10

Artificial Intelligence

When I was 20 or so, I paid a visit to an older acquaintance who had just acquired a new boyfriend. At one point during a long walk, out of earshot from her partner, my acquaintance confided in me. 'He works in artificial intelligence', she said circumspectly. I had not heard this expression before, but her grave delivery left me in no doubt about the weight of his activities. To hear that this boyfriend had managed to secure gainful employment was reassuring... but artificial intelligence? I thought it wise not to inquire further.

This chapter will introduce the goals that researchers in artificial intelligence (AI) have *actually* set themselves, and how their efforts at reaching these goals have come along during AI's first half century.[1] In the process, we shall get acquainted with some of the ways in which AI has dealt with vagueness.

A brief history of AI

There does exist a kind of artificial intelligence (AI) that is as ambitious as the term suggests. In its most pure form, the ambition of AI is to build working models of the entire human mind, and sometimes the human body too. Mary Shelley's novel *Frankenstein*, published in 1818, comes to mind, in which a student creates a monster from parts of corpses. Even earlier, in the seventeenth century, the Jewish myth of the Golem involved a similar feat, in which a Rabbi Loew of Prague builds a manlike creature from sticky river mud, to help his community who are going through difficult times. Understandably, he makes the Golem a bit larger and stronger than ordinary people—something that became an enduring theme of the robotic genre, to be played out in numerous novels and films.

It took AI until the 1950s to enter mainstream science. The year that is usually cited is 1956. Computers were only beginning to come out of the research laboratories: reportedly, by 1953, IBM had sold no more than nineteen computers. Computer memory and processing power were extremely limited. Even the largest computers in existence could not store more than about 5 million bytes; it would be years before floppy disks and PCs were to appear on the scene. Most important of all, people's ability to construct useful computer programs was still in its infancy. Yet, computer scientists were starting to dream. A brainstorming event known as 'the Dartmouth conference' was organized around the following proposal:

> We propose that a (...) study of artificial intelligence be carried out (...). The study is to proceed on the basis of the conjecture that every aspect of learning or any feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.

Two years later, the researchers Simon and Newell predicted that significant advances in a number of major areas of AI would be achieved in just ten years. Among other things, they predicted that by that time computers would have learned to play chess better than any human. Fifty years later, computers have become much more 'intelligent' in many respects, including the areas on which Simon and Newell concentrated. At the time of writing, for example, the world has started to accept that some computer chess programs—answering to the names of 'Deep Blue' and 'Fritz'—are stronger than even the best human players. Simon and Newell did eventually have their way, though it happened thirty years later than they predicted.

With hindsight, it seems hard to understand the unbridled optimism of those early days. Even in areas where success has indeed been achieved, human abilities have proved difficult to simulate. Consider chess again. Human players, whose thinking has been much studied by psychologists, appear to work with highly general, 'strategic' rules, such as one sometimes attributed to the former Cuban world champion Capablanca: 'Before launching an attack, close the pawn formation' (this means positioning your pawns so snugly to those of your opponent that neither player's pawns can capture the others, thus bringing a degree of stability to the game). Maybe one day computers will use strategic rules of this kind, but at present they don't. For them, chess is all tactics and no strategy. Essentially, they make most of their decisions by means of brute force search: going through all possible moves, all possible responses to these moves, and so on. Sophisticated search strategies invented for chess—which are able to selectively disregard certain moves—are a very substantial contribution to computing, with applications in many areas. Clearly, computer chess has achieved a lot. Still, some researchers are disappointed, because *people* could never play chess using search alone: even grandmasters cannot think far enough ahead to make it work. Conversely, computers would have great difficulty making sense of Capablanca's rule. The reason is that this rule is merely a rule of thumb. Suppose, for example, you can checkmate your opponent; closing your pawn formation first would be a blunder in this case. A rule of thumb is a *vague* rule: a rule that applies only most of the time.

