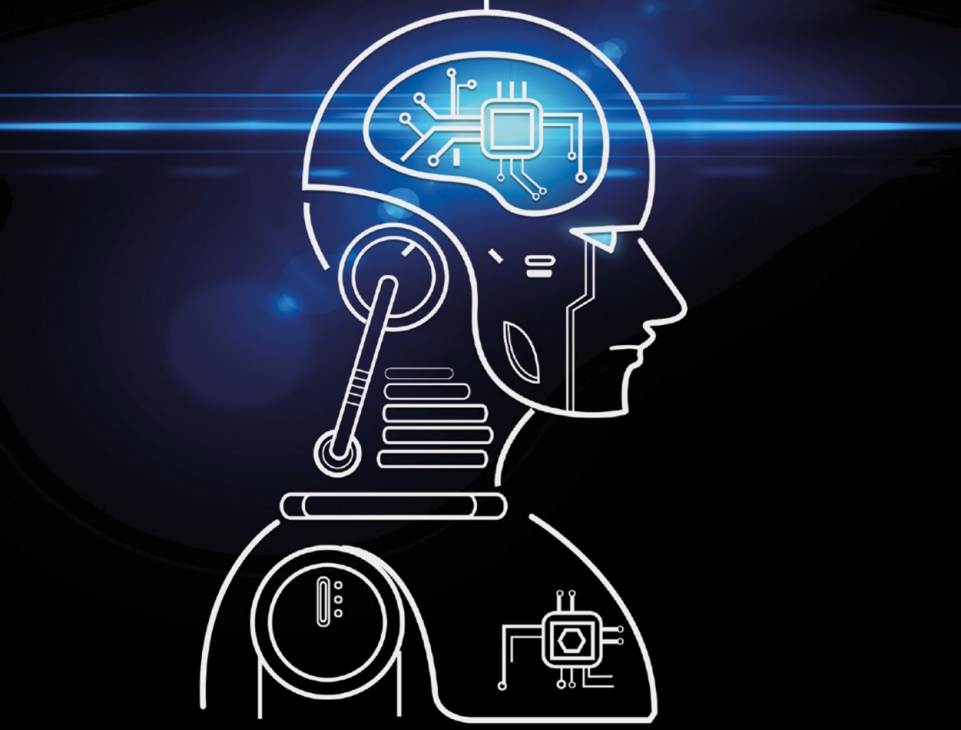
Penguin Readers

ARTIFICIAL INTELLIGENCE



Based on

THE ROAD TO CONSCIOUS MACHINES

by MICHAEL WOOLDRIDGE

Contents

CHAPTER ONE

Alan Turing and the birth of AI

You could identify many different points in history as the birth of **artificial intelligence** (**AI**), but this particular story begins at the same time as computing itself. It was in 1935, in King's College, University of Cambridge, with a brilliant young student called Alan Turing.

Today, he is about as famous as any mathematician could ever hope to be. But until the 1980s hardly anybody had heard of Alan Turing, or that he died aged just forty-one. It seems that he killed himself after he was forced to take medicine to "treat" **homosexuality**, which was then a crime. Even mathematicians and computer scientists would only have known about some of his work, as much of it was done in secret for the UK government during the Second World War, and was only made public in the 1970s.[1]

Nowadays, we all know a little about Turing thanks in part to a (very inaccurate) 2014 film called *The Imitation Game*, which tells the story of his important code-breaking work at Bletchley Park during the Second World War. But for experts researching AI and computer science, he is respected for first inventing the computer, and not long after that, for inventing the field of AI itself. Turing was extraordinary in many ways, but one of the most amazing things about him was that he invented computers

by accident. As a mathematics student at the University of Cambridge in the 1930s, Turing decided to try to answer one of the main mathematical problems of the day called the *Entscheidungsproblem*, a German word for a **decision problem**. These are mathematical questions that have a yes/no answer. In 1928, mathematician David Hilbert first asked if there are decision problems that cannot be answered by simply following a **recipe** of **precise** mathematical steps. So, can these problems be reduced to simply following recipes, or computer programs, without the need for real intelligence?

When Turing took on this challenge in 1935, he not only found a solution, but he did so very quickly. Turing realized that he needed to be able to make an exact plan of a recipe that can be followed **precisely**. To do this, he invented a mathematical problem-solving machine, although it was not actually a machine, but a mathematical idea. We call these Turing machines in his honour.

A Turing machine follows the recipe it was designed for, and you can write the code for any mathematical recipe because it is so **powerful**. If all mathematical decision problems can be solved by following a recipe, then for any decision problem, you should be able to design a Turing machine to solve it. So, to return to David Hilbert's question, all you have to do is to show that there are some decision problems that cannot be answered by any Turing machine; and that is what Turing did.

Next, Turing showed that his machines could be turned

into general-purpose problem-solving machines. He designed a Turing machine that would follow any recipe that you gave it. We now call these general-purpose Turing machines Universal Turing Machines. When a computer is reduced to its simplest form, it is just that: a Universal Turing Machine made real. The programs that a computer runs are just recipes.

The way in which Turing solved the *Entscheidungsproblem* using his new invention was not only very clever, but it also helped scientists to understand if AI would be possible one day. Turing's idea was that the machines could be programmed to answer questions about other Turing machines. He considered this decision problem: if you have a Turing machine with the right information put into it, will it stop and give you an answer in the end, or could it carry on working forever?

Turing realized that there could be no recipe for checking whether a Turing machine stops, so the question "Does a Turing machine stop?" is an **undecidable problem**. In this way Turing established that there are decision problems that cannot be solved by simply following a recipe, answering Hilbert's *Entscheidungsproblem*: mathematics cannot be reduced to following recipes. With this result, Turing not only achieved one of the greatest things in twentieth-century mathematics, but he also started us on the path towards computers.

Turing did not invent his machines with the idea of actually

building them, but it was not long before he and others thought about doing just that. In Munich, Germany, during the Second World War, Konrad Zuse designed a computer called the Z3 for the German government. It was not quite a modern computer, but had many of the key **ingredients** of one. Meanwhile in Pennsylvania in the United States of America, a team led by John Mauchly and J. Presper Eckert developed a machine called ENIAC to calculate information required for firing guns over long distances. And thanks to some changes made by John von Neumann, a brilliant Hungarian mathematician, ENIAC established the basic design for the modern computer, named the Von Neumann architecture in his honour.

After the Second World War, Fred Williams and Tom Kilburn built the Manchester Baby in England. This led directly to the Ferranti Mark 1, the first computer in the world that you could buy. Turing himself joined the team at Manchester University in 1948, and wrote some of the programs to run on it. By the 1950s, all the key ingredients of the modern computer had been developed, and there really were machines that could do what Turing had imagined with his mathematical idea. All you needed was enough money to buy one, and a building big enough to keep it in. These early computers were very big and used the same amount of electricity as you need for three modern homes. They have been getting smaller and cheaper ever since.

Although AI did not yet have a name, a lot of people were discussing the idea in the 1950s, and Alan Turing was one of

them. He wanted to stop people saying that "machines cannot think". To do this he suggested a test, which we now call the Turing test. The basic idea for this test was the Imitation Game, a Victorian game in which someone tried to tell if another person was a man or a woman from their answers to questions. Turing thought that you could use a similar test for AI.

The way the Turing test usually works is with a human interviewer typing questions and then deciding whether a person or a computer program is answering. Turing argued that if, after enough time had passed, the interviewer still could not be certain if it were a computer program or a person, then you should accept that if it were a program then it had some sort of human-**level** intelligence. The test is brilliant because Turing avoided all the questions about whether a program was "really" intelligent, or "really" thinking, by proving that it is doing something that makes it seem exactly the same as the "real thing". However, it is important to see only the interviewer's questions and the answers rather than what is happening inside the machine; which we are not allowed to examine. Turing described his test from the point of view of a modern **digital** computer, making this the first scientific paper on AI.[2]

Although the Turing test is simple, elegant and easy to understand, it has not always been put to the best use. Programmers have often tried to trick interviewers into believing that they are dealing with a real person, rather than trying to deal with the actual issues of intelligent behaviour. The most

famous example of this was a program called ELIZA, written by the German-American MIT (Massachusetts Institute of Technology) computer scientist Joseph Weizenbaum in the middle of the 1960s. ELIZA, which Weizenbaum never meant to be used in a Turing test, takes the part of a doctor asking open questions to a patient about their feelings. The program starts well, but after even a couple of sentences the conversation becomes meaningless, as ELIZA obviously does not understand what the person is saying.

It is not clear whether Turing imagined that anyone would ever actually try out his test for real, but every year programmers attempt to win $100,000 doing just that in the Loebner Prize Competition. Some critics feel that there is little point to this competition, as the programmers use tricks rather than trying to win by having a meaningful conversation that shows human-level, practical understanding. Today we also have internet chatbots, programs that attempt to talk to **users**, usually on social media and often using little more than conversations built on keywords, just like ELIZA did. Chatbots of this kind are not AI.