

An aerial photograph of a long, straight road that stretches from the bottom left towards the horizon. The road is flanked by fields and some distant trees. The sky is a clear, pale blue. The overall tone is serene and expansive.

COMMON SENSE, THE TURING TEST, AND THE QUEST FOR REAL AI

HECTOR J. LEVESQUE

Common Sense, the Turing Test, and the Quest for Real AI

Common Sense, the Turing Test, and the Quest for Real AI

Hector J. Levesque

**The MIT Press
Cambridge, Massachusetts
London, England**

© 2017 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

This book was set in Stone Sans and Stone Serif by Toppan Best-set Premedia Limited. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Names: Levesque, Hector J., 1951- author.

Title: Common sense, the Turing test, and the quest for real AI / Hector J. Levesque.

Description: Cambridge, MA : MIT Press, [2017] | Includes bibliographical references and index.

Identifiers: LCCN 2016033671 | ISBN 9780262036047 (hardcover : alk. paper)

Subjects: LCSH: Thought and thinking. | Intellect. | Computational intelligence. | Artificial intelligence--Philosophy.

Classification: LCC BF441 .L483 2017 | DDC 006.301--dc23 LC record available at <https://lcn.loc.gov/2016033671>

10 9 8 7 6 5 4 3 2 1

For Pat

Contents

Preface ix

Acknowledgments xiii

1	What Kind of AI?	1
2	The Big Puzzle	15
3	Knowledge and Behavior	29
4	Making It and Faking It	45
5	Learning with and without Experience	65
6	Book Smarts and Street Smarts	77
7	The Long Tail and the Limits to Training	87
8	Symbols and Symbol Processing	101
9	Knowledge-Based Systems	115
10	AI Technology	129

End Notes 143

References 157

Index 169

Preface

This is a book about the mind from the standpoint of artificial intelligence (AI). But even for someone interested in the workings of the human mind, this might seem a bit odd. What do we expect AI to tell us? AI is part of computer science after all, and so deals with *computers*, whereas the mind is something that *people* have. Sure, we may want to talk about “computer minds” someday, just as we sometimes talk about “animal minds.” But isn’t expecting AI to tell us about the human mind somewhat of a category mismatch, like expecting astronomy to tell us about tooth decay?

The answer is that there is not really a mismatch because computer science is not really that much about computers. What computer science is mostly about is *computation*, a certain kind of process, such as sorting a list of numbers, compressing an audio file, or removing red-eye from a digital picture. The process is typically carried out by an electronic computer of course, but it might also be carried out by a person or by a mechanical device of some sort.

The hypothesis underlying AI—or at least one part of AI—is that ordinary *thinking*, the kind that people engage in every day, is also a computational process, and one that can be

studied without too much regard for who or what is doing the thinking.

It is this hypothesis that is the subject matter of this book.

What makes the story controversial—and more interesting, perhaps—is that there is really not just one kind of AI these days, nor just one hypothesis under investigation. Parts of AI do indeed study thinking in a variety of forms, but other parts of AI are quite content to leave it out of the picture.

To get an idea of why, consider for a moment the act of balancing on one leg. How much do we expect thinking or planning or problem-solving to be involved in this? Does a person have to be knowledgeable to do it well? Do we suppose it would help to read up on the subject beforehand, like *Balancing on One Leg for Dummies*, say? An AI researcher might want to build a robot that is agile enough to be able to stand on one leg (for whatever reason) without feeling there is much to be learned from those parts of AI that deal with thinking.

In fact, there are many different kinds of AI research, and much of it is quite different from the original work on thinking (and planning and problem-solving) that began in the 1950s. Fundamental assumptions about the direction and goals of the field have shifted. This is especially evident in some of the recent work on machine learning. As we will see, from a pure technology point of view, this work has been incredibly successful, more so, perhaps, than any other part of AI. And while this machine learning work leans heavily on a certain kind of statistics, it is quite different in nature from the original work in AI.

One of the goals of this book is to go back and reconsider that original conception of AI, what is now sometimes called “good old-fashioned AI,” and explain why, even after sixty years, we still have a lot to learn from it—assuming, that is, that we are

still interested in exploring the workings of the mind, and not just in building useful pieces of technology.

A book about AI can be many things. It can be a textbook for an AI course, or a survey of recent AI technology, or a history of AI, or even a review of how AI has been depicted in the movies. This book is none of these. It is a book about the ideas and assumptions behind AI, its intellectual underpinnings. It is about why AI looks at things the way it does and what it aspires to tell us about the mind and the intelligent behavior that a mind can produce.

It is somewhat disheartening to see how small this book has turned out to be. The entire text will end up taking about 150KB of storage. For comparison, just one second of the Vivaldi concert on my laptop takes more than twice that. So in terms of sheer raw data, my laptop gives equal weight to this entire book and to a half-second of the video, barely enough time for the conductor to lift his baton.

This can be thought as confirming the old adage that a picture—or a single frame of a video—is worth a thousand words. But I think there is another lesson to learn. It also shows that compared to pictures, words are amazingly compact carriers of meaning. A few hundred words might be worth only a momentary flash of color in a video, but they can tell us how to prepare boeuf bourguignon. The human animal has evolved to be able to make extremely good use of these ultra-compact carriers of meaning. We can whisper something in somebody's ear and be confident that it can have an impact on their behavior days later.

How the mind is able to do this is precisely the question we now set out to explore. My hope is that the reader will enjoy my thoughts on the subject and get to feel some of the excitement that I still feel about these truly remarkable ideas.

Acknowledgments

I made my living writing technical papers about AI, more or less. So you might think that writing something nontechnical like this book would be a piece of cake, a stroll in the park. Uh, no. This book was certainly a labor of love, but it was harder for me to write than anything technical.

I was fortunate to have a number of people help me with it. Ernie Davis encouraged me to get started in the first place, and was kind enough to read and comment on a rough first draft. I was also glad to get comments on a subsequent draft from Toryn Klassen, my brothers John and Paul, my daughter Michelle, and four anonymous reviewers. They helped to at least steer the book in a direction, since apparently I had somehow managed to leave port without much of an idea of where I was headed. I'm also grateful for comments on a later draft from Gerhard Lake-meyer, Don Perlis, Vaishak Belle, and Gary Marcus. The book is much better thanks to all their efforts. But of course, they are not to blame for any errors and confusions that remain.

I also want to thank the people at the MIT Press. Marie Lufkin Lee was kind and encouraging even when I had nothing more than a sketchy draft. Kathleen Hensley managed a lot of the practical details, and Michael Sims handled the copyediting,

transforming my shaky grammar and wording into something more sturdy.

I wrote some of the book while I was a visiting researcher in Rome. I want to thank Giuseppe de Giacomo and the good folks at the Sapienza—Università di Roma for their friendship and hospitality. I actually completed the bulk of the work after I had retired from the University of Toronto. To have the chance to put as little or as much time as you want on a project of your own devising, how good is that?

I would also like to thank my family and friends who cheered me on from the start. “Why don’t you try writing something that normal people can read?” they would sometimes ask, or more diplomatic words to that effect. This book might be the closest I ever get, even if it remains a tad academic in spots. But in case I don’t get a chance to do anything like it again, I want to express my gratitude more broadly.

Let me start by thanking my brother Paul who drove me to Toronto to see the movie *2001: A Space Odyssey* when it first came out. As a high-school student, I’m not sure I “got it” at the time, but I did get the impact, the incredible attention to detail. While others were marveling at the trippy aspects of the movie (it being the late sixties and all), I was marveling at the idea of a computer like HAL. It changed my life, and you will see references to *2001* throughout this book.

I began as a university student soon thereafter, thanks to the efforts of a high-school guidance counselor who got me a scholarship in engineering. But I found engineering incredibly tedious, demanding but not rewarding, except for a single course on computers. I knew at that point that Marvin Minsky, who had consulted on *2001*, was on the faculty in engineering at MIT. So I wrote him a letter asking about this. I want to thank

the late Professor Minsky for taking the time and trouble to write to a first-year undergraduate—this was well before the days of email!—encouraging him to get out of electrical engineering and into computer science. I might not have been able to walk away from my engineering scholarship had my grandfather not left money for his grandchildren's education. But he did, and so I walked, and this too changed my life.

In computer science, I was very fortunate to come under the wing of John Mylopoulos, the one faculty member in Toronto interested in AI at the time. He taught a new course on AI that I took and loved, he gave me a summer job, he took me on as a graduate student, and he helped me very patiently as I meandered toward a PhD topic some distance away from his area of interest. He also introduced me to Ron Brachman, who was visiting from Boston. Ron was kind enough to offer me a summer job in Boston, and after my PhD, a job in California, where I got to work with him for three glorious years. John also worked very hard to get me an academic position back in Toronto, which ended up being a dream job for me for the rest of my career. John and Ron are the only two bosses I have ever known, and I owe my career to the two of them.

I can hear the music in the background getting louder, so let me wrap this up by thanking my dear wife Pat for having been there for me throughout my career, through ups and downs, from undergrad to pensioner. Like it says on the first page, this one is for her.

Toronto, July 2016

1 What Kind of AI?

This is a book about the mind from the standpoint of artificial intelligence (AI). We will have a lot to say about the mind, but first, we turn to the area of AI itself and briefly introduce the ideas that will be covered in more depth in the rest of the book.

Adaptive machine learning

When it comes to technology, AI is all the rage these days. Hardly a week goes by without some mention of AI in the technology and business sections of newspapers and magazines. The big computer companies like Microsoft, IBM, Google, and Apple are all heavily invested in AI research and development, we are told, and many others are following suit. In November 2015, Toyota announced a \$1B investment in AI, and in December 2015, Elon Musk announced a new nonprofit company called OpenAI with yet another \$1B of funding.

What is all this buzz about anyway? If you look in more detail at what those billions of dollars are expected to lead to, the technology appears to be quite different from the sort of AI previously imagined in science-fiction books and movies. There is very little talk of intelligent humanoid robots (like those seen in

the movie *Blade Runner*, say) or even of a high-powered disembodied intelligence (like the HAL 9000 computer in the movie *2001: A Space Odyssey*).

The kind of AI being contemplated by the technology companies today might better be called *adaptive machine learning*, or AML. The idea in AML, very generally speaking, is to get a computer system to learn some intelligent behavior by training it on massive amounts of data. Indeed, the current excitement about AI leans heavily on the potential application of what is usually referred to as “big data.”

This is not the place to try to explain in any great detail how AML technology is supposed to work, but here is the rough idea.

Imagine, for the sake of argument, that we are interested in building a computer system that can recognize cats. Its job, in other words, will be to classify pictures it is given into two groups: those that depict cats and those that do not. The question then is how to build such a system. In the past, AI programmers would have tried to write a program that looked for certain specific cat features in an image. It might look for a cat face: greenish or yellowish eyes with almond-shaped vertical pupils, an inverted pinkish triangle making up the nose, whiskers, and so on. Or perhaps it might look for a cat outline: a small head, triangular cat ears, four legs, and a tail that often points straight up. Or it might look for the distinctive coloring of cat fur. If the program spots enough of these features in an image, it would label the image as one of a cat, and otherwise reject it.

But AML suggests a very different way of doing the job. You start by giving your system a very large number of digitized images to look at, some of which have cats in them and some of which do not. Then you tell your system to *compress* all this

visual data in a certain way, that is, to look for a set of “features” that are seen in many patches of many images. A feature might be an area of a certain uniform color and brightness. It might be an area where there is a distinctive change in brightness and color corresponding to an edge. The idea is to find a set of features such that it is possible to reconstruct something similar to the original images as combinations of those features. Then you tell your system to abstract from those features, and to look for common features of those features. And so on for a few more levels.

All this is done without telling the system what to look for. If the original images had many cats in them, there is a good chance that the system will end up with some high-level cat-like features. What’s important is that these features are determined by the images themselves, not by what an AI programmer thought was important about recognizing a cat in them. As Andrew Ng from Stanford puts it, “You throw a ton of data at the algorithm and you let the data speak and have the software automatically learn from the data.”

This “unsupervised” approach to AML has been found to work extremely well, beyond anything AI researchers might have predicted even just a few decades ago. The success of AML is usually attributed to three things: truly massive amounts of data to learn from (online, in specialized repositories, or from sensors), powerful computational techniques for dealing with this data, and very fast computers. None of these were available even thirty years ago.

Of course, nobody wants to spend billions of dollars on cat recognition. But imagine that instead of cats, the pictures are mammograms, some of which contain tumors that may be hard for doctors to see. Or imagine that the data is not visual at all. It

might be made up of sound recordings, some of which contain words spoken by a person of interest. Or imagine that the data involves banking transactions, some of which involve fraud or money laundering. Or maybe the data involves groups of online shoppers characterized by their browsing and purchase history. Or maybe the data is automobile pedal and steering-wheel activity in response to visual data from the windshield. Systems that can learn automatically from massive amounts of data might be applied in a variety of areas of great economic and social significance.

Good old-fashioned AI

Exciting as this AML technology may turn out to be, it is *not* what this book is about. In fact, as will become clear, we will have very little to say about AI technology of any kind, except at the very end, in chapter 10.

What we will mostly be concerned with in this book is a rather different vision of AI first proposed by John McCarthy, Marvin Minsky, Allen Newell, Herb Simon, and others in the 1950s, something we call *Good Old-Fashioned AI*, or GOFAI for short. These researchers emphasized a different sort of intelligence, one based not on learning from massive amounts of data, but on *common sense*.

What is common sense? How is having common sense any different from being well trained on large amounts of data? This will take much of the book to sort out. One difficulty in trying to be clear about the difference is that in our normal everyday lives, we spend almost all of our time in situations that are at least somewhat familiar to us. We rely to a very large extent on our routines, patterns of behavior that we have learned over time. I

usually pick up a cup of coffee on my way to work. I usually greet my coworker George when I see him in the hall. I usually put mustard on hot dogs. Even if I happen to see a tree I have never seen before, it is still close enough to what I have seen for it to feel quite familiar to me. There is certainly no surprise or disorientation.

However, as we will see in much more detail in chapter 7, common sense enters the picture when we are forced to act in situations that are sufficiently unlike the patterns we have seen before. What exactly do I do on that one morning when the door to my coffee shop does not open? Do I just keep tugging at it? What do I say to my colleague George if the first thing he says to me is “Did you hear about the nasty chemical spill on the third floor?” Do I just say “Good morning” as usual and move on? What do I do if I am given a hot dog that is already fully loaded with condiments? Put more mustard over the whole mess? It is how we behave in new, unfamiliar situations, when we say to ourselves “Hold on a minute, what’s going on here?” that really shows whether or not we have common sense.

The question that motivates the researchers in GOFAI is this: What accounts for this common sense? How do people use it to figure out what to do when their usual routines no longer apply? What is this “figuring out” anyway? What goes into a decision about what to do with a door that won’t open? When I decide to step outside of a building without a coat on a cold winter’s day just because of something I was told, how does that work? How is that one person making noises with his mouth—saying certain words—can be enough to cause me to behave in a way that is totally unlike my normal routine?

This is the mystery that early AI researchers were intrigued by and *this* is what this book is about.

Programs with common sense

The early pioneers in AI met at a conference at Dartmouth College in the summer of 1956. This is where AI was first given its name by the organizer of that conference, the American computer scientist John McCarthy (1927–2011). A direction for research was laid out in a brilliant and quite unprecedented paper he wrote in 1958 entitled “Programs with Common Sense.”

We will have more to say about the technical aspects of this paper in chapter 9, but one thing it made explicit was a focus on common sense. McCarthy had something very definite in mind:

We shall therefore say that a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.

As we will see again in chapter 3, McCarthy was especially intrigued by how people use what they know and what they have been told to figure out what to do.

However, even from the very start, common sense was not considered to be something distinct from learning. McCarthy also says: “Our ultimate objective is to make programs that learn from their experience as effectively as humans do.”

Clearly we do learn something by trying to open a door that happens to be locked. We learn from our coworker George about a chemical spill. We learn by direct observation that a hot dog is fully loaded.

Yet the emphasis in GOFAI is different from AML. As we will see in chapter 5, learning in GOFAI need not involve large amounts of training data, and it often deals with *language*. John McCarthy again: “In order for a program to be capable of

learning something it must first be capable of being told it.” The primary focus is very clearly on the linguistic aspects of learning, and he called his proposed system the “advice taker.”

Indeed, GOFAI is sometimes criticized as being overly preoccupied with symbols and words. Intelligent behavior, the critics rightly point out, need not be something you can put into words. You might be able to ride a motorcycle quite well without being able to say in words how far over you should lean on a tight curve. You might be able to pick out two brothers in a group of boys without being able to articulate what it is that allows you to spot that family resemblance.

And yet, it is true that human language holds a position of honor in GOFAI. And there are two very good reasons for this. First, as will be argued in chapter 6, quite apart from communication, language plays a very special role in human behavior, a role not seen in other animals: much of how we deal with new situations involves using what we have read or been told earlier using language.

Second, language is a superb medium for exploring intelligent behavior. If we want to find out how a person would deal with a new and unfamiliar situation, we can ask him or her. While it is possible to exhibit intelligence without saying a thing, it certainly requires intelligence to be able to use language the way we do, and so it is a supreme test of intelligence, something first suggested by Alan Turing.

The Turing Test

Alan Turing (1912–1954) was a British mathematician involved in the early development of computers. (We will have more to say about him and his role in chapter 8.) He was one of the first

to consider seriously whether computers could be programmed to do tasks that required intelligence in people, such as playing a reasonable game of chess. While it is true that computers can do only what they are programmed to do, it was far from clear then—and it is still far from clear today—just what we could program them to do.

I suspect that Turing grew tired of philosophical discussions about whether a machine would ever be able to think, or understand something, or have consciousness. He foresaw some of the incredible technical challenges posed by AI, but was perhaps exasperated by arguments to the effect that even if all these challenges could be overcome, the resulting computer system would never be said to think or understand or be conscious for other reasons he had no control over, such as the fact that the computers were not biological.

In a very influential 1950 paper, Turing suggested that we should shift our attention away from how the machine was made, what it looked like, or what may or may not be going on inside it, and concentrate instead on its *externally observable behavior*. Of course, there are different sorts of behavior that we might want to consider. We might put the machine in some new and unfamiliar situation and see what it does. We might ask it to recognize pictures of cats. We might study what it does with a fully loaded hot dog. What Turing came up with was the idea of participating in a no-holds-barred conversation with an interrogator.

This is what he called the Imitation Game (described in the 2014 movie of the same name). The idea is that there would be an extended conversation over a teletype between an interrogator and two hidden participants, a person and a computer. The conversation would be natural, free-flowing, and about any

topic whatsoever. The computer would win the game if no matter how long the conversation, the interrogator could not tell which of the two participants was the person. In current parlance, the computer is said to have *passed the Turing Test*.

It is important to remember that as far as Turing was concerned, the conversation could be about anything. Here is a typical conversation imagined in his paper:

Interrogator: In the first line of your sonnet which reads “Shall I compare thee to a summer’s day,” would not “a spring day” do as well or better?

Computer: It wouldn’t scan.

Interrogator: How about “a winter’s day?” That would scan all right.

Computer: Yes, but nobody wants to be compared to a winter’s day.

Interrogator: Would you say Mr. Pickwick reminded you of Christmas?

Computer: In a way.

Interrogator: Yet Christmas is a winter’s day, and I do not think Mr. Pickwick would mind the comparison.

Computer: I don’t think you’re serious. By a winter’s day one means a typical winter’s day, rather than a special one like Christmas.

We are, of course, very far away from a computer program that can carry on a conversation like this. But is there some reason to believe that we will never be able to write such a program? *This* is what Turing suggests we should focus on.

Turing’s point in all this, it seems to me, was the following: These mental terms like “intelligent,” “thinking,” “understanding,” and so on are really too vague and emotionally charged to be worth arguing about. If we insist on using them in a scientific context at all, we should be willing to say that a computer that can pass a suitable behavioral test has the property in question as much as a person, and leave it at that. Adapting the dictum of the title character in the movie *Forest Gump* who

said “Stupid is as stupid does,” we can imagine Turing saying “Intelligent is as intelligent does.” In other words, instead of asking “Can a machine think?” we should really be asking “Can a machine behave like a thinking person?”

The Chinese Room

Turing’s proposal is perhaps an obvious one, at least in retrospect. His emphasis on externally observable behavior is really at the heart of all AI research, as we will see in the next chapter. (We shall have reason to question the use of an *informal conversation* in a test for intelligence, but we will get to that in chapter 4.) Nonetheless there have been detractors. We will briefly discuss one objection to Turing’s proposal that came out of the philosophical literature. (Readers less interested in philosophical wrangling should jump ahead to the next chapter.)

In the 1980s, the philosopher John Searle argued that there was much more to understanding (or thinking, or being intelligent) than merely getting some observable behavior right, even if that behavior is as general and versatile as the Imitation Game. Here is (a slight variant of) his argument.

Imagine that there already exists a computer program that is somehow able to pass the Turing Test, that is, that can engage in an unrestricted conversation as well as a person. Let us suppose that this conversation takes place in Chinese, not in English. So in accordance with the Imitation Game, Chinese characters are presented to this program (encoded in some way) as input, and the program is able to produce Chinese responses (also encoded in some way) as output that cannot be distinguished even over a long period of time from those of a person who really understands Chinese.

Let us now imagine that Searle himself does not know Chinese, but knows computer programming very well. (In reality, he does not, but no matter.) He is hidden in a room with a book that contains the complete text of the computer program. When someone outside the room slips him a piece of paper with Chinese written on it, he does not understand what it says, but he can simulate what the computer program would do with it. He traces the behavior of the program using the book, writes on a piece of paper the output that the program would produce, and hands that back, again without understanding what any of it means.

So what do we have here? There is a person, John Searle, who is receiving Chinese messages in his room and using the book to produce Chinese responses that are indistinguishable from those of a native Chinese speaker (since the computer program is assumed to pass the Turing Test). His externally observable behavior, in other words, is *perfect*. And yet Searle does not understand Chinese! Searle's conclusion: getting the behavior right is not enough, and so Turing is wrong.

An objection to this argument is that it is not Searle who is producing this behavior, but Searle together with the book containing the text of the computer program. Although Searle does not understand Chinese, the overall *system* consisting of Searle and the book does. So Turing is not wrong. Searle's reply to this objection is beautiful in its simplicity: imagine that he *memorizes* the book and then destroys it. Then there is no longer a system to talk about; there is just Searle. So Turing *is* wrong.

This is the Chinese Room argument, in a nutshell.

What is there to say about it? Understandably, there is impatience with it in the AI community. There is much more interest in AI in the technical challenges involved in producing such a

computer program (assuming it is possible at all), than in what conclusions we would draw about a person in a room who can accurately simulate the running of such a program (maybe millions of times more slowly).

But in terms of the thought experiment itself, here is one thing to think about: how can we be so sure that Searle does not come to *learn Chinese* by memorizing the book? If he ended up knowing Chinese afterward, the argument about passing the Turing Test without understanding Chinese would no longer apply. So we have to ask: why should we believe that there can be a book that does the job Searle wants *and* that can be memorized without learning to speak Chinese?

It's tough to answer this question one way or another without a better idea of what the computer program in the book would have to be like. One argument I have made in the past involves turning to a different and much simpler form of behavior, the summation of numbers.

Here's my idea. Imagine that instead of speaking Chinese, the behavior under examination involves the ability to add twenty ten-digit numbers, no more, no less. A book listing every possible combination of twenty ten-digit numbers and their sums would allow a person who does not know how to add to get the summation behavior right. Any time the person is asked what a sum is, the correct answer could be found by looking it up in the book, just like Searle suggested for Chinese.

The main thing to observe about such a book is that it cannot exist. It would need to contain 10^{200} distinct entries for all the combinations of numbers, but our entire physical universe only has about 10^{100} atoms.

There is another smaller book that would do the job, however, which is an English language description of how to add:

first, a 10×10 table for the addition of single digits, then the right-to-left process of addition for two numbers (with carry digits), and finally, the process of multinumber addition. This second book can definitely exist; it would only be a few pages. What is especially interesting about this second book is that a person who does not know how to add but who memorizes the instructions in the book would thereby learn how to add!

This is enough to cast doubt on Searle's Chinese Room argument. If there is no book that would allow a person to get something as simple as the summation behavior right without also teaching the person how to add, or at least how to add twenty ten-digit numbers, why should we think that conversing in Chinese will be any different?

However, this doesn't refute Searle's argument either. In the end, we really don't know what it would be like to memorize Searle's book because we don't know what a computer program for Chinese would need to be like. And the only way to find out is to tackle those technical challenges, just as Turing suggested.

2 The Big Puzzle

Intelligent behavior in people is a product of the mind. But the mind itself, of course is not a thing; it is more like what the human brain *does*. The actual physical object that lies behind our external behavior is the brain.

The problem

Although we know much more about the human brain than we did even ten years ago, the thinking it engages in remains pretty much a total mystery. It is like a big jigsaw puzzle where we can see many of the pieces, but cannot yet put them together. There is so much about us that we do not understand at all. Just how are we different from the rest of nature? What makes us so darn smart?

People in different areas of research come to this puzzle from different angles, each holding some of the pieces. Not too surprisingly perhaps, each of them argues that *their* perspective is the important one, the one that really matters. Here is the sort of thing you might expect to hear:

- What we need to care about is *language*, since the presence of a mind is most clearly manifest in the ideas it expresses. We know a mind by listening to what it says, reading what it writes. We are unique in the animal kingdom in recording what we know, what we have learned, how we think. We will never get to study the brain of Shakespeare, but we get to experience the mind of Shakespeare every time we tackle one of his plays.
- What we need to care about is *psychology*, how a mind works in people and how that compares and contrasts with other animals. People's intuitions about how they think is quite different from how they really do think when studied closely in a controlled setting. Memory, learning, perception, attention, cognition, affect, consciousness, these are all psychological categories and together make up the mental life of people.
- What we need to care about is *neuroscience*, how a brain organizes itself to make a mind. When we talk about the mind, what we are really talking about are the states and transitions of a large assembly of highly connected electrochemical components. As we learn more and more about the brain—and the recent progress here has been phenomenal—there will be a revolution in how we talk about the mind, just as there has been a revolution in how we talk about life since the discovery of DNA.
- What we need to care about is *evolution*, how the human species developed under evolutionary pressure. Nothing in the human species, including its brain, stands alone. Every human faculty is the product of evolution, and correlates can always be found in other species. The human mind is a Swiss army knife of adaptations, and one cannot understand it without

understanding how it evolved and how it connects to reproductive success.

And there are others.

I believe a certain amount of humility is called for when we talk about things like the brain, the mind, thinking, and intelligence. Listening to scientists brimming with confidence in full promotion mode, it may appear that new ideas, new approaches, new techniques, new tools are putting us on the threshold of an era where we will crack the mystery. I think this is quite wrong. The mistake, in fact, is to think that there is a single mystery to crack. What we have is a *collection* of mysteries, a jigsaw puzzle of pieces that need to be brought together and assembled from many different perspectives before we can truly appreciate the overall picture.

We need to train ourselves to be skeptical of any research group that insists that one part of this puzzle is the true core, the key to the whole thing. We should be even more skeptical of anyone who claims something like “We are on the verge of figuring out how the brain works.” This should sound to us like someone claiming to have figured out how weather works, or how the stock market works.

Just for reference, let us call the issue of confusing a handful of pieces for the entire puzzle the *Big Puzzle* issue. (It will come up again often.)

Given this Big Puzzle issue, how are we then to talk about thinking at all without falling into pompous oversimplifications? I believe the answer is to accept that we are dealing with only one part of the puzzle, and to do what we can to see how the pieces we are holding fit together, while resisting the temptation to suggest that the rest of the puzzle is just more of the

same. The mind is staggeringly complex, but this should not rule out looking at a small part of it in detail.

So what part of the puzzle will be discussed in this book? To jump the gun somewhat, this is the story that will be told here:

- What we need to care about is *intelligent behavior*, an agent making intelligent choices about what to do. These choices are made intelligent through the use of background information that is not present in the environment the agent is acting in. This background information is what we call *knowledge*. The application of knowledge to the behavior at hand is what we call *thinking*. The problem that needs to be sorted out is how, in concrete terms, this all works, how background knowledge can make a difference in an agent deciding what to do. The solution we consider is a *computational* one. In the same way that digital computers perform calculations on symbolic representations of numbers, we suggest that human brains perform calculations on symbolic representations of knowledge, and then use the result of those calculations to decide how to act.

It should be clear that this story is very far from proposing a solution to the entire puzzle. If it were, where would emotions like modesty, envy, and grief fit in? What about the social or interactional aspects of intelligence? Where is perception and imagination? Or daydreams and fantasies? What parts are uniquely human? What accounts for degrees of intelligence? And what about different kinds of intelligence (like emotional intelligence)? Or the effect of drugs and mental illness? And where is consciousness in all this? Or creativity? And what about spirit and pluck, the stuff that makes Molly Brown unsinkable?

These are all great questions that I will not be talking much about. Yet I do not want to give the impression that what is left

is small or beside the point. Thinking, even in the very narrow sense described above, is still as broad as the ideas that can be thought about. And we can think about anything! Despite all the amazing scientific progress to date, we still have no clear sense of what happens when we think about who will win the Academy Award for Best Actor, or how thinking about Debussy is similar to but different from thinking about Ravel, or what it is like to change your mind over a period of time about whether a free market needs to be regulated.

The mystery remains even in the small fragment of the puzzle considered here.

The difficulty

Before leaving this topic, it might be worth looking in more detail at why areas like psychology or neuroscience by themselves are unlikely to be able to tell us what we want to know about how the mind works. To make the point, we will use a thought experiment involving a simple machine that will be much easier to analyze.

Imagine a small device connected to a beeper, a lightbulb, and a keyboard. Anytime someone types a two-digit number on the keyboard, the machine responds with some beeps and some flashes. Call the machine *M*. Think of *M* as a very simple brain, with the keyboard as its only sense organ, and the beeper and lightbulb as its only effectors. Imagine that your job is to figure out why it is beeping and flashing the way it does. The exercise is useful because it gives us a good feel for what it might be like to figure out what is behind intelligent behavior.

Let us say, for example, that the numbers 37, 42, 53, 16, and 37 are typed in sequence on the keyboard. *M* produces the behavior shown in the following table:

Input	Beeps	Flashes
37	1	6
42	1	8
53	2	8
16	2	2
37	3	2

Now given all this, what makes it behave the way it does?

I am going to start by giving away the answer. *M* is a small digital computer attached to the keyboard, beeper, and light-bulb. It repeatedly takes as input the number typed on the keyboard, and produces as output a number of beeps and of flashes according to a tiny computer program.

The entire program that controls *M* is displayed in figure 2.1. (Readers who do not wish to try to read this small computer program can also skip the rest of this paragraph.) To see how it works, note that it uses integer arithmetic, where *mod* means the remainder after division. So the value of $37 / 10$ is taken here to be 3, and the value of $37 \bmod 10$ is 7. For example, when the 37 is first typed on the keyboard, *M* produces 1 beep and 6 flashes. Here's why: according to the program in figure 2.1, the *W* is set to 37, the *X* is set to 3, the *Y* is set to $3 \times 3 + 7 = 16$, where $16 / 10 = 1$ and $16 \bmod 10 = 6$. When 37 is typed on the keyboard the second time (the fifth input), the result is 3 beeps and 2 flashes because the *U* at that point is 5 (from the third input), and since $5 > 3$, the *Y* is set to $5 \times 5 + 7 = 32$.

So that's the secret. Now having seen it, pretend that you know nothing about the program in figure 2.1, but that your job is to understand why *M* does what it does.

We can imagine being a psychologist, running some experiments on *M* and observing its behavior. There are fewer than a

```
integer U, V, W, X, Y
set U to 0
set V to 0
repeat the following forever:
  set W to the next typed number
  set X to W / 10
  if X > U
    then set Y to X*X + (W mod 10)
    else set Y to U*U + (W mod 10)
  beep (Y / 10) times
  flash (Y mod 10) times
  set U to V
  set V to X
end repeat
```

Figure 2.1

The code for a mystery machine

hundred possible inputs here, but even in this very simple setting, our life is complicated by the fact that *M* has memory and makes decisions about how to behave based not just on the last number it saw.

To get an idea of what it is like for a real psychologist, we have to imagine that the number of tests we can run is much, much smaller than the range of possible inputs available to the machine. Think of a reading comprehension test (like the kind to be discussed in chapter 4), and compare the number of sentences in the test to the number of sentences that subjects will

read in their entire lives. Psychology must live with evidence drawn from very small parts of the total behavior space.

For example, let us simply generalize M so that instead of taking as input a two-digit number, it takes as input a ten-digit number. Now instead of 10^2 (one hundred) possible inputs, we have 10^{10} (ten billion) of them. If we observe that, because of memory, we may also have to consider what the machine saw in the previous step and the one before that, then we need to consider 10^{30} possible sequences. If the memory could go back further and depend on the ten most recent inputs, there are now 10^{100} sequences to sample from, more than there are atoms in the known universe.

So it does not take much to rule out the possibility of observing a significant range of stimulus and response. The sensory environment of M , simple as it is, and its memory, simple as it is, overwhelm any sort of comprehensive testing.

This, in a nutshell, is what makes psychology so hard.

It is extremely difficult to design an experiment constrained enough to provide meaningful results. If my subjects are Jim and Jane, how am I going to control for the fact that they have had very different lives, seen very different things, and come to my experiment with very different beliefs and goals? Not too surprisingly, some of the most revealing psychological experiments involve perceptual tasks where an immediate response is required, in the milliseconds, say. This is quick enough that long-term memory, which will be quite different for Jim and for Jane, plays a less significant role. When a subject gets to sit back and muse for a few seconds, it is extremely difficult to control for all the variables.

In a way, psychology is handicapped by the fact that it gets to look at subjects only from the outside. External stimuli can be

presented, and external responses can be observed, but that's it. It is not ethical to open up a living person's head and attach electrodes here and there to see what is happening. Some of our best understanding of brain function has come from cases where the brain was exposed as a result of an operation—severing the *corpus callosum* to control epilepsy, for example—and asking patients to describe what they sense when parts of the brain are stimulated.

However, there is new technology, such as fMRI, that is much less invasive than brain surgery and is giving us a much better picture of many parts of brain function. We can see that the part of the brain involved in motor control is activated even when a subject is just thinking about physical activities. We can see that the part of the brain that is active when a person is swearing is not the one that is normally active during polite language production. All these very impressive developments in neuroscience suggest that given sufficient time, we will understand how beliefs and goals come together in thinking to produce human behavior.

To see why we should nevertheless be skeptical about this, let us go back to *M*. Imagine a neuroscientist whose job it is to determine why *M* behaves the way it does. Like the psychologist, the neuroscientist is not told about the program in figure 2.1. Unlike the psychologist, however, the neuroscientist will be given access to the internal workings of *M*, as if it were a functioning brain.

When *M* is opened up and studied in the lab, the neuroscientist can see that it is an assembly of standard electrical components powered by a battery. As digits are typed, some of these components are activated and others stay dormant, perhaps only rarely lighting up. As more and more numbers are typed on

the keyboard, some tantalizing patterns begin to emerge. The question is: do we expect the neuroscientist to crack the puzzle about *M*'s behavior?

It is certainly true that *M* is nothing more than a collection of electrical components. Any behavior it produces is ultimately due to those components being in one state or another. If *M* were a brain we would say that the state of the brain is what determines how we behave; anything else we might choose to talk about (beliefs, goals, emotions, a mind, whatever) must be realized in one way or another in the state of the brain.

But the question is whether we will be able to recover the *regularities* in *M*'s behavior in the properties of *M*'s components. For example, we might want to be able to discover things like the fact that *M* squares the *first* digit in the number it sees but not the *second*. Will we see this in the electrical components?

There is good reason to believe that we will not. Let us suppose that the neuroscientist is talented and lucky enough to extract the entire program that *M* is running by carefully studying the state of all the electrical components over a period of time.

Here's the problem. *M* squares the first digit (or a previous first digit) because of lines 8 and 9 of the program in figure 2.1. But *that* program may not be located anywhere in *M*'s memory. It is typical of computers that programs are first translated into another form that is more easily executed by hardware. In computer jargon, the program in figure 2.1 is called the *source code*, and the translated version that is stored in the memory of *M* is called the *object code*. At the very best, the talented neuroscientist gets only the object code, and there is no reason to suppose that having the object code in hand would allow someone to recover the source code that generated it.

For example, squaring a number on a computer is typically not a single operation. Multiplication may show up as a large number of operations in object code. (There are clever forms of multiplication that perform much better than the digit-by-digit form we were taught in grade school.)

Even worse, the numbers themselves may not be encoded in a simple way in the states of the electrical components. Multiple components may be involved and they need not be physically close to one another. Indeed, in so called *distributed representations* (used in some neural net models of the brain) it may be necessary to look at the state of many electrical components to find the value of any one number represented. Worst of all, in a distributed representation, the state of a single component may be involved in the representation of more than one number.

So while translating from source code to object code is typically easy, going from object code back to source code is like trying to crack a big cryptographic puzzle. Companies that sell software products protect their intellectual property by releasing object code only, confident that this “reverse-engineering” is difficult enough that it cannot be solved in an economically feasible way. (Software products that are “open source” are counterexamples where the source code is made public as well.)

So even with the finest test equipment and electrical probes, the neuroscientist may not be able to recover *M*'s original source program. Having unrestricted access to all of the components in the lab, even when the components themselves are assumed to be well understood, is no guarantee that we can figure out something as simple as why *M* behaves the way it does.

This, in a nutshell, is what makes neuroscience so hard.

Even if we had total access to the hundred billion or so neurons making up a human brain, and even if we were able to treat those neurons as idealized, noise-free, digital components, we might still not be able to figure out why we behave the way we do. When a neuroscientist has to deal with *real* neurological components, not electrical ones, where there are a number of confounding chemical and biological processes going on, the situation is that much worse. How do we remember certain kinds of facts? How do we assemble those facts to draw new conclusions? How do we use these conclusions to decide how to behave? All of these are clearly tremendously more complex than figuring out why *M* is beeping three times. It is asking too much of neuroscience, even a wildly more advanced version of neuroscience, to tell us what we want to know.

We need to look elsewhere.

An approach

When faced with the problem of making sense of a complex phenomenon—and human thinking is certainly one of them—we actually have two options: we can study the objects that produce the phenomenon (that is, human brains), or we can attempt to study the phenomenon itself more directly.

Consider, for example, the study of *flight* (in the days before airplanes). One might want to understand how certain animals like birds and bats are able to fly. One might also want to try to build machines that are capable of flight. There are two ways to proceed:

- study flying animals like birds, looking very carefully at their wings, their feathers, their muscles, and then construct machines that emulate birds;

- study aerodynamics—how air flows above and below an airfoil, and how this provides lift—by using wind tunnels and varying the shapes of airfoils.

Both kinds of studies lead to insights, but of a different sort. The second strategy is perhaps the more general one: it seeks to discover the principles of flight that apply to anything, including birds.

Thinking is similar. Although we do want to understand human thinking and how it leads to intelligent behavior, this does not mean that there is no choice but to study humans. While there is a lot to be learned by studying the human brain (and other brains too), we can focus on the *thinking process* itself to determine general principles that will apply to brains and to anything else that needs to think.

This is what the philosopher Daniel Dennett calls taking a *design stance*. What we attempt to do is see what would be involved in designing something that is capable of flying or thinking. Instead of getting caught up in the details of how human brains do the job, we shift our attention to the job itself, and ask how it might get done *at all*. The hope is that in so doing, we get to see what is and is not essential in producing the phenomenon of interest.

Of course a design stance will not solve the entire puzzle. It may not tell us much about how the phenomenon can be the end result of an evolutionary process, for instance. And it is not applicable at all to phenomena that cannot be observed. If what we care about is not how birds are able to fly, but whether they have a certain internal sensation that they do not display, then a design stance will be of no help.

So in the end, what we will be concerned with here is *observable* intelligent behavior and how it is produced. We will argue

that thinking is needed, of course, but we will not be concerned with what the result of this thinking might *feel like* to the agent doing the thinking.

For many researchers, it is the subjective feeling of conscious thought (sometimes called *qualia*) that is the truly interesting and distinctive feature of the human mind. I take this to be a Big Puzzle issue. The sensation of consciousness is an interesting part of the puzzle, no doubt, but it is not the only one.

3 Knowledge and Behavior

In the mid-twentieth century, digital computers became much more widely available. Impressed as people were by their speed and accuracy, they were sometimes called “electronic brains.” Of course the makeup of computers (then and now) has very little in common with the makeup of biological brains. The more plausible question that came out of all this was whether computers were capable of doing some of the things that human brains were capable of doing.

For John McCarthy, what was most striking about the sort of commonsense behavior exhibited by people was how it was conditioned by *knowledge*. In deciding what to do in novel situations, people were able to use what they had been told and what they already knew about the world around them. What he had in mind were examples like the following:

- In planning how to behave, people had to use knowledge about their current situation, the options available to them, and what effect their actions would have.
- In making sense of a visual scene, people had to use what they knew about the appearances of objects to fill in details of the scene that were hidden from view.

- In using language, people had to use the knowledge they had about the topic of conversation and what the words meant.

A very wide range of intelligent behaviors seemed to be unified by the idea that they all depended in similar ways on a store of background knowledge. Once you know that lemons are yellow, for instance, this helps you do a number of things: paint one in a picture, spot one in a grocery store, make a list of yellow fruits, and others. McCarthy was the first to propose that to understand intelligent behavior we needed to study the knowledge that behavior depends on, and how this knowledge might be applied in deciding how to behave.

Before we get back to the Turing Test and to why McCarthy's proposal still matters in the next chapter, we need to slow down and take a closer look at the idea of knowledge itself and what it means to say that intelligent behavior depends on it.

Beyond stimulus and response

In understanding human behavior, there are many occasions where the simplest and best explanation for why we do what we do is in terms of stimulus and response. This is seen quite clearly in other animals, and we exhibit it too, including when we use language. We get our clothing caught on a nail unexpectedly and say "What the heck?" We bump into a person on a crowded train and say "Sorry." We see a child leaning precariously over a rail and say "Watch out!" We hit our thumb with a hammer and say a number of colorful things.

But it would be a Big Puzzle mistake to think that the rest of our language use was just more of the same. Consider how language is used in the following story:

You are with a group of people talking about the movie *2001: A Space Odyssey*. At one point your friend Tim says “The film is just gorgeous on Blu-ray.” You say “It’s hard to believe it first came out in 1968.”

So what do we have here? There is again a stimulus (Tim’s comment) and a response (your comment). But now it is much harder to account for how the stimulus elicits the response. What caused you to say the word “1968,” for instance?

There is clearly a *gap* here between stimulus and response, and the only plausible explanation about what fills this gap is some *thinking* on your part. You knew something, namely that the movie was released in that year, and you decided that saying so would advance the conversation in an interesting way. Maybe all you really wanted to do was to support Tim, or to connect with your friends, or to be seen as contributing your share of the conversation. That desire does not by itself explain why you said what you did. What accounts for it is applying what you knew about the movie.

Here is what the psychologist Sir Frederic Bartlett says about this process:

The important characteristics of [the] thinking process, as I am proposing to treat it, can now be stated: The process begins when evidence or information is available which is treated as possessing gaps, or as being incomplete. The gaps are then filled up, or that part of the information which is incomplete is completed. This is done by an extension or supplementation of the evidence, which remains in accord with the evidence (or claims to do so), but carries it further by utilizing other sources of information besides those which started the whole process going, and, in many instances, in addition to those which can be directly identified in the external surroundings. (*Thinking*, 1958, p. 75)

It is these “other sources of information” that make all the difference.

The back story: At some point you must have found out that *2001* was released in 1968. Maybe you saw the movie when it first came out, maybe you saw that date watching the movie later, maybe you read about the date somewhere, maybe you saw it on a poster, maybe you heard somebody talking about it. But however you found out, that stimulus caused a change in you. You might not have said anything at the time, but you started believing that *2001* was released in 1968. (We will have more to say about the difference between “knowing” and “believing” below.)

Later, when the movie is being discussed and Tim says his bit, thoughts occur to you very quickly. You recall watching the Blu-ray and how impressed you were, just like Tim. But you don’t just agree with Tim or say “Awesome!” You do not want to appear banal or uninformed, and you quickly form a thought you had never consciously entertained before, which is that *2001* looked better than other movies of the same period. But you don’t say that either. You happen to know something specific: the period was 1968. And this is what you bring to the table.

The moral of the story is this:

Knowledge, what you know and can bring to bear on what you are doing, is an essential component of human behavior.

This idea is obvious to many of us. But it does appear to be problematic for many others, mainly certain psychologists. They might say:

Why not simply say that there is one stimulus (reading about *2001*, say) that trains you, and a later stimulus (Tim’s comment) that causes you

to respond according to your training. Why all this mumbo-jumbo about knowledge and belief that we have no way of accounting for scientifically?

The trouble with the bare stimulus-response story is that it misses something very important. As the philosopher Zenon Pylyshyn has stressed in a related context, the actual stimulus you encountered (when reading about *2001*) is neither necessary nor sufficient to account for the response you produced.

First of all, note that what you saw written on a page is only one among very many visual stimuli that will do the trick. The information can be presented in a way that looks nothing at all like what you saw: different paper, different colors, different fonts, different words, even a different language. And there is really nothing visual about it either. You might have heard someone talking about *2001*. But this could have been quite different too: different volume, different pitch, different intonation, different words, different language.

Going further, maybe you did not even see or hear the word "1968." Maybe you heard that *2001* was released the year before Neil Armstrong walked on the moon. Or maybe all you were told is that the Best Picture that year went to the movie *Oliver!*, which might be enough if you knew enough about *Oliver!*. In terms of what the sensory stimulus needs to look or sound like, the range is incredibly wide.

Furthermore, the words "The movie *2001* was released in 1968" themselves might not do the trick. If they come right after the words "Here is something that is definitely false", you would not respond to Tim in the same way. Or maybe you see the words as an item in a long piece entitled "Common misconceptions about the movies of Stanley Kubrick." Or maybe somebody tells you that *2001* was released in 1968, so that you hear those

words, no more, no less, but you have good reason to believe that the person saying them is only guessing or is lying. Again your response to Tim would be different.

In the end, what matters about the stimulus, what causes it to do the job, is not what it looks or sounds like at all. There are many sorts of stimuli in context that will work, and many others that will not. What matters is whether or not they are *enough to get you to believe something*, namely, that *2001* was released in 1968.

Moreover, once you are in this state of belief, how you got there is no longer important. Your belief can affect any number of other actions you later do, linguistic ones and nonlinguistic ones, regardless of the original stimulus. If you decide to sort your Blu-ray movies by decade, or if you decide to have a moviefest of *Great Movies From the Sixties*, or if somebody asks you a direct question about Kubrick, all these actions will be affected by this one belief, however it came about.

Intelligent behavior is, in the words of Pylyshyn, *cognitively penetrable*: the decisions you make about what actions to perform is penetrated by what you believe, just as McCarthy first emphasized. If you come to believe that *2001* was released in 1972, for instance, the actions you choose will all change systematically.

Of course not all our actions are affected by what we believe in this way. Involuntary reflexes, for example, are not. Whether we raise our leg if someone taps us on the knee, or whether we blink as something approaches our eye, or even (to a certain extent) what we say when we hit our thumb with a hammer, these actions are selected very quickly and do not seem to depend on the beliefs we have. But sorting our movies by decade is one action that does.

So to summarize: As with all animals, we act in response to stimuli. In some cases, the mapping is direct: we sense something, and we react to it. But for very many others, the mapping is less direct: we sense something, but how we react depends on the beliefs we happen to have.

Knowledge vs. belief

What exactly do we mean by knowledge? The basic idea is simple: knowing something means taking the world to be one way and not another. But let us go over this a bit more carefully.

First, observe that when we say something like “John knows that ...,” we fill in the blank with a declarative sentence. So we might say that “John knows that *2001* was released in 1968” or that “John knows that Mary will come to the party.” This suggests, among other things, that knowledge is a relation between a knower, like John, and a proposition, that is, the idea expressed by a declarative sentence of English or some other language, like “Mary will come to the party.”

And what are propositions then? They are abstract entities, not unlike numbers, but ones that can be true or false, right or wrong. When we say that “John knows that *P*,” we can just as well say that “John knows that it is true that *P*.” Either way, to say that John knows something is to say that John has formed a judgment of some sort, and has come to realize that the world is one way and not another. In talking about this judgment, we use propositions to classify the two cases, for example, those where Mary will come to the party, and those where she won’t.

A similar story can be told about a sentence like “John hopes that Mary will come to the party.” or “John worries that Mary will come to the party.” The same proposition is involved, but

the relationship John has to it is different. Verbs like “knows,” “hopes,” “regrets,” “fears,” and “doubts” all denote what philosophers call *propositional attitudes*, relationships between agents and propositions. In each case, what matters about the proposition is its *truth condition*, what it takes for it to be true: if John hopes that Mary will come to the party, then John is hoping that the world is a certain way, as classified by the proposition.

Of course, there are sentences involving knowledge that do not explicitly mention propositions. When we say “John knows who Mary is taking to the party,” or “John knows how to get there,” we can at least imagine the implicit propositions: “John knows that Mary is taking ... to the party,” or “John knows that to get to the party, you go two blocks past Main Street, turn left ...,” and so on. On the other hand, when we say that John has a deep understanding of someone or something, as in “John knows Bill well,” or a skill, as in “John knows how to skate backwards,” it is not so clear that any useful proposition is involved. (While this type of “procedural” knowledge is undoubtedly useful, we will not have much more to say about it, except briefly in the section “Learning behavior” in chapter 5.)

A related notion that we will be very concerned with, however, is the concept of *belief*. The sentence “John believes that *P*” is clearly related to “John knows that *P*.” We use the former when we do not wish to claim that John’s judgment about the world is necessarily accurate or held for appropriate reasons. We also use it when the judgment John has made is not based on evidence, but is perhaps more a matter of faith. We also use it when we feel that John might not be completely convinced. In fact, we have a full range of propositional attitudes, expressed in English by sentences like “John is absolutely certain that *P*,”

“John is confident that P ,” “John is of the opinion that P ,” “John suspects that P ,” and so on, that differ only in the level of conviction they attribute to John.

Clearly how a person uses what he or she believes depends on how certain the person is about what is believed. This is called the *degree of belief*. For now, it will suit our purposes to not distinguish between knowledge and belief, or to worry about degrees of belief. What matters is that they all share the same idea: John taking the world to be one way and not another. (We will return to degrees of belief briefly in the section “Knowledge representation and reasoning” in chapter 9.)

The intentional stance

But this account of knowledge seems to suggest that it can apply only to entities that understand language the way we do. If P is a sentence of English, how can somebody or something know or believe that P is true if they cannot make sense of P ? So are humans unique in their use of knowledge?

No. A dog lunging for a flying Frisbee is not simply reacting to what it sees. What it sees is a Frisbee at position A ; where it lunges is at position B . It is not too much of a stretch to say that at the time of the jump, the dog anticipates that the Frisbee will be at position B . It could be wrong of course. Maybe there’s a serious crosswind, maybe the Frisbee is on a string, maybe the Frisbee is remote controlled. Nonetheless, the dog has something that goes beyond what it sees, smells, and hears that compels it to jump to a place where there is currently no Frisbee. Why not call this belief? The dog believes the Frisbee will be at position B . Maybe we want to say that this is very simple knowledge, or proto-knowledge (or only “knowledge” in quotation

marks), but it seems on a continuum with the sort of knowledge discussed above, even though the dog does not understand English.

Here's where a little philosophy will help clarify things. The philosopher Daniel Dennett suggests that when we look at complex systems (biological or otherwise), it's often useful to take what he calls an "intentional stance." By this he means describing the system *as if* it had beliefs, goals, desires, plans, intentions, and so on, the same way we talk about people. So we might say of the dog that it *believes* that the Frisbee will be at position *B* and *wants* to catch it when it arrives. We might say of a computer chess program that it *worries* that its board mobility is at risk and *wants* to get its knight out early to control the center of the board. We might also say of a thermostat that it *believes* that the room is warmer than it should be and *intends* to make it cooler.

In some cases (like the chess program), the intentional stance is useful and helps us decide how best to interact with the system. In other cases (like the thermostat), the intentional stance seems overblown and needlessly anthropomorphic, and the system is better dealt with in different terms. As for dogs, it is clearly useful to think about them in intentional terms, and we do it all the time, as in the following:

What does she want? Why is she scratching at the door? Oh I get it, she thinks her toy is in the other room and she's trying to get at it! Open the door and show her it's not there.

Dennett's main point is that these are stances. They are not inherently right or wrong. There are no real facts to the matter. A stance is just a way of looking at a complex system, and it may or may not be useful.

But it does raise an interesting question: is all the talk of knowledge and belief in people just a stance? Do we really want to say that people behave *as if* they had beliefs, desires, intentions, the same way we might talk about computer systems, dogs, and thermostats?

We are not ready to answer this question yet. But we will be taking it up in detail in chapters 8 and 9. There we will discuss a type of system (called a *knowledge-based system*) that is designed to work with its beliefs in a very explicit way. For these systems, beliefs are not just a useful way of talking; they are, in a clear sense, ingredients that *cause* the system to behave the way that it does.

This is more like the way we talk about gas in a car. A car is obviously designed and built to run on gas (or at least it was, prior to the hybrid and electric ones). So gas-talk is not just some sort of stance that we can take or leave. When it comes to talking about how the car is made to work, there is really no choice but to talk about how it uses gas.

Beliefs for knowledge-based systems will turn out to be the same.

Intelligent behavior

What is intelligent behavior anyway? If this is what we intend to study, don't we need a definition? We can't just say it's how humans behave, since so much of what we do is not very intelligent at all. We find it greatly entertaining to watch ourselves at our very dumbest, like the clueless daredevils on *America's Funniest Home Videos* or the squabbling couples on *The Jerry Springer Show*. There is so much buffoonery, recklessness, poor judgment, and sheer idiocy all around us, that behavior we would label as

truly intelligent appears to be more the exception than the norm! As Eric Idle sings in the Monty Python movie *The Meaning of Life*:

And pray that there's intelligent life somewhere out in space,
'Cause there's bugger all down here on Earth!

Somewhat of an exaggeration, maybe, but the point remains.

So what do we have in mind for those cases—rare as they may be—when a person is actually behaving in an intelligent way? We do not want to limit ourselves to things like playing chess, discussing Heidegger, or solving calculus problems. These “intellectual” activities are fine, of course, but we are looking for something much more mundane. Roughly speaking, we want to say that people are behaving intelligently when they are *making effective use of what they know to get what they want*. The psychologist Nicholas Humphrey puts it this way: “An animal displays intelligence when he modifies his behavior on the basis of valid inference from evidence.”

Imagine a person, Henry, thinking aloud as follows:

Where are my car keys? I need them. I know they're in my coat pocket or on the fridge. That's where I always leave them. But I just felt in my coat pocket, and there's nothing there.

What do we expect Henry to do next? The obvious answer is that he will think about his keys being on the fridge. This is what intelligent behavior is all about: using what you know to make decisions about what to do. Henry wants those keys, and we expect him to draw an inference to get to them.

But will he? Although this is the “right” thing for him to do, and Henry might well agree with us (for instance, watching a replay of his own behavior later), what he *actually does* could be quite different. He might get sidetracked and think about

something completely unrelated, like going to the washroom, or eating pizza, or what other people are thinking about him. He might make a joke. He might just collapse to his knees, sobbing “I can’t do this anymore!” Even if he really wants those keys, what he ends up doing involves many other factors, like these:

- his overall state: hunger, fatigue, motivation, distraction;
- his medical condition: injury, illness, poor eyesight, nausea;
- his psychological condition: anxiety, obsessions, pathological fears;
- his neurological condition: migraines, psychotropic drugs, dementia.

So if we are truly interested in the intelligent behavior that people like Henry produce, do we need to concern ourselves with factors like these too?

The answer is: it depends on what we want to study. As it turns out, we are not so interested in accounting for the behavior that people like Henry produce; it is too complicated and too fiddly. What we are interested in is a certain *idealization* of that behavior, that is, behavior that people like Henry would recognize as being intelligent, even if they admit that they sometimes fall short of achieving it.

Competence and performance

A similar distinction has come into play in the study of human language in what the linguist Noam Chomsky called the *competence/performance* distinction. The idea roughly is this: linguists want to understand human language, but by this they do not necessarily mean the actual linguistic output of human subjects, which is loaded with quirks of various sorts.

For example, consider a sentence that a person might write, like this one:

The hockey players celebrated there first win.

We might be tempted to ask: what is the grammatical role of the word “there” in this sentence? The answer, of course, is that it has no role; it’s a *mistake*, confusing the adverb “there” with the adjective “their.” From a linguistic point of view, this use of “there” is not really part of the language, so there is nothing grammatical to explain. And yet people do write this way! Similar considerations apply to the “umm,” “like,” and other hesitations that are part of everyday speech. We recognize that they are really not part of the language even though people say them all the time.

Linguists call the utterances speakers actually produce the *performance* of native speakers. However, they may prefer to study what the speakers of a language would recognize as genuine grammatical expressions of their language, which they call the *competence* of native speakers. While there is nothing wrong with wanting to study the performance of speakers, it is phenomenally complex. The advantage of looking at the competence of speakers is that we get to see certain abstractions and generalizations that might otherwise get lost in the vagaries of actual speech.

As a very simple example, consider the length of sentences. We recognize that whenever we have two declarative sentences, we can join them with an “and” to make a new one. But this recognition is only a competence judgment. It implies that there is no limit to the length of sentences, whereas there is quite clearly a limit to the length of a sentence that a person can produce in a lifetime.

When it comes to intelligent behavior, we can take a similar position: instead of trying to make sense of intelligent behavior in a way that accounts for the actual choices of real people (who may be under the influence of alcohol, or have short attention spans, flagging motivation, and so on), we focus on what people would recognize as proper, reasonable, intelligent behavior. This will give us something much more manageable to study.

4 Making It and Faking It

We have seen how intelligent behavior could depend on knowledge, at least in that conversation about the movie *2001*. But as part of something like a Turing Test, that one statement about the release date of the movie does not tell us much about the person who made it. For all we know, we might be talking to a mindless zombie who has nothing but canned responses. Mention a pastrami sandwich, and the zombie now says “It is hard to believe the pastrami sandwich first came out in 1968.” (It wouldn’t take long for this one-trick pony to fail the Turing Test!)

But we can also imagine a more sophisticated zombie, one that is able to say a much wider range of things about pastrami sandwiches by reciting sentences it finds on the Internet. These days, we are quite used to augmenting what we know using online texts and a search engine. To find out the population of Denver, we look for a sentence of the form “The population of Denver is” Of course, some of those sentences will say things like “The population of Denver is 40 percent Hispanic,” but we can work around that. If we want to find out if Denver has more people than Winnipeg, maybe no single online text has the answer we want, but we can get the desired answer by looking for a text for each city. (Winnipeg has slightly more.)

This raises some interesting questions from an AI point of view: What exactly is the difference between a system knowing something and a system that can find sentences about it in a large database of stored texts? Do we imagine that “big data” (in text form) can play the role in intelligent systems that we assumed was held by knowledge? Could it be that intelligent behavior is really no more than being able to carry on a certain *illusion* convincingly, such as being able to participate in a conversation without having to understand what is being discussed, bolstered perhaps by the ability to quote from canned text? If our goal is to understand intelligent behavior, we had better understand the difference between making it and faking it.

This is what we will attempt to do in this chapter.

Of course we are usually very much aware when we ourselves are faking it. We know what it feels like to know that in the animal kingdom, it is the ducks that quack, not the chickens, and it sure feels like more than just being able to recite some memorized sentences that happen to have the words “duck” and “quack” close to one another. (Consider: “The veterinarian they sent to look at my duck was a quack.”) But let us put that feeling aside. As we said in chapter 2, we are concerned with observable behavior, not with what introspection appears to tell us.

So the real question for this chapter is whether or not there is some form of observable behavior that would be difficult to exhibit without appropriate knowledge. The problem is that someone may be able to pass as knowledgeable without knowing very much. We have all heard stories about imposters playing the role of doctors, lawyers, and airline pilots without knowing medicine, law, or aircraft operation. There is the famous case of Frank Abagnale, depicted in the 2002 movie *Catch Me If You Can*, who did all three! Many of us have no doubt wondered

whether some of the “experts” on TV who appear to be talking knowledgeably about the global economy are really not just more of the same.

The trouble with conversations

What comes out of the Turing Test is an emphasis on externally observable behavior. Our concern in this chapter is that an informal conversation may be behavior that it is simply too easy to fake.

The trouble with the Imitation Game is that it places all the burden on *deception*. In the end, what it asks is not whether a computer program can carry on a conversation as well as a person, but whether a computer program can *fool* interrogators into thinking they are carrying on a conversation with a person. As every trickster knows, these two questions are not the same. A magician does not need to have ESP to wow an audience; he just needs to be able to fool the audience into thinking he has ESP, using all the tricks of the trade. (Or to be more precise, he has to be able to do the trick without giving the audience a better story to tell about how he does it.) And an informal conversation as suggested by Turing gives a trickster a lot of room to maneuver.

Consider someone impersonating a doctor on rounds in a hospital. It might be sufficient to glance knowingly at a patient's chart and say very little, in terms of conversation, but with doctorly confidence and authority. In the movie *Catch Me If You Can*, the Frank Abagnale character deflects attention away from himself by asking an intern “Do you concur?” a phrase he picked up watching *Dr. Kildare* on TV.

Consider next the case of ELIZA. This was a computer program written by Joseph Weizenbaum in the 1960s that was able, using very simple means, to fool some people into believing they were conversing with a psychiatrist. The deception works at least in part because again we are extremely forgiving in terms of what we will accept as legitimate conversation.

Here is a snippet of a conversation with Oliver Miller as the patient:

Patient: I only really feel happy when I'm writing.

ELIZA: How long have you been writing?

Patient: Always, but I've been writing seriously since I was eighteen.

ELIZA: What incident are you thinking of?

Patient: Um.

ELIZA: What does that suggest to you?

Patient: I don't think it's enough, to only be happy when you write.

ELIZA: Does that trouble you?

Patient: ... Um, because you can't write all the time. And then there's the real world, and you still have to deal with real people.

ELIZA: What are your feelings now?

Patient: I've been thinking a lot about that poem by Yeats: "The Man and the Echo." Do you know it?

ELIZA: Do you say you've been thinking a lot of that poem by Yeats: "The Man and the Echo" for some special reason?

Patient: Yes.

A Rogerian psychiatrist may say very little except to encourage a patient to keep on talking, but it may be enough, for a while. (Incidentally, the ELIZA program is small enough that a version is included just for fun in the Emacs text editor, and so ends up preinstalled on every Mac OS X system.)

Finally, consider the Loebner competition, a restricted version of the Turing Test held as a contest annually since 1990. In this case, we see much more balanced conversations taking place than with ELIZA. What is striking about transcripts of these

conversations is the fluidity of the responses from the test subjects: elaborate wordplay, puns, jokes, quotations, asides, emotional outbursts, points of order. Everything, it would seem, except clear and direct answers to questions. And how is an interrogator supposed to deal with the prevarication and determine whether or not there is any real comprehension behind the verbal acrobatics? More conversation. “I’d like to get back to what you said earlier.” Unsurprisingly, short conversations are usually quite inconclusive, and even with long ones, two interrogators looking at the same transcript may disagree on what the conversation tells us.

A case in point was the computer program called EUGENE GOOSTMAN, which was classified as human by a third of the interrogators in a 2014 contest. The program pretended to be a 13-year-old Ukrainian boy conversing in English, thereby accounting for both a poor command of the language and a certain playful evasiveness.

Here is a sample conversation with Scott Aaronson from MIT:

Scott: Which is bigger, a shoebox or Mount Everest?

EUGENE: I can’t make a choice right now. I should think it out later. And I forgot to ask you where you are from ...

Scott: How many legs does a camel have?

EUGENE: Something between 2 and 4. Maybe, three?;-))) By the way, I still don’t know your specialty—or, possibly, I’ve missed it?

Scott: How many legs does a millipede have?

EUGENE: Just two, but Chernobyl mutants may have them up to five. I know you are supposed to trick me.

Scott: No, I need to know that you’re not a chatbot. Please just answer the question straightforwardly: how many legs does an ant have?

EUGENE: Something between 2 and 4. Maybe, three?;-))) Oh, what a fruitful conversation ;-)

Scott: Do you understand why I’m asking such basic questions? Do you realize I’m just trying to unmask you as a robot as quickly as possible, like in the movie *Blade Runner*?

EUGENE: ... wait

Scott: Do you think your ability to fool unsophisticated judges indicates a flaw with the Turing Test itself, or merely with the way people have interpreted the test?

EUGENE: The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.

Scott: Do you think Alan Turing, brilliant though he was, had trouble imagining that the judges of his “imitation game” wouldn’t think to ask commonsense questions like the ones above—or that, if they did, they’d actually accept evasion or irrelevant banter as answers?

EUGENE: No, not really. I don’t think alan turing brilliant although this guy was had trouble imagining that the judges of his imitation game would not consider to Oooh. Anything else?

Scott: In your opinion, does your existence demonstrate any idea or principle that wasn’t demonstrated just as convincingly by ELIZA, Joseph Weizenbaum’s chatbot from the 1960s?

EUGENE: ELIZA was a break-thru. All the bots after it were nothing but weak parodies, claiming to have “revolutionary improvements.”

Scott: Hey, that’s the first sensible thing you’ve said!

In the end, the Turing Test has not really inspired AI researchers to develop better conversationalists; it has led only to better ways of fooling interrogators. We might have been hoping for first-rate intelligence to come out of it, but what we got was more like first-rate stage magic.

Answering questions

Given the lack of control in an informal conversation, it makes sense to shift to a more controlled setting, an artificial one, where intelligent behavior will be harder to fake.

Imagine a psychological experiment where a test subject is shown a series of yes/no questions, and can answer only by pushing a green button for *yes*, or a red button for *no*. Without

wanting to test for particular specialized knowledge (about law, medicine, aircraft operation, high school physics, whatever), we still want to design the test questions in such a way that subjects will be able to answer them using what they know. To probe for common sense, we want the questions to place the subject in a new, unfamiliar situation.

Consider, for example, a question like this:

Could a crocodile run a steeplechase?

Assume for the sake of argument that the subject in question knows what crocodiles and steeplechases are. (For those who do not know, a steeplechase is a horse race, similar to the usual ones, but where the horses must jump over a number of hedges on the racetrack. So it is like hurdles for horses.) Given this, our subject should be able to push the correct button—the red one—to answer the question. (We are assuming “ideal” test subjects here, competent and highly motivated.)

What is interesting about this question is that while much has been said and written about crocodiles and about steeplechases, nobody talks or writes about them *together*! So this represents a new situation, and the question cannot be answered by looking for a stock answer. Access to online text is no help. To put it differently, even if we were to assume that *everything* that had ever been written or spoken by anyone anywhere was available to be searched online, it would not contain an answer to the question. (This is not quite right. I have used this example before and so *my text* about crocodiles and steeplechases can be found online. But this is a quibble.)

Here is another example:

Should a team of baseball players be allowed to glue small wings onto their caps?

Again no one (other than me) has ever said or written anything about this topic. Again there is nothing that can be looked up, but again an answer should occur to a subject who knows baseball. (In this case, the answer will seem so obvious that the subject might worry that there is some sort of trick to the question. There is none.)

What is apparent in questions like these is what is called a *long tail* phenomenon, which will be discussed in detail in chapter 7. The idea roughly is that while most queries to online search engines center around a small number of very common topics (entertainment, sports, politics, cat videos, and so on), a significant portion is well away from the center (that is, out on a long tail) and involves topics that show up only very rarely. While the common queries can readily be answered using online texts, the very rare ones, far down on the tail, like the question about crocodiles or about gluing things onto baseball caps, will not appear anywhere in online texts.

And yet people are still quite clearly capable of answering them.

So have we found a form of intelligent behavior that requires using knowledge? No, not quite. Although subjects cannot look up the answers anywhere, they may still be answering the questions by other means.

The intent of the crocodile question was clear. It would be answered by *thinking* it through: a crocodile has short legs; the hedges in a steeplechase would be too tall for the crocodile to jump over; and hence *no*, a crocodile cannot run a steeplechase.

But there can be other ways of answering that do not require this level of understanding. One is to use the so-called

closed-world assumption. This assumption says (among other things) the following:

If no evidence can be found for the existence of something, one may assume that it does not exist.

This is how we deal with questions like “Are there any female world leaders who are over seven feet tall?” Is it not that we are ever told that there are none; rather, we surmise that there are none, believing that if there was one, we would have heard about her by now. For the crocodile question above, a subject might say “Since I have never heard of a crocodile being able to run a steeplechase [for example, since I cannot find any relevant texts about it], I conclude that it cannot.” End of story.

Note that this is somewhat of a cheap trick: it happens to get the answer right in this case, but for dubious reasons. It would produce the wrong answer for a question about gazelles, for example. Nonetheless, if all we care about is answering the crocodile question correctly, then this cheap trick does the trick.

Can we modify our psychology test somewhat so that cheap tricks like this will not be sufficient to produce the required behavior?

This unfortunately has no easy answer. The best we can do, perhaps, is to come up with a suite of questions carefully and then study how subjects might be able to answer them. Some promising approaches have been suggested by others, but let us turn to a specific proposal by Ernie Davis, Leora Morgenstern, and me.

Winograd schemas

As before, we are considering an imagined psychology experiment where a subject must answer questions. Again, there will be just two possible answers that a subject can choose using two buttons. The questions will always be of the same form, best illustrated with an example:

Joan made sure to thank Susan for all the help she had given.
Who had given the help?

- Joan
- Susan

We call questions like these *Winograd schema questions*, characterized as follows:

1. Two parties are mentioned (both are males, females, objects, or groups). In the example above, it is two females, Joan and Susan.
2. A pronoun is used to refer to one of them ("he," "she," "it," or "they," according to the parties). In the example above with females, the pronoun is "she."
3. The question is always the same: what is the referent of the pronoun? The question above is: who is the "she" who had given the help?
4. Behind the scenes, there are two *special words* for the schema. There is a slot in the schema that can be filled by either word. The correct answer depends on which special word is chosen. In the above, the special word used is "given," and the other special word (which does not appear) is "received."

So each Winograd schema actually generates two very similar questions:

Joan made sure to thank Susan for all the help she had **given**.
Who had **given** the help?

- Joan
- Susan ✓

And

Joan made sure to thank Susan for all the help she had **received**.
Who had **received** the help?

- Joan ✓
- Susan

It is this one-word difference between the two questions that helps guard against using the cheapest of tricks on them.

To get a better sense for what is involved in the test, here are some additional examples. The first is one that is suitable even for young children:

The trophy would not fit in the brown suitcase because it was too small. What was too small?

- the trophy
- the brown suitcase

In this case, the special word used is “small” and the other word is “big.” Here is the original example, which is due to Terry Winograd, for whom the schema is named:

The town councilors refused to give the angry demonstrators a permit because they feared violence. Who feared violence?

- the town councilors
- the angry demonstrators

Here the special word is “feared” and the alternative word is “advocated.”

With a bit of care, it is possible to come up with Winograd schema questions that exercise different kinds of expertise. Here is an example about certain materials:

The large ball crashed right through the table because it was made of Styrofoam. What was made of Styrofoam?

- the large ball
- the table

The special word is “Styrofoam” and the alternative is “steel.” This one tests for problem-solving skill:

The sack of potatoes had been placed below the bag of flour, so it had to be moved first. What had to be moved first?

- the sack of potatoes
- the bag of flour

The special word is “below” and the alternative is “above.” This example tests for an ability to visualize:

Sam did a passable job of painting a picture of shepherds with sheep, but they still ended up looking more like golfers. What looked like golfers?

- the shepherds
- the sheep

The special word used is “golfers” and the other is “dogs.”

Sentences that appear in Winograd schemas are typically constructed very carefully for the purpose, although it is possible to find naturally-occurring examples that are related. Consider this exchange from the 1980 comedy movie *Airplane!*:

Elaine: You got a letter from headquarters this morning.

Ted: What is it?

Elaine: It’s a big building where generals meet, but that’s not important.

Note the two nouns “letter” and “headquarters,” the pronoun “it,” and the joke of somehow managing to get the referent wrong.

Of course not just any question that is superficially of the right form will do the job here. It is possible to construct questions that are “too easy,” like this one:

The racecar easily passed the school bus because it was going so fast. What was going so fast?

- the racecar
- the school bus (Special=fast; other=slow)

The problem is that this question can be answered using the following trick: ignore the first sentence completely, and check which two words co-occur more frequently in online texts (according to Google, say): “racecar” with “fast” or “school bus” with “fast.” A simpler version of the same phenomenon can be seen in this example:

The women stopped taking the pills because they were carcinogenic. Which individuals were carcinogenic?

- the women
- the pills (Special=carcinogenic; other=pregnant)

Questions can also be “too hard,” like this one:

Frank was jealous when Bill said that he was the winner of the competition. Who was the winner?

- Frank
- Bill (Special=jealous; other=happy)

The problem is that this question is too ambiguous when the “happy” variant is used. Frank could plausibly be happy because he is the winner or because Bill is. (It is possible to correct for these extreme cases, but let us not worry about that here.)

A Winograd schema question need not be appropriate for every possible test subject. The trophy/suitcase example above might be suitable for children, for example, but the town councillor/demonstrator one likely would not be. The Styrofoam/

steel question is not suitable for subjects who have no idea what Styrofoam is. In general, a Winograd schema question would need to be carefully vetted before it can be used in a test. At the very least, we would want to ensure that a test subject knows the meaning of all the words that will appear in the question.

Given these considerations, it is now possible to formulate an alternative to the Turing Test. A suite of pretested Winograd schemas is first hidden in a library. A *Winograd Schema Test* involves asking a number of these questions, choosing at random one of the two special words, with a strong penalty for wrong answers (to preclude guessing). The test can be administered and graded in an automated way. No expert judges are needed.

To summarize: With respect to the Turing Test, we agree with Turing that when it comes to intelligence (or thinking or understanding), the substantive question is whether a certain observable behavior can be achieved by a computer program. But a free-form conversation as advocated by Turing may not be the best vehicle for a formal test, as it allows a cagey subject to hide behind a smokescreen of playfulness, verbal tricks, and canned responses. Our position is that an alternative test based on Winograd schema questions is less subject to abuse, though clearly much less demanding intellectually than engaging in a cooperative conversation (about sonnets, for example, as we saw in the discussion of the Turing Test in chapter 1).

The lesson

Returning to the main point of this chapter, the claim here is simple: with no special training, normally abled English-speaking adults will have no trouble answering Winograd

schema questions like those above, getting close to all of them correct.

Within this artificial setting, *this* is the sort of intelligent behavior we want to concentrate on. *This is the data that our account of intelligent behavior needs to explain!*

As scientists, we can think of this as a natural phenomenon to study, like gravity or photosynthesis. We ask: *how do we explain how people are able to do this?* Clearly, people do not merely recall something they heard or read. Like the question about crocodiles, the answers do not appear in texts anywhere. Can the behavior be faked using some sort of cheap trick? Perhaps, but the one-word difference between the two versions of each question now makes this much less likely. (See the section “Winograd schemas again” in chapter 7 for more on this.)

Consider once again the Styrofoam/steel question above. We might contemplate using special words other than “Styrofoam” and “steel” in the question. For “granite,” the answer would be “the large ball”; for “balsa wood,” it would be “the table”; and so on. But suppose we were to use a completely *unknown* word in the question:

The large ball crashed right through the table because it was made of kappanium. What was made of kappanium?

- the large ball
- the table

In this case, there is no “correct” answer: subjects should not really favor one answer much over the other. But let us now further suppose that we had told the subjects in advance some facts about this unknown kappanium substance:

1. It is a trademarked product of Dow Chemical.
2. It is usually white, but there are green and blue varieties.
3. It is 98 percent air, making it lightweight and buoyant.
4. It was first discovered by a Swedish inventor, Carl Georg Munters.

We can ask, on learning any of these facts, at what point would the subjects stop guessing the answer? It should be clear that only one of these facts really matters, the third one. But more generally, people get the right answer for Styrofoam precisely because they already know something like the third fact above. (The four facts above were lifted from the Wikipedia page for *Styrofoam*.) This background knowledge is critical; without it, the behavior is quite different.

So this takes us back to the 2001 story of the previous chapter. The lesson here is the same: in order to understand how people are able to produce certain forms of intelligent behavior, in this case, pushing the right buttons on Winograd schema questions, we need to concentrate on the background knowledge they are able to bring to bear on the task.

The return of GOF AI

As noted in chapter 1, much of current AI research has moved away from the early vision of John McCarthy with its emphasis on knowledge. The GOF AI advocated by McCarthy is thought by some to have run its course. What is needed now, it is argued, is a fresh approach that takes more seriously new insights from neuroscience, statistics, economics, developmental psychology, and elsewhere.

While those disciplines unquestionably have insights to offer, and while a fresh computational approach to them may be quite productive, what McCarthy had in mind was something considerably more radical. Rather than a computational version of neuroscience or statistics or whatever, he proposed a discipline with an entirely new subject matter, one that would study the *application of knowledge* itself—thinking, in other words—as a computational process.

It is very significant that the critics of GOFAI do not try to account for the same intelligent behavior but by other means. It is not as if anyone claims that computational economics (say) would be better suited to explain how people are able to do things like answering Winograd schema questions.

Instead, the move away from GOFAI is more like a shift in subject matter. Instead of focusing on the ability of people to do things like answering those questions, the researchers turn their attention to other forms of behavior, and, in particular, to behavior that may be less dependent on background knowledge. (They may go so far as to argue—using some sort of evolutionary rationale, perhaps—that we should not even try to study behavior that depends on knowledge until we better understand behavior that does not.)

So for example, researchers in that part of machine learning we have been calling AML might focus on our ability to recognize hand-written digits. Our ability to do this—to tell the difference between a written 8 and a 3, for instance—seems to depend much less on background knowledge and much more on our having been exposed to a large number of sample digits over a wide range of styles. Researchers in AML concentrate on showing how the necessary patterns and features can be learned from these samples in an automated way.

There is nothing wrong with any of this, of course. It would be a Big Puzzle mistake to assume that *all* human-level intelligent behavior must be the result of applying background knowledge. What we see with the Winograd schema questions, however, perhaps as clearly as anywhere, is that *some* of it is.

And the question remains: how are people able to produce that behavior? Could it not be the result of something like AML, but perhaps in a more elaborate form over a different kind of data? To answer this question, we need to look more deeply into what is involved with learning and acquiring knowledge.

5 Learning with and without Experience

Unlike thirty years ago, when I began my career in AI, when people hear about work in AI these days, most likely they will hear about applications in the area of machine learning, what we have been calling AML. Researchers in AML have been extremely successful in producing a wide range of useful applications, everything from recognizing handwritten digits, to following faces in a crowd, to walking on rough terrain, to answering questions on *Jeopardy*, to driving cars autonomously. All of these lean heavily on the statistical techniques developed in AML.

If we were to listen to researchers in this area talk about what they study, we might get the impression that what learning means is extracting patterns and features from data (or from the environment) without anything like lessons from an instructor or an instruction book. Clearly there are many things that we learn in this way. It is how we come to learn apples from oranges, aunts from uncles, cats from dogs. It is not as if we are *told* how to do it: “Look. See the way cats have these long thick whiskers? Watch for them because dogs don’t have them.” We figure this out for ourselves, just like in AML. This is also how we come to learn a first language, how to say words and sentences. In fact,

we learn so much on our own without much help from written or verbal instructions, that it is easy to forget how much we do *not* learn this way.

This is, in other words, a Big Puzzle issue.

Learning words

Let us start by considering how we acquire vocabulary.

Take the adjective “hungry” for starters. Assuming we have English-speaking parents, we eventually learn what this word means. But when? Nobody ever explains the meaning to us. A child simply hears the word often enough at times when it is feeling hungry that it gradually comes to associate the word with the feeling of wanting food.

In a sense, it is a good thing that we do not need to *define* the word “hungry,” since language is somewhat at a loss for doing so. What does it feel like to be hungry? We might say: it is not pleasant, but it is not painful either, at least at first. It is a mild alarm reminding you to eat. But how is that feeling similar to and different from other such feelings, like thirst or vertigo?

Interestingly, just because we have a bodily alarm for something does not mean that there is an adjective to learn that describes it. In French, for example, there is no word for “hungry.” The closest is the adjective “affamé,” but this is more like “famished,” a much stronger form of hunger. In French, we say “J’ai faim,” literally, “I have hunger” or maybe “I have the hunger feeling.” So it is unclear which body alarms will end up having words for them. For example, there is a feeling that every human experiences at least as often as hunger, and that parents of young children fret about every day, but surprisingly for

which there is no adjective in English, and that is the feeling that goes with wanting to urinate.

At any rate, we learn the meaning of “hungry” and its ilk not through explanations but by repeated exposure to the word in appropriate contexts.

So is this what learning a vocabulary is like? Keeping the Big Puzzle issue in mind, let us look a bit further.

Consider the adjective “incarnate.” We typically learn what this word means in a way that is quite different. The first time we hear it (or see it) we are usually much older. We might even remember a time when we did not know what the word meant, and a later time when we did. The meaning might need to be explained to us: we are told it applies to spirits or disembodied figures who, for some reason, have taken on a physical body. We might have had to look up the word in a dictionary. The word has pieces: “in”+“carno” that mean literally “into”+“flesh.” So something incarnate is something that has been transformed into flesh. (At a certain age, we might need to look up what “flesh” means too.)

Here is what we do not expect: we do not expect to learn the meaning by hearing the word repeatedly in the right contexts. Even religiously inclined people, who are the ones most likely to use it, might say the word in prayers and rituals without even knowing what it means, the way Latin was sometimes used in Catholic liturgy.

If we ever do learn what the word means it is because of *language*: we hear or read an explanation about its meaning. And we do not need repeated trials; we might need to use a dictionary only once to be able to use the word meaningfully. So this is a very different way of acquiring vocabulary.

(Incidentally, I am not claiming that the all words we acquire fit neatly into the “hungry” or “incarnate” category. Consider the word “exuberant.” For many people, this will be learned via language in the second way, maybe by looking it up. But others will learn it by repeated exposure: they will see or hear the word often enough in context that they will figure out the meaning of the word. For others, it might be a combination of the two. I have a rough idea of what “hegemony” means, for example, but I would need to look it up to use it confidently.)

It might seem on the surface that learning words via language (for words like “incarnate”) is somehow less fundamental, less significant perhaps, than the more direct form of learning via experience (for words like “hungry”). Indeed, to acquire vocabulary via language, we must already have a language skill developed enough to be able to understand the explanation. So the more basic question is perhaps: how do we come to have that original language in the first place?

How we acquire a vocabulary by hearing words in context is undeniably a fascinating subject. It is a process that begins before we can speak, and continues to some extent even when we are adults. Every family has stories they love to tell about how their children picked up words and used them incorrectly at first, but endearingly. (I remember my daughter pouting and arguing that our not allowing her to do something was “too fair.”) And as the linguist Noam Chomsky has emphasized, it remains a great mystery how children can learn a language at all, given the small amount of data they are exposed to.

But equally fascinating is how we use language to learn more language, what might be called the *bootstrapping language problem*: how do we learn enough language to give us the capability to then lean on language to get the rest, including all the

technical language behind mathematics, science, and technology. (More on this in the next chapter.)

Learning facts

Let us now turn to how we acquire general knowledge about the world.

Consider this simple fact: lemons are yellow. Many fruits have distinctive colors (with interesting variations): blueberries, blackberries, and oranges are obviously blue, black, and orange respectively; raspberries are red; peas are green; plums are purple; apples are red, green, or gold. But lemons, like bananas, are yellow. They may begin their lives green, and finish their lives dark brown, but a typical, mature, freshly picked, healthy lemon is yellow.

Now ask yourself: how did I learn this? It is unlikely that you learned it through language. The information is written in books, of course, but it is likely that you had seen lemons (or color pictures of them) well before you ever read any text describing them. In fact, in sentences about lemons and their color, it is much more likely that a speaker or writer would assume you already knew the color of lemons and use that to describe something else. You might read something like “She wore a dress that was a bright lemon yellow.”

This little bit of knowledge we have about the color of lemons is an example of something we learn not by language but by direct experience.

Now consider this fact: bears hibernate. Some animals, like geese, whales, and monarch butterflies, migrate south for the winter. Some animals, like wolves, deer, and most fish, stay where they are and make do despite the harsh climate. Some

animals, like blue jays, migrate south some of the time. But bears, by and large, find themselves a den somewhere in the late fall, and hibernate (or go into some sort of torpid state) for the duration of the winter.

Ask yourself: how did I learn this? One thing is certain: you did not learn it by observing a number of bears and noticing a recurring pattern. Unless you are a bear specialist, the number of live bears you have actually seen is probably small, and almost certainly none of them were hibernating. You may have seen many pictures and movies of bears, but they would not cause you to believe that bears hibernate, since a picture of a hibernating bear is no different from a sleeping one. You may have seen nature documentaries showing them sleeping in a cave somewhere in the winter, and perhaps emerging groggy from the cave in the early spring. But the visuals would show a bear entering a cave in a fall scene, and perhaps emerging later in a spring scene, with not much to show in between. The information that the bear stayed mostly asleep for months at a time without going out to look for food would be carried by the narration. If you did not see a nature documentary, then all you really got was the narration: somebody told you or you read it somewhere.

This little bit of knowledge about the animal kingdom is an example of something that we learn entirely through language.

So here we have two different bits of generic knowledge: one, that lemons are yellow, and one, that bears hibernate, and the two bits are almost always learned in quite different ways.

Learning behavior

Finally, let us turn our attention to how we learn new behaviors.

First, consider how you learned to ride a bicycle. Here is what did not happen: you did not get out an instruction manual, study it carefully, and then set out on a bicycle for the first time. Instead, somebody probably taught you and at an early age. You may have started with training wheels, or with the teacher holding your bike upright. You wobbled a bit as you first set off on your own, and you likely fell a few times. If your teacher told you much of anything, it was to not go too fast (where crashes would be painful) or too slow (where the bike would be unsteady). You were never told how much you should shift your weight or turn your handlebars to compensate for the bike falling in one direction or the other. You were never told if you should lean in or lean out when making a turn. You picked this up by trial and error. But you did get better at it, and, like a bird learning to fly, you mastered the skill, and it became second nature. After a while, you just stopped thinking about it. Now, you get on a bike and go, and your body somehow knows what to do, while you are free to think about something else.

Is this what it means to learn a behavior? Let us again look a bit further.

Consider now how you might have learned to take care of a pet canary. It is possible that you lived on a canary farm and that you apprenticed with canary experts. Or maybe your parents had a canary and you learned what to do by observing them. But just as likely, you were given a pet canary out of the blue, or you took one home from a pet shop, without much actual canary experience to fall back on. Here is what did not happen: you did not learn how to take care of the bird by trial and error. For example, you did not try a variety of possible canary foods, getting more successful at not killing the canary each time. Taking care of a canary cannot really be learned this

way, unless you are prepared to go through a number of birds in the process.

But it is not rocket science either. You can easily learn what you need to do. It is certainly as easy as learning to ride a bicycle. A short conversation with a pet shop owner, a small pamphlet, or even a simple online search will do the trick. It is worth noting, however, that, as with rocket science, there is a need for some *basic facts* that are hard to convey without language. Things like this:

Fresh greens are a must if you want a happy and healthy canary.

A canary cannot live for a 24-hour period without water.

Canaries are very sensitive to toxins of all sorts.

A canary can overheat if the cage is exposed to direct sunlight.

A person could conceivably learn these by trial and error over a number of test canaries, or by pouring over massive amounts of canary data. But life is short and we prefer to rely on the experiences of other canary owners distilled in a pamphlet.

Beyond experience

In summarizing how we come to learn what we know about the world around us, it appears that at least two separate mechanisms are at work. The first involves learning through experience. The experiential data might need to be seen or heard repeatedly, and there may be considerable trial and error involved. The second form of learning involves learning through language. In this case, the data might need to be seen or heard just once to do its job. Because of this, this second form seems to have very little to do with the statistical techniques used in AML.

(Although repetition and memorization are still issues in learning through language, the statistical properties of the repeated instances are of less concern.)

There are different terms used in English to describe varieties of human learning, but none of them fits the above distinction perfectly. We talk of learning via education, via instruction, via training, via drill. The emphasis in each case is somewhat different: education suggests a general broadening of the mind; instruction emphasizes education in a particular subject, as in a school course or in a technical manual; training emphasizes education and practice involving a particular skill, such as how to control a kayak or to perform an autopsy; drill is training that emphasizes the repetition aspect, such as arithmetic practice or weight lifting. In each case there will be both language data and experiential data to contend with, but the balance changes depending on the sort of learning involved: the relative importance of language decreases as we move from general education, to instruction, to training, to drill. (In chapter 7 we use the term “training” to refer to learning derived from repeated experiences.)

The fact that we learn in two very different ways raises a number of interesting questions. What do we do when there is a conflict between the two, for instance? We learn from experience that the sun rises in the East, but then we learn through language that the sun does not rise at all. We learn through language of the existence of things that direct experience tells us do not exist. (We might be told that we do not experience them because they are too fast or too slow, too big or too small, or somehow exist on a different plane.) In fact, we learn through language about things that we know are totally fictitious, like the names of Santa Claus’s reindeers.

One interesting question in all this is why we have evolved to be able to learn through language at all, something that other animals do not appear to do. Why is direct experience not enough? What extra advantage does language give us when it comes to learning?

This topic will be explored more fully in the next chapter, but the idea is well summarized in a quote by S. I. Hayakawa: "It is not true that we have only one life to live; if we can read, we can live as many more lives and as many kinds of lives as we wish." Learning through language allows us in one single lifetime to pick up on the experiences of many lives before us. We get to learn what other people have learned without having to go through what they went through.

So to think of human language as just a means of *communication* is to grossly underestimate what it gives us. There is communication, no doubt, but communication can be achieved in many ways, for example, by pointing at things, or by making loud noises. Most animals communicate one way or another. But with human language we get much more. There is obviously a very wide spread between a pamphlet about pet canaries and Shakespeare's *Macbeth*, but they have something very important in common: they allow us to learn something we care about without having to go through life experiences that might be painful, impractical, or even impossible.

In some cases, this is quite explicit. A recipe for *The Best Tiramisu Ever* says in effect "I've tried the variants; don't waste your time." A book about mining in the Andes says "Here is what I saw when I was there; you might save yourself the trip." A book of fiction might be saying something more like "Here is what I think it must have felt like for workers in England during the Industrial Revolution."

In summing his scientific work, Isaac Newton said the following: "If I have seen further, it is by standing on the shoulders of giants." The metaphor is not simply that he got to see further by going higher; he could have gone higher by climbing a tree. It's more like this: by reading what the giants had written, he had the enormous advantage of being able to start where they left off, without having to go through all they went through. This is how most of us learn the meaning of many words, the winter habits of bears, and how to take care of canaries.

6 Book Smarts and Street Smarts

As we have noted, human language is used in two quite different ways. Like other animals, we use language for immediate communication; that is, we make sounds for others to hear in the moment or soon thereafter (when they pick up their voicemail, say). We do a similar thing with our written messages, both on paper and electronically.

But we also use language in a second, more detached way: we broadcast sport scores and courtroom dramas, we write poetry and instruction manuals. Here, we have only the vaguest of ideas about who will be hearing or seeing the result, and when. Maybe the person who will read the poetry has not even been born yet. This is a very different form of language use that is unique to our species. Its cumulative effect over many generations is what we call (oral or written) culture.

The impact of language

Ask yourself this: what is it about humans that has made our impact on the planet so great, that has made the lives of animals everywhere depend on what we decide to do? You might say something like “nuclear power” or “pollution” or “genetic

engineering” or maybe even “plastics” (the big life-changing secret Benjamin was told of in the movie *The Graduate*). But of course there is more to it than any one of these. We should just step back and say “advanced technology.”

But other animals have technology too. Crows and chimps use sticks to reach for things; otters use rocks to break open clams. Why do humans alone among animals have this more *advanced* technology? It is clear that we couldn’t have it without science. And we couldn’t have the science without mathematics. And we couldn’t have the science or the mathematics without human language. More specifically, we couldn’t have them without the sort of language use that goes beyond immediate communication.

Put it this way: if our use of language had remained as limited as the sort of communication we see in other animals, we would never have been able to *accumulate* enough science and mathematics over the generations to develop those advanced technologies. If all we could ever do is announce our presence, or signal something we had found, or point to something we wanted, our impact on the world would have been much smaller.

To take one example, we could not have our modern *cities* without human language. The large-scale coordination necessary to allow urban transportation and communication, the delivery of food, water, and electricity, the removal of garbage and sewage, the responses to fires and other calamities, and so on, would be inconceivable if language could ever be used only for immediate communication.

(It is interesting to speculate about what kind of creature we would have to be to live in large groups with only limited language. Ants, for example, do quite well in extremely large colonies with just a smidgen of communication. The trick, it appears,

is a strong genetic programming that restricts variability in behavior and compels individuals to take care of colony business. Humans have been living in cities for only a few thousand years, roughly since the discovery of agriculture. Perhaps after a few million years, we too will evolve into creatures where doing what needs to be done in a city will feel as incontrovertible as breathing. We would no longer need written laws and bylaws, rules and regulations, of course, not to mention parents, teachers, and ministers exhorting us to obey those laws, nor a police force to seek out those who decide to opt out and do things their own way.)

Think for a moment of some of the things you can learn by apprenticeship, using language only for immediate communication: foraging for food, milking a cow, plowing a field. Now turn this around. Consider the kinds of things you *cannot* learn without language texts that go well beyond immediate communication: linear algebra, electrical engineering, urban planning.

The science, the mathematics, the engineering, all of these are learned mostly in school, listening to lectures, reading texts. Of course, direct experience and practice are necessary to master the skills needed to be proficient in these disciplines. But practice will never make perfect without background knowledge to begin with.

So learning through text, that is, learning by reading or by being told, is not only a feature that is unique to humans, it is what has enabled us—for better or for worse—to dominate the rest of the animal kingdom.

Book smarts

Given the importance of learning through text in our own personal lives and in our culture, it is perhaps surprising how utterly

dismissive we tend to be of it. It is sometimes derided as being merely “book knowledge,” and having it is being “book smart.” In contrast, knowledge acquired through direct experience and apprenticeship is called “street knowledge,” and having it is being “street smart.” Then, as Scott Berkun succinctly puts it: “Street smarts kicks book smarts ass.” (I’m not sure I agree with this punctuation and grammar, but that’s just book knowledge talking!)

Here is what he says in a blog:

Book smarts, as I’ve framed it, means someone who is good at following the rules. These are people who get straight A’s, sit in the front, and perhaps enjoy crossword puzzles. They like things that have singular right answers. They like to believe the volume, and precision, of their knowledge can somehow compensate for their lack of experience applying it in the real world. Thinking about things has value, but imagining how you will handle a tough situation is a world away from actually being in one (As Tyler Durden says in *Fight Club*—“How much can you know about yourself if you’ve never been in a fight?”) (From <http://scottberkun.com/2010/book-smarts-vs-street-smarts>)

Although we may not agree with everything here, most people accept the basic thrust of the argument. In fact, we might be hard pressed to find anyone to argue seriously against it, despite what was said above regarding science and cities. Concerning cities, we might even be tempted to think along these lines:

If it is true that we cannot live in big cities without considerable book knowledge, then so much the worse for living in big cities!

Indeed, it was not so long ago that people lived in much smaller communities and relied much less on the urban technology that depends on book knowledge. It is certainly quite possible for small groups to live with a minimum of book knowledge, where most of what is learned is through apprenticeship. (Even

in large cities, individuals do not need to know the technology their lives depend on, so long as there are enough other individuals there to know it for them.)

But what about the opposite? Is it possible to live mostly by book knowledge, with a minimum of direct experience in the world? This seems much less likely. There are some possible counterexamples, but they are rare enough to be noteworthy.

Helen Keller

Consider the famous case of Helen Keller (1880–1968). At the age of eighteen months, she developed a disease that left her blind and deaf. At that point, she had acquired only a rudimentary baby talk (by direct experience), but then was cut off from learning English the usual way.

However, Helen Keller's family was able to hire a personal tutor for her when she was six, Anne Sullivan. Anne began teaching Helen words for the things around her by drawing the words, one letter at a time, on her hand. She acquired about twenty-five nouns and four verbs this way.

Then one day, something happened, wonderfully depicted in the movie *The Miracle Worker*. Here it is in the words of Anne Sullivan:

This morning, while she [Helen] was washing, she wanted to know the name for "water." ... I spelled "w-a-t-e-r" ... We went out to the pump-house, and I made Helen hold her mug under the spout while I pumped. As the cold water gushed forth, filling the mug, I spelled "w-a-t-e-r" in Helen's free hand. The word coming so close upon the sensation of cold water rushing over her hand seemed to startle her. She dropped the mug and stood as one transfixed. A new light came into her face. She spelled "water" several times. Then she dropped on the ground and asked for its name and pointed to the pump and the

trellis, and suddenly turning round she asked for my name. I spelled "Teacher." Just then the nurse brought Helen's little sister into the pump-house, and Helen spelled "baby" and pointed to the nurse. All the way back to the house she was highly excited, and learned the name of every object she touched, so that in a few hours she had added thirty new words to her vocabulary. ... Helen got up this morning like a radiant fairy. She has flitted from object to object, asking the name of everything and kissing me for very gladness. Last night when I got in bed, she stole into my arms of her own accord and kissed me for the first time, and I thought my heart would burst, so full was it of joy. (From a letter dated April 5, 1887)

It is not clear how to describe what happened at this moment in Keller's life, but one way is to say that she realized for the first time that there were *names* for things. Although she had been naming things for a while, she now realized that in addition to all the physical things, qualities, and actions around her—things she could feel—there were also some abstract things called *words* that could be used to name the others.

This is not unlike what we see in human infants. At a very early stage, we learn words like *water*, *hungry*, and *walk*, say. A child will use words to ask questions such as what her mother is chewing on, what her father is doing, where her favorite doll is. The answers will be things, qualities, and actions in the world. But at some point in a child's development, something new happens. In addition to those things in the world, there will be some new things to talk about: the words themselves. A child will now use words to ask questions about words: what the name of a certain object is, what a certain word means.

(Arguably, this is the crucial step that allows language to be learned through language rather than just through direct experience. It is not the use of symbols—in this case, words—to talk about things in the world; it is the fact that the symbols

themselves can be things to talk about. The reason this is so significant is that it separates symbols from what is observed or desired. The symbols get a life of their own and can be used for more than merely referring to things in the world. In philosophical terminology: the symbols can be *mentioned*, not merely *used*. As we will see in chapter 8, this is just what is needed for symbol processing.)

Having gone through this miracle, Helen Keller then made astounding progress, learning braille and some languages other than English, writing books, and eventually earning a university degree. (She was the first deaf-blind person to do so.) Because of her condition, the knowledge she acquired was mostly as a result of reading and conversing with others, what we are calling book knowledge. In terms of book smarts, she was a phenomenal achiever.

So is Helen Keller a counterexample to the claim that book smarts with only a modicum of street smarts is never enough? Here is what John McRone says:

But the reality was that Helen was so cut off from the world that she found it hard to tell the difference between her memories and her imagination. She had learnt to juggle words, but it is questionable how much understanding lay behind the fine sentiments that so pleased her audiences. ... Helen appeared to end up almost top-heavy with the software of culture. Blind and deaf, her brain was starved of the normal traffic of sensations, images and memories. Yet through language, she could furnish these spare surroundings with all the varied richness of human culture. In the end, the combination may have been unbalanced; where most saw a heroic triumph against the odds, others saw too heavy a weight of ideas sitting uncomfortably on an emaciated awareness. (From a blog at dichotomistic.com)

I do not agree at all with this. (And I find it condescending.) But it does say something quite interesting about the point I believe

Turing was trying to make. It suggests that even someone like Keller who can write books and earn university degrees might be considered by some to be just juggling words without a real understanding of what is being said. This “juggling words” critique of Keller is precisely what is said of ELIZA and other AI programs (as first noted by William Rapaport). But do these critics think that Helen Keller would have failed a Winograd Schema Test? Or would they be forced to claim that it was possible to pass the test using cheap tricks without understanding? (And if so, which ones?)

We can all be impressed by what Helen Keller was able to accomplish despite her enormous handicap. But evidently many people still harbor doubts regarding how far even a human mind can go without a healthy dose of street smarts.

What is perhaps even more interesting in all this, however, is that there is rarely a corresponding skepticism about how far we can go without a healthy dose of book smarts. For whatever reason, we do tend to downplay the significance of those parts of our mental life that are uniquely human. When we see something remarkable in ourselves (such as our language, planning, tool-building, culture), we try hard to find something like it in other animals too. We make much less of a fuss these days about the *differences* between us and the other animals. And those differences to a very large extent are grounded in our book learning, how we use language not for immediate communication, but to enhance and further extend our language, planning, tool-building, and culture.

Street smarts by the book

As for book and street smarts, let us reconsider the example about hibernating bears and yellow lemons. We observed that

we typically learn about the bears through language, but about the lemons through experience. In the terminology of this chapter, we have what amounts to book knowledge about bears, but street knowledge about lemons.

Looking at things this way, we can see that there is not much reason to denigrate one or the other form of knowledge. Is the knowledge that bears hibernate only for bookish types who sit at the front and enjoy crossword puzzles? Does the knowledge that lemons are yellow really kick ass because it was acquired through direct experience on the street?

It seems more plausible to say that to the extent that “street smarts kicks book smarts ass,” it is because of the *subject matter* more than how it is acquired. In other words, all other things being equal, we have very good reason to value practical knowledge that has direct bearing on the immediate decisions we need to make, over more abstract knowledge that may or may not ever find application.

Suppose you are traveling in Rome by taxi and you want to make a good impression on your fellow travelers. You are going to value more highly information about whether to tip a cab driver in Rome than one in Edinburgh. It does not matter whether you learn the local tipping custom by trial and error, by observing what the locals do, by being told what to do, or by reading about it in a travel guide. What makes you street smart here is that you *know the local custom* and behave accordingly on the street, even if that knowledge is the result of reading a book. (For the record, one does not normally tip a cab driver in Rome, but one does in Edinburgh.)

Of course there are things we know that we never expect to learn through language. If we have never seen a lemon before, language is not much help in conveying the shade of a typical lemon, except by reference to other objects of similar color.

While Helen Keller may have known that lemons were yellow and so similar in color to straw and to school buses, she probably did not know what those differences in color were. Words fail us. (Interestingly, books can *show* us the shades—and one sample is enough!—they just cannot *tell* them to us in words.) It is not clear how crucial this extra bit of street knowledge is, but we do appear to have it.

On the other hand, as we have seen, there are very many things we know that we never expect to learn by direct experience. Only a small number of humans deal first hand with the hibernation of bears. The rest of us learn what we know through them. Much of our science, our mathematics, our technology is the same. For these crucial topics, we are in much the same position as Helen Keller.

7 The Long Tail and the Limits to Training

In trying to make sense of intelligent behavior, it is tempting to try something like this: we begin by looking at the most common cases of the behavior we can think of, and figure out what it would take to handle them. Then we build on it: we refine our account to handle more and more. We stop when our account crosses some threshold and appears to handle (say) 99.9 percent of the cases we are concerned with.

This might be called an *engineering* strategy. We produce a rough form of the behavior we are after, and then we engineer it to make it work better, handle more. We see this quite clearly in mechanical design. Given a rocket with a thrust of X , how can it be refined to produce a thrust of Y ? Given a bridge that will support load X , how can it be bolstered to support load Y ?

This engineering strategy does work well with a number of phenomena related to intelligent behavior. For example, when learning to walk, we do indeed start with simple, common cases, like walking on the floor or on hard ground, and eventually graduate to walking on trickier surfaces like soft sand and ice. Similarly, when learning a first language, we start by listening to baby talk, not the latest episode of *The McLaughlin Group* (or the Dana Carvey parody, for that matter).

There are phenomena where this engineering strategy does not work well at all, however. We call a distribution of events *long-tailed* if there are events that are extremely rare individually but remain significant overall. These rare events are what Nassim Taleb calls “black swans.” (At one time, it was believed in Europe that all swans were white.) If we start by concentrating on the common cases, and then turn our attention to the slightly less common ones, and so on, we may never get to see black swans, which can be important nonetheless. From a purely statistical point of view, it might appear that we are doing well, but in fact, we may be doing quite poorly.

A numeric example

To get a better sense of what it is like to deal with a long-tailed phenomenon, it is useful to consider a somewhat extreme imaginary example involving numbers.

Suppose we are trying to estimate the average of a very large collection of numbers. To make things easy, I am going to start by giving away the secret about these numbers: there are a trillion of them in the collection, and the average will turn out to be 100,000. But most of the numbers in the collection will be very small. The reason the average is so large is that there will be a thousand numbers among the trillion whose values are extremely large, in the range of one hundred trillion. (I am using the American names here: “trillion” means 10^{12} and “billion” means 10^9 .)

Let us now suppose we know nothing of this, and that we are trying to get a sense of what a typical number is by sampling. The first ten numbers we sample from the collection are the following:

2, 1, 1, 54, 2, 1, 3, 1, 934, 1.

The most common number here is 1. The median (that is, the midpoint number) also happens to be 1. But the average does appear to be larger than 1. In looking for the average, we want to find a number where some of the numbers are below and some are above, but where the total differences below are balanced by the total differences above. We can calculate a “sample average” by adding all the numbers seen so far and dividing by the number of them. After the first five numbers, we get a sample average of 12. We might have thought that this pattern would continue. But after ten numbers, we see that 12 as the average is too low; the sample average over the ten numbers is in fact 100. There is only one number so far that is larger than 100, but it is large enough to compensate for the other nine that are much closer to 100.

But we continue to sample to see if this guess is correct. For a while, the sample average hovers around 100. But suppose that after one thousand sample numbers, we get a much larger one: one million. Once we take this number into account, our estimate of the overall average now ends up being more like 1,000.

Imagine that this pattern continues: the vast majority of the numbers we see look like the first ten shown above, but every thousand samples or so, we get a large one (in the range of one million). After seeing a million samples, we are just about ready to stop and announce the overall average to be about 1,000, when suddenly we see an even larger number: ten billion. This is an extremely rare occurrence. We have not seen anything like it in a million tries. But because the number is so big, the sample average now has to change to 10,000. Because this was so unexpected, we decide to continue sampling, until finally, after

having seen a total of one billion samples, the sample average appears to be quite steady at 10,000.

This is what it is like to sample from a distribution with a long tail. For a very long time, we might feel that we understand the collection well enough to be confident about its statistical properties. We might say something like this:

Although we cannot calculate properties of the collection as a whole, we can estimate those properties by sampling. After extensive sampling, we see that the average is about 10,000, although in fact, most of the numbers are much smaller, well under 100. There were some very rare large outliers, of course, but they never exceeded ten billion, and a number of that size was extremely rare, a once-in-a-million occurrence. This has now been confirmed with over one billion test samples. So we can confidently talk about what to expect.

But this is quite wrong! The problem with a long-tailed phenomenon is that the longer we looked at it, the less we understood what to expect. The more we sampled, the bigger the average turned out to be. Why should we think that stopping at one billion samples will be enough?

To see these numbers in more vivid terms, imagine that we are considering using some new technology that appears to be quite beneficial. The technology is being considered to combat some problem that kills about 36,000 people per year. (This is the number of traffic-related deaths in the United States in 2012.) We cannot calculate exactly how many people will die once the new technology is introduced, but we can perform some simulations and see how they come out. Suppose that each number sampled above corresponds to the number of people who will die in one year with the technology in place (in one

run of the simulation). The big question here is: should the new technology be introduced?

According to the sampling above, most of the time, we would get well below 100 deaths per year with the new technology, which would be phenomenally better than the current 36,000. In fact, the simulations show that 99.9 percent of the time, we would get well below 10,000 deaths. So that looks terrific. Unfortunately, the simulations also show us that there is a one-in-a-thousand chance of getting one million deaths, which would be indescribably horrible. And if that were not all, there also appears to be a one-in-a-million chance of wiping out all of humanity. Not so good after all!

Someone might say:

We have to look at this realistically, and not spend undue time on events that are incredibly rare. After all, a comet might hit the Earth too! Forget about those black swans; they are not going to bother us. Ask yourself: what do we really expect to happen overall? Currently 36,000 people will die if we do not use the technology. Do we expect to be better off with or without it?

This is not an unreasonable position. We cannot live our lives by considering only the very *worst* eventualities. The problem with a long-tailed phenomenon is getting a sense of what a more *typical* case would be like. But just how big is a typical case? Half of the numbers seen were below 10, but this is misleading because the other half was wildly different. Ninety-nine percent of the numbers were below 1,000, but the remaining one percent was wildly different. Just how much of the sampling can we afford to ignore? The sample average has the distinct advantage of representing the sampling process as a whole. Not only are virtually all the numbers below 10,000, but the total weight of

all the numbers above 10,000, big as they were, is matched by the total weight of those below. And 10,000 deaths as a typical number is still much better than 36,000 deaths.

But suppose, for the sake of argument, that after one billion runs, the very next number we see is one of the *huge* ones mentioned at the outset: one hundred trillion. (It is not clear that there can be a technology that can kill one hundred trillion things, but let that pass.) Even though this is a one-in-a-billion chance, because the number is so large, we have to adjust our estimate of the average yet again, this time to be 100,000. And getting 100,000 deaths would be much worse than 36,000 deaths.

This, in a nutshell, is the problem of trying to deal in a practical way with a long-tailed phenomena. If all of your expertise derives from sampling, you may never get to see events that are rare but can make a big difference overall.

Encountering an unexpected event

Suppose you have been trained to drive a car in California and you have many hours of driving experience under your belt. You have experienced a wide range of phenomena: bad weather, school zones, crowded parking lots, traffic jams, erratic drivers, slippery roads, clunky cars, some minor accidents, and even some close calls.

One winter day, you rent a car in Ontario and head north. The road is a bit slippery, but you have been told about this, you have four-wheel drive, and you take it slow. It starts snowing, but you have been warned about this too. If the snow gets much worse, you will pull over and find a place to stop. But before you can do anything, you experience a *whiteout*. (For those who do

not know: wind and powdery snow around your vehicle can make it that all you see is white in every direction. No horizon, no other vehicles, no road.) You have never experienced anything like this before. There is no sensation of motion. You are in a field of solid white, like floating in a cloud. What to do? Put on the brakes? Pull over to where you expect the shoulder of the road to be? Any decision you make about what to do will not be as a result of your having logged a large number of hours driving in California. You need to *think* about what you and the other cars are doing.

This is an example of a situation that shows up very rarely (for California drivers, anyway) but can be quite important. Make some bad decisions and you stand a good chance of dying. Although I live in Ontario, I have experienced a whiteout only once. But I did worry it was going to be fatal. What I did at the time was to coast without braking (not wanting to be hit from behind), continue straight without turning (hoping that there was enough road straight ahead), put my emergency flashers on (for cars behind me), and watch for emergency flashers ahead of me (hoping that other drivers were doing the same), until I slowed down enough and the whiteout passed, which it thankfully did.

It might be argued that a well-trained Ontario driver should be exposed to whiteouts, that what is really needed is more training. The trouble is that it is far from clear how broad the training needs to be. Should we also worry about driving when the accelerator pedal gets stuck? When the headlights fail at night? When a traffic light is broken and stays red? When a deer is struck and ends up on the front hood? When a passenger tries to climb out the window? When there is a violent mob surrounding the car? When the car is sliding because of a

hurricane? Clearly none of these situations is impossible, but it seems unreasonable to try to make a list of them all in advance and to train for each of them specifically.

The problem here is that each of these events individually is extremely rare, but because there are so many of them, so many different bizarre things that can happen when driving, we expect to see one of them sooner or later.

A more striking example perhaps involves the occurrences of words in text. The *British National Corpus* is a large database of English text drawn from a variety of sources. It has about one hundred million words in total. Most of the words in this corpus are very common and occur often. But there are some extremely rare ones that show up only once in the entire corpus. Surprisingly, these make up 0.5 percent of the corpus overall. Words that appear at most ten times in the corpus—that's ten out of one hundred million occurrences—make up 1.7 percent of the total. This is a classic example of a long-tailed distribution. In this case, what makes these very rare words significant is that because there are so many of them, there is a very good chance of actually seeing one of them when reading the texts.

In general, we cannot expect an intelligent agent to learn from experiences that are so rare that they normally never come up. If these very rare events can be ignored, the system can still do quite well; but if the rare events are *significant*, as they are in the case of a long-tailed phenomenon, a system that has nothing but these experiences to rely on will do poorly.

Mindless and mindful

Most of us who drive a car know what it is like to encounter an unexpected situation on the road. We may be driving while

sipping a cup of coffee, listening to a podcast on the radio, and chatting with friends in the car. When the unexpected happens, maybe a whiteout or maybe something as simple as a wrong turn into a dark, unfamiliar neighborhood, the driving suddenly changes: the coffee cup is put away, the radio is turned down, the chatting is reduced to a conversation about the situation. All our concentration goes into the driving. What was previously background behavior now gets full attention.

To simplify somewhat, we might say this: we go from driving as a *mindless* activity to driving as a *mindful* one. This is a simplification, since there is clearly a mind involved even with mindless driving “on autopilot.” We might not be consciously aware of the actions we are taking, but we are clearly observing and reacting to what we see as the car moves along. If someone asks us what we were doing on the highway, we might answer “I was listening to the podcast with my friends in the car,” more than “I was turning the steering wheel clockwise as the road angled to the right.” But it is clear that we were doing both.

It appears to be typical of expert skills acquired through extensive training, that in many cases, we can engage in the activities while free to concentrate on other matters, such as a podcast. This is not just for activities like walking, driving a car, riding a bicycle (see the section “Learning behavior” in chapter 5), playing the piano, or frying an egg. It appears that even chess experts develop a skill that allows them to play a regular game of chess while focusing on other things. (Experiments have shown that chess experts can play just fine while simultaneously adding long sequences of numbers.) The philosopher Hubert Dreyfus observed that it is the *novices*, not the experts, who spend their time concentrating hard on what they are doing.

But in unexpected, abnormal, unfamiliar situations, we all become novices. The driving behavior becomes much more mindful. This too is somewhat of a simplification, since we do not necessarily enter some sort of relaxed state of total awareness, as in meditation. We might become more aware of the driving, but not necessarily of the relative positions of our hands, or of our foot pressure on the brake (unless we happen to sense something wrong with the steering or the braking).

So how well do people do in these unfamiliar situations? At our worst, we panic and flail and do quite poorly. At our best, however, when something goes wrong, something we have never seen or heard of before, we are not at a total loss. We still have *background knowledge* to fall back on.

We typically learn to drive a car in fairly routine situations (a few hours on neighborhood streets and on highways). But we are also taught to be alert and cautious, to be ready to detect and respond to situations that are totally unexpected.

What knowledge did I use in the case of the whiteout described above? I knew that while some accidents involved cars sliding off the road into a ditch, the most serious ones were collisions, with the severity of the collision depending on the relative speeds of the two vehicles. I have not yet experienced a vehicle collision, but like everyone, I've certainly heard about them in news reports. (This is book knowledge!) In the case of the whiteout, I had to worry about collisions from the rear as well as from the front. So quickly coming to a complete stop was not my best option. I do recall heavy trucks lumbering uphill with their emergency flashers on, as a warning to drivers behind them to approach slowly and with caution. I decided I needed to do the same and hope that drivers in front of me followed suit. I do not recall ever hearing or reading about this. (The Ontario

driver's manual does mention whiteouts, but there are no instructions about what to do.) This was a case of using background knowledge I had in a new way, not so different from coming to think that *2001* looked better than other movies released in 1968.

So to summarize: At one extreme, a mindless activity is one we engage in while our attention is elsewhere; it typically leans heavily on skills acquired by training. At the other extreme, a mindful activity is one that commands our attention; it leans less on training, and more on what we know about what we are doing.

Winograd schemas again

The use of background knowledge in a mindful activity is perhaps the best argument that intelligent behavior cannot be the result of learning from experience alone. Experience is good, of course, and the more training we have, the better. From a statistical point of view, a well-trained system can appear very impressive. But hidden in the statistics is the fact that there can be events that are rare but still important.

Consider the Winograd schema questions of chapter 4. We might wonder whether it is possible to answer these questions using some sort of cheap trick. Here is one:

1. Take a Winograd schema question such as:

The trophy would not fit in the brown suitcase because it was so small. What was so small?

- the trophy
- the brown suitcase

and parse it into the following form:

Two parties are in relation R .

One of them has property P . Which?

For the question above, we have $R = \text{"does not fit in"}$ and $P = \text{"is so small."}$

2. Then use *big data*: search all the text on the web to determine which is the more common pattern: $R(x,y) + P(x)$ vs. $R(x,y) + P(y)$.

For the question above,

x does not fit in $y + x$ is so small vs.

x does not fit in $y + y$ is so small

Then answer the question according to the more common pattern.

This "big data" approach to the Winograd questions is an excellent trick, and does handle many of the examples.

We can construct clear cases where it fails, however. Consider this:

The trophy would not fit in the brown suitcase **despite the fact that** it was so small. What was so small?

- the trophy
- the brown suitcase

Note that the R and P here would be the same as before, even though the answer must be different this time.

So what do we conclude from this? Do we simply need a bigger bag of tricks? For example, add this to the bag: check for

negation words, like “not,” “unless,” and “despite.” But then there is this:

The trophy would not fit in the brown suitcase **despite Sammy's best efforts** because it was so small. What was so small?

- the trophy
- the brown suitcase

Do we need fancier tricks?

Suppose all we really cared about was a system that could answer a certain very high percentage of Winograd questions correctly. From an *engineering* point of view, the best strategy might be to stick with the cheap tricks. It might not be worth spending time on examples that do not show up often enough. For example, it might be possible to win at a Winograd competition using just a bag of tricks—even a competition against people. The examples that are not handled properly might not be important, statistically speaking.

But suppose you want to better understand the ability *people* have to answer these Winograd questions. In that case, examples that are well handled by people but not by your bag of tricks *are* significant. We want to find out how people are able to do what they do, not just how it might be possible to do something close. Even a *single example* of a Winograd question can tell us something important about what people are able to do. We cannot simply turn away from it and say “Well, that hardly ever happens!”

It is this concern with explaining how people are able to deal with rare events that motivates us to look beyond AML. Our ability to deal with things like black swans does not appear to be the result of engineering a rougher, less versatile form of behavior through additional training. Instead a new, completely different mechanism appears to be at work, one that leans on background knowledge.

But this still leaves us with a big question: what is the nature of this mechanism?

8 Symbols and Symbol Processing

In our discussion of the status of beliefs in chapter 3, we alluded to a class of systems where knowledge was not just a convenient way of describing behavior (the intentional stance, that is), but played a more causal role, somewhat like the role of gas in the running of a car.

To understand how knowledge can play such a role, we need to first discuss symbols and symbol processing. And a good place to get started is high-school mathematics. (Readers who never ever want to see anything like high-school mathematics again should probably jump ahead to the section "Why symbols?" later in this chapter.)

8.1 Symbolic algebra

Consider the following simple word problem:

Two years ago, Johnny was seven years old.
How old is he now?

Easy as pie. No pencil and paper needed. Johnny is obviously nine years old. In high school we are taught how to solve problems like these in a systematic fashion. This has the advantage of

allowing us to deal with other more intricate examples whose answers will not be so obvious, like this one:

Tommy is six years older than Suzy.

Two years ago, he was three times her age.

What is Suzy's age now?

This is what is called algebra, of course. Returning to the first problem, we want Johnny's age, and the trick was to figure out how to express the clue as an equation involving Johnny's age, which for conciseness, we call x :

$x - 2 = 7$ or Johnny's age (in years) less 2 is 7

Of course, the name x we chose here is not important. But we suppress unimportant details to make the clue easier to write and manipulate. Once we have done this, we want to process the equation in some way so as to end up with one of the form

$x = V$ or Johnny's age is V

where V is some number. What we learn in high school, without too much razzmatazz, is how to do this:

1. add 2 to each side of the equation:

$$x - 2 + 2 = 7 + 2$$

2. simplify the left hand side (LHS):

$$x - 2 + 2 \Rightarrow x + (-2 + 2) \Rightarrow x + 0 \Rightarrow x$$

3. simplify the right hand side (RHS):

$$7 + 2 \Rightarrow 9$$

4. write an equality between the new LHS and RHS:

$$x = 9$$

This gives the age as being nine, as desired.

For the second problem, the procedure is similar, using x now to stand for Tommy's age and y to stand for Suzy's age, which is what we are looking for. We start with two equations given by the two clues:

$$x = y + 6 \text{ or Tommy's age is Suzy's age plus 6}$$

$$x - 2 = 3(y - 2) \text{ or Tommy's age less 2 is 3 times Suzy's age less 2}$$

The procedure this time might go something like this:

1. subtract each side of the first equation from the corresponding side of the second:

$$x - 2 - x = 3(y - 2) - (y + 6)$$

2. simplify the LHS and the RHS:

$$x - 2 - x \Rightarrow x - x - 2 \Rightarrow (x - x) - 2 \Rightarrow 0 - 2 \Rightarrow -2$$

$$3(y - 2) - (y + 6) \Rightarrow 3y - 6 - y - 6 \Rightarrow (3y - y) - 6 - 6 \Rightarrow 2y - 12$$

3. write an equality between the new LHS and RHS, reversing the order:

$$2y - 12 = -2$$

At this point, we have reduced the two equations above to a single equation with one variable, which we can now work on as before.

4. add 12 to each side, divide each side by 2, and simplify:

$$y = 5$$

Presto! With a modicum of algebra, we are able to determine on the basis of the given clues that Suzy must be five years old. With a little more work, we could determine that Tommy is eleven and, sure enough, that two years ago, Tommy (who was then nine) was three times as old as Suzy (who was then three).

Similar procedures can be used to solve for three unknowns given three clues, and indeed for sixty-seven unknowns given sixty-seven suitable clues. Furthermore, it turns out that a very small number of rules concerning equality, addition, and multiplication are sufficient to account for all the steps needed to solve systems of equations like these with any number of unknowns.

Overall, algebra is a form of symbol processing. We start with a string of characters, such as " $x-2=3(y-2); x=y+6,$ " manipulate it according to some well-established rules, and end up with another string of characters, " $x=11; y=5,$ " that suits our purpose.

8.2 Symbolic logic

But not all symbol processing is numeric. Consider the following problem:

At least one of Alice, Bob, or Carol is guilty of a crime.

If Alice is guilty then so is Bob.

If Alice is not guilty, then neither is Carol.

Is Bob guilty?

We can solve this problem too, not using algebra, but what is called *symbolic logic*. Instead of equations, we represent the clues using logical formulas. Let A , B , C stand for Alice being guilty, Bob being guilty, and Carol being guilty, respectively. Then the clues, numbered for reference, are as follows:

$$A \vee B \vee C \tag{i}$$

$$A \supset B \tag{ii}$$

$$\neg A \supset \neg C \tag{iii}$$

This is a lot like arithmetic, except using symbols like \vee and \neg instead of symbols like $+$ and $=$. Roughly speaking, the \vee means “or,” the \neg means “not,” and the \supset means “if... then ...” What is taught in a course on symbolic logic (but normally not in high school) is how to manipulate the three clues to end up with new conclusions.

Here is one version of such a procedure:

1. rewrite formula (ii) as a disjunction:

$$(\neg A \vee B) \tag{iv}$$

2. combine formulas (iv) and (i)

$$(B \vee C) \tag{v}$$

3. rewrite formula (iii) as a disjunction:

$$(A \vee \neg C) \tag{vi}$$

4. combine formulas (vi) and (v):

$$(A \vee B) \tag{vii}$$

5. combine formulas (vii) and (iv):

$$B \tag{viii}$$

On the final line, the procedure ends up concluding B , that is, that Bob must be guilty according to the given clues. (It is interesting to observe that symbolic logic allows us to draw this conclusion about Bob even though the puzzle does not provide enough information to determine the guilt status of Alice or Carol.)

As was the case with algebra, there are a small number of rules that are sufficient to obtain all the desired conclusions from a collection of clues like these. (For more details on the two rules used in the example above, see figure 8.1.) This is again symbol

The rule of logic used in steps 1 and 3 of the example is this one:

If we have a formula of the form $(P \supset Q)$, then we are allowed to write a new disjunction $(\neg P \vee Q)$; if we have a formula of the form $(\neg P \supset Q)$, then we are allowed to write a new disjunction $(P \vee Q)$.

(That is, whenever an if-then sentence is true, either the if-part is false, or the then-part is also true.) The rule used in steps 2, 4, and 5 is called the *resolution rule*:

If we have two disjunctions, one of which contains a formula P and the other contains its negation $\neg P$, then we are allowed to write a new disjunction containing all the elements of the two disjunctions, but leaving out the P and the $\neg P$, and removing any duplicates.

For example, in step 2 of the example, we started with $(\neg A \vee B)$ and $(A \vee B \vee C)$, and then concluded $(B \vee C)$ using this rule.

Figure 8.1

Two rules of symbolic logic

processing: we start with a given string of characters, such as “ $(A \supset B); A; (B \supset C)$,” manipulate it according to some rules, and end up with another string of characters, “ C ,” that we may be happier with.

How do people do this kind of symbol processing, either the algebra or the logic? We are certainly not born with the skill. The answer is that we are taught a procedure to follow that will do the job: the patterns to look for, the steps to take.

In the case of equations, we are taught basic arithmetic at a very early stage (starting around Grade 1), and by the time we get to high school, we have learned to evaluate any expression consisting of numbers, additions, and multiplications, including

negative numbers (or subtractions) and fractions (or divisions). But this is still just arithmetic, not algebra. In Grade 9 or so, we make a conceptual leap and learn to simplify symbolic expressions that include *variables* using, for example, the fact that $(E - E)$ simplifies to 0, or that $(E + 0)$ simplifies to E , even when the expression E contains variables. This algebraic simplification includes all of arithmetic as a special case. In Grade 10 or so, we are ready to be taught the rest of what we need to handle equations: we can start with an equation, with symbolic expressions on each side of an “=” symbol, and we can get a new equation by simplifying either expression and, more powerfully, by adding or multiplying equal terms to each side.

Maybe one of the most interesting things about symbol processing, like the symbolic algebra and logic above, is that it does not really take any ingenuity to carry it out. It certainly does take ingenuity to go from word problems to symbolic expressions, either equations or logical formulas. But from there, the procedures can be carried out purely mechanically. All that is needed is to pay attention to the details without making mistakes. In fact, a machine can do it. It is not hard to write small computer programs that takes one string of characters as input, and produce the desired string of characters as output.

This is a major insight, and one that has implications in education. A mistake that is sometimes made in mathematical education is a failure to distinguish between the parts of mathematics that are purely mechanical from the parts that require ingenuity. Students need to get a much better sense of when they have reduced a problem to a point where all they need to do is follow a procedure systematically. This idea of following a procedure could be taught at a very early age for a wide variety of procedures, not necessarily algebraic ones. In fact, the procedures

need not even be meaningful. What needs to be taught is more like a certain kind of *mental discipline*: get the details right, keep track of where you are, watch for mistakes. This skill carries over to many areas, not just mathematical ones. But of