The same pattern can be observed in much of AI: successes are achieved, but not overnight, and in unexpected ways. The use of ordinary language, on which we shall focus shortly, is a case in point. The trend is perhaps most evident in *machine translation*, which aims to let computers *translate* text from one language to another. Machine translation could, of course, be tremendously useful: if successful, it would amount to the demolition of the Tower of Babel! For some time, machine translation was seen as a potential 'killer application' for AI, but in 1966 a report commissioned by the American government concluded that machine translation was never going to work: the argument involved examples such as the following.

Suppose a computer needs to translate sentences from English to German. Suppose it reads the word 'open'.[2] How should this word be translated? If this word is posted on the door of a store then it means the shop is not closed: a German would say that it is *offen*. But if it appears on a banner in front of the store it is likely to mean something else: the shop has just been opened for the first time. In German this happens to be expressed differently, by the phase *Neu eröffnet* (i.e. 'newly opened').

The example hinges not on vagueness, but on that other pain in the linguistic neck: ambiguity. The problem is that the English word 'open' can mean two different things, while German does not have one expression covering both of these. Consequently, the only way to arrive at a good German translation of 'open' without guessing is by assessing how the English word was intended. The author of the report, a machine translation researcher named Yehoshua Bar-Hillel, argued that it would be so difficult for a computer to do justice to all the factors that can possibly be relevant for this assessment that it is simply not feasible.

Bar-Hillel's report was a blow for machine translation, effectively halting its progress for a long time. Forty years later, machine translation has finally recovered. Machine translation systems are starting to approach the point where they become commercially attractive. (Don't ask them to translate a poem though!) By and large, it is corpus-based methods (discussed in Chapter 6) that are making the difference. Current machine translation systems work by applying statistical methods to huge *parallel* language corpora, in which the same information is expressed in different languages. Broadly speaking, these methods figure out in which contexts a word such as 'open' tends to be translated as *offen* and in which contexts as *neu eröffnet*. (The latter expression tends to co-occur frequently with such expressions as 'new', and 'location', whereas the other interpretation co-occurs more frequently with 'closed', for example.) Once again, an AI problem is starting to be cracked using techniques very different from the ones used by people.

After a lull in the 1980s, AI is once again thriving, though in a more modest spirit than before. Perhaps most importantly, researchers have come to realize that many things which *seem* easy are not. Playing chess may be challenging, but building a computer that can make conversation, recognize a person in a crowd, or play football, is proving to be even more challenging. It is, of course, rather humbling to realize that we share these 'problematic' abilities with little children. Perhaps we need to think differently about what it means to be human. If it is so difficult to let robots perform these 'non-intellectual' tasks, then perhaps we need to take them more seriously. It could consequently be argued that they should feature more strongly in our assessment of *human* intelligence (see Chapter 3).

Artificial intelligence?

Are computers intelligent? Arguably this question was thrown up in earnest for the first time by one of the founding fathers of computing, the Englishman Alan Turing. His work was so important and wide-ranging that, from where I stand, it is difficult to think of a more influential figure in the entire previous century.[3] Our understanding of the power and limitations of computers, for example, owes a tremendous amount to his work. His practical achievements are no less impressive: some historians believe that the Second World War would have ended differently without his contributions to code-breaking. In what follows, we focus on another, more speculative part of his work, involving what has come to be known as the Turing Test.

Thinking about intelligence, Turing was fascinated by a conversation game. The game is best played using a keyboard, so the participants communicate without hearing or seeing each other. In one version of the game, the role of a player we shall call the *deceiver* is to fool the other about his gender; the role of the player, whom we shall call the *detective*, is not to be fooled. The detective might, for example, start the game by asking what the deceiver is wearing. If the deceiver is a man, intent on making the detective believe he is a woman, he might lie and boast about his dress. (You haven't seen anything like it, an absolute bargain.) Being asked about the brand and cost of the dress, he should show decent knowledge of such things, or else the detective will see through his dastardly scheme.

For the deceiver to succeed, he has to think like a woman. Turing realized that the game could be turned on its head if the role of the deceiver is played by a computer rather than a person. The task for the detective in this case is no longer to determine the deceiver's gender—it doesn't even have any—but to find out whether the deceiver is a computer or a real person. The computer's task is to fool the detective into believing the latter. Analogous to the original game, the computer can win only by thinking like a human, to such an extent that, by observing the deceiver's contributions to the conversation, the detective cannot tell that the deceiver is a computer. Now suppose one carried out a test in which a computer was able to fool a majority of human subjects into believing it to be human. This would mean that people really cannot tell whether they are dealing with a person or a computer: we're just guessing. If this situation ever comes to pass, then surely the computer must count as human as far as its intellectual capacities are concerned, so Turing argued. If this ever happens the computer will be said to have *passed the Turing Test*, in which case one of the main aims of AI will have been achieved.

Turing's ideas about the Turing Test have sometimes been questioned, but the concept behind it is very much alive. In 1990, for example, the American philanthropist Hugh Loebner promised to award $100,000 for the first computer program to pass the Turing Test, and smaller yearly prizes are awarded to the program that comes closest to passing it. Additionally, a lot of other AI research focuses on small parts of Turing's challenge and can be seen as ultimately motivated by a desire to create machines that are intelligent in Turing's sense.

In the next chapter, we shall look in greater detail at AI researchers' attempts to get computers to pass the Turing Test. But first, let us review a few other areas where AI has had to deal with gradable phenomena. To start with, let us take a look at models of non-specialists thinking about physical processes.

Qualitative reasoning

The aim in *qualitative reasoning* is to construct computer models that focus on the essence of a process instead of its details. A model of this kind might say, for example, that the colour of your coffee will get lighter as you add more milk, without quantifying the effect. Let me explain how AI systems in this area work by looking at another example, involving a bathtub which is being filled with water.[4]

What happens if you open the tap and let water flow into the tub? Most of us would answer this question along the following lines: if the drain is closed then the level of the water will rise until it reaches the top. At this point the water will overflow and flood the bathroom. If the drain is open then the amount of water escaping via the drain will increase as the water level increases, and the tub will overflow only under certain conditions, relating to the amount of water flowing in, the size of the drain, and the shape of the tub. Physicists, of course, would describe the problem in much more quantitative terms. Most of us get by very well without so much detail, however, using a model that is merely *qualitative*.

Computer models of 'proper' physics are not difficult to construct, as long as the processes involved are not too complex. But if our aim is to model the knowledge that ordinary people bring to everyday tasks then a less detailed, qualitative model would be interesting to have. Such a model could be practically useful too, for example for educational purposes. A qualitative model of the human heart, for example, could help to explain the function of the heart to a medical student, elucidating the basic principles that are at work while suppressing unimportant details. Or suppose you were an inventor, playing around with innovative new heating systems: at the early stages of your work, you might benefit from a qualitative model that shows you which ideas might work and which ones won't. Details can obscure the things that matter.

What might a qualitative model of a bathtub look like? The basic behaviour of one famous program goes as follows, focusing on the situation in which the drain is open. The laws governing the system are simplified to the following regularities, where *Amount* is the amount of water in the bathtub, *Level* the level of the water in the tub, *Inflow* the amount of water flowing from the tap, and *Outflow* the amount of water escaping through the drain. *Netflow* is the net gain resulting from Inflow and Outflow.

1. Level goes up if and only if Amount goes up.
2. Outflow goes up if and only if Level goes up.
3. Outflow + Netflow = Inflow.
4. Netflow gets added to Amount.

The second of these regularities, for example, says that the amount of water escaping through the drain increases as the water level goes up, but *without* saying how one increase depends on the other. The program starts by activating a set of initial conditions which hold that, before the tap is opened, Amount = Level = 0. In this situation, the third law allows the system to infer from 0 + Netflow = Inflow that, at this moment, Netflow = Inflow. Since the tap is running, the fourth law implies that Amount must be going up. Now the first two laws kick in again to tell us that Level and Outflow go up as well. Since Inflow is constant, the third law implies that Netflow must be going down. The most interesting part of the model is reached when another law kicks in which says that several things can happen to the Level: either it may never reach the top of the bathtub, staying steady at some lower level because an equilibrium is reached, or it may reach the top. In the latter case, the water might either be overflowing, in which case we enter a different system of regularities, or just happen to remain steady at the top—an unlikely equilibrium, but not impossible.

Qualitative reasoning systems capture regularities while suppressing details. In ordinary language, the same function is often performed by degree adjectives:

> If a river flows fast mud and sand is carried out to sea. If a river flows slowly the mud sinks to the bottom—and makes the river shallow. The Clyde needed to be deep enough for ships, so walls were built to make the river narrower and faster. (Exhibit explanation in Clydebuilt, a museum devoted to Glasgow's industrial heritage)

When used in this way, there is nothing vague in words such as 'fast': the speed of a river is positively correlated with its depth, although nothing is said about the strength of this correlation. Similarly, the bathtub laws say that a higher value for Level implies a higher value for Outflow, and conversely. Nothing would be easier to express in classical logic. It is only if qualitative reasoning is combined with fuzzy logic that vagueness starts playing a role.

An area of AI where vagueness really does take centre-stage involves the construction of computer systems that give people *advice* about what to do in difficult situations: these systems offer decision support. To explain the role of gradability and fuzziness, we shall have to go into a fair amount of detail.

Applying fuzzy logic: An artificial doctor

Decision support systems are important in fault-critical domains such as medicine. Systems of this kind work with rules that give advice in various situations. These rules are often the result of extensive interviews with experts, but it is their combination that the computer program is responsible for. Doctors might, for example, inform the designer of the system of the following rules that are relevant for a particular cardiac operation:[5]

Rule 1: *If* blood pressure is low *or* body temperature is low *then* risk is low.
Rule 2: *If* blood pressure is moderate *and* body temperature is high *then* risk is moderate.
Rule 3: *If* blood pressure is high *then* risk is high.

Rules of this kind do not state ordinary conditionals as they are used in classical logic. (If they did, then contradiction would loom if blood pressure is high but body temperature low.) Also, Rule 3 does not simply divide patients into the ones that do and the ones that do not have high blood pressure; rather, it suggests that, as blood pressure increases, so does the risk involved in the operation, other things being equal. A natural way to understand these rules arises if we take a fuzzy logic perspective. Let us see how this can be done.

Suppose, for simplicity, that Rules 1–3 are the only rules, so each of the three risk levels, low, moderate, and high, is affected by only one rule. Let us see how rules of this kind can produce conclusions about a patient. In doing so, we shall make use of one of the oldest proposals in this area, by Ebrahim Mamdani and his colleagues in London in the 1970s. Once the rules are in place and basic measurements have been taken, Mamdani-style inference proceeds in four steps, called fuzzification, rule evaluation, output aggregation, and defuzzification. Suppose the patient's blood pressure is 150/90 and her body temperature is 37.7 °C. (See Fig. 19, where the two figures that make up a blood pressure reading are averaged. Similar graphs are assumed to be available for body temperature and risk level.) Here is how her level of risk is computed.

1. *Fuzzification.* The first step is to 'interpret' the measurements taken from the patient in terms of the concepts featuring in the rules. The rules are cast in terms of blood pressure and risk being low/moderate/high, and temperature being low/high. The price to be paid for this subtlety is that degrees of truth must now be assigned to *each* of these three values: we have to say to what extent the patient's blood pressure counts as *low*, and to what extent as *moderate*, and to what extent as *high*. Let us suppose that, given her age and other factors, the patient's blood pressure counts as v(Pressure is high) = 0.5, v(Pressure is moderate) = 0.2, and v(Pressure is low) = 0. (See Fig. 19. Note that fuzzy logic does not require these values to add up to 1.) Suppose her body temperature is such that v(Temperature is low) = 0.1 and v(Temperature is high) = 0.7. In other words, our patient's blood pressure and body temperature are both on the high side.

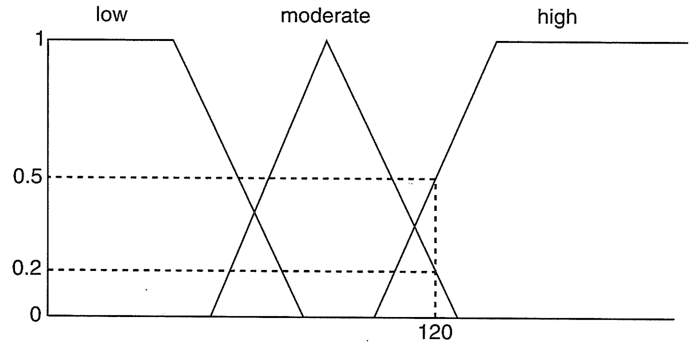

FIG. 19 Blood pressure in a complex membership function (average between systolic and diastolic values)

2. *Rule evaluation.* The results of the previous step determine to what extent each of the three rules 'fires'. The most straightforward is Rule 3, which has only one precondition, that *pressure is high*. Since fuzzification tells us that v(Pressure is high) = 0.5, this rule fires to degree 0.5. As a result, the conclusion offered by this rule, that *risk is high*, also has a value of 0.5. The same technique is applied to Rules 1 and 2, but a few wrinkles need to be ironed out, because these rules involve *compound* preconditions, involving 'and' or 'or'. Rule 1, for example, has the precondition *blood pressure is low* or *body temperature is low*, whose first part has the value 0 and whose second part has the value 0.1. The value

of a disjunction equals that of the 'truest' of its parts, as we have seen, which is 0.1, meaning that this rule is only marginally relevant. Rule 2 is only slightly better off, because it fires to a degree of 0.2, the *minimum* of 0.7 and 0.2.

3. *Output aggregation.* We now know that the risk from the operation is *high* to degree 0.5, *moderate* to degree 0.2, and *low* to degree 0.1. This information is now combined into one complete membership function, which says for each risk level between 0 and 1 to what degree this risk level applies to our patient. As always, this information can be displayed in a membership graph.

4. *Defuzzification.* So how much is our patient at risk? Mamdani assumed that the take-home message for doctors should not take the form of a complex membership graph (as delivered by the previous step) or vague words such as 'high' or 'low', but a single *number*: a fuzzy truth value between 0 and 1. One way to find a suitable number is by drawing a vertical line through the graph of the membership function right where it cuts the area under the graph in two equal halves. The location of this line will tell us that, in the situation of our example, the risk is slightly above 0.5.

These techniques—which we have sketched loosely here, glossing over many mathematical details in the last two steps—represent only one way of doing things. At every step, sensible alternatives exist. This is perhaps easiest to appreciate in connection with *fuzzification*, but it is also true for *rule evaluation*. It is, for example, not obvious that we have done justice to Rule 2 by taking the minimum of the values 0.2 (for the degree to which pressure was moderate) and 0.7 (for the degree to which temperature was high) since, in the end, this failed to take temperature into account entirely. Complications with *defuzzification* arise if several rules address one and the same factor, in which case a way should be found to deal with accrual of evidence. (If three people tell you that the butler is the murderer, this should carry more weight than if only one person does.) In all these cases, alternatives exist. This is no coincidence: once truth values proliferate, then so do the possibilities for handling them. This is an embarrassment of riches that fuzzy logicians have to live with.

The future of AI

In middle age, after its first half century, artificial intelligence has become more like other science subjects. Exaggerated promises are seldom heard, except in the popular media, where robots go on conquering the world. A new realism has turned AI into a research area some of whose most application-oriented parts are closely linked with the computing industry, while its more theoretical parts are essentially an area of psychology. Debates about the limits of AI (Can computers be conscious? Can they have emotion? Could they ever become as intelligent as people? Do robots have rights?) flare up from time to time, but conclusions are not in sight. We leave these issues to better minds.

It would be ridiculous to suggest that *all* the obstacles facing AI stem from vagueness, but vagueness is implicated in a number of them. We have seen that this is true for computer chess, and it can certainly be argued in the case of AI work on communication, as we shall soon see in more detail. Other examples abound. Consider visual *object recognition*, for instance, which is the task of allowing computers to recognize particular types of objects, such as guns for example (in airport security), landmines (for mine clearance), or people. One of the most serious problems in this area is that the types of object in question are often difficult to define. Consider *people*, for example. The possible dimensions of a person are virtually impossible to state with any precision, and any attempt would tend to include false positives (plastic fashion dolls taken to be people) and false negatives (people who fail to be recognized as human because they are unusually thin). The challenges to automatic recognition of speech and handwriting, both of which are key problems in AI, are not dissimilar. It would be reasonable, in my view, to hope that a better understanding of vagueness might shed light on some of these other problems as well.

Given that the boundaries of artificial intelligence are only loosely defined, it is difficult to quantify the extent to which today's world is already applying AI techniques. Yet, some people have tried. A 2003 report by the Business Communication Co., for example, estimated the global market for AI at close to $12 billion. What is clear is that there are important applications in areas such as fraud detection (e.g. detection of credit card fraud), security (e.g. automatic analysis of email conversations), car navigation (e.g. route finding and description), and the medical domain (e.g. decision support and data-mining). Defence is probably the biggest single investor, reportedly with huge pay-offs: it has been estimated that the deployment of a single logistics support aid during the Desert Shield and Storm Campaigns against Iraq in 1990–1 paid back all US government investment in AI research over a thirty-year period. Whether these are the best possible uses of AI is a question I will not attempt to answer.

As for *communication* with computers, there is a long way to go before computers might ever be able to use language in a way that resembles human speaking or hearing, but progress is being made all the time. It is this area of AI on which the next chapter will focus.

Things to remember

- Artificial intelligence (AI) is an area of computer science in which computer programs are constructed that mimic human abilities. This can be done either for practical purposes (i.e. because the program performs a useful task) or because a computer program is an interesting model of a thinking agent.

- AI has become strikingly successful in recent years, particularly in areas such as computer chess, planning, robotics, and decision support.

In many cases, however, the most successful computer programs work quite differently from people. They may emulate *what* people do, but seldom *how* they do it.

- The drawbacks of fuzzy logic that were identified in our theoretical discussions of Chapter 9 turn out to plague practical applications of fuzzy logic as well. This became clear in our discussion of decision support systems, which could produce advice only once a programmer had decided how to define the logical connectives of the system.

- Qualitative reasoning systems perform automated reasoning with physical quantities by focusing on broad tendencies (e.g. a variable whose value increases, decreases, or stays the same) while disregarding quantitative detail. Because of their ability to suppress details, qualitative reasoning systems can provide interesting and useful models of people's understanding of physical and other regularities.

- Vagueness and a number of problems structurally very similar to vagueness are implicated in many of the greatest obstacles facing AI at the moment.

- Artificial intelligence has experienced a peculiar reversal: tasks that were originally thought to be hard (such as chess, for example) have turned out to be much easier to tackle than some tasks that are sometimes thought to be easy (such as interacting socially with one's environment). It can be argued that this experience should have implications for the testing of human intelligence, where more emphasis might be placed on *social* abilities than is commonly done.

- Communicating through human languages (such as English or Chinese, for example) is one of the AI tasks that are harder than anticipated. One aspect of human communication will be discussed in the next chapter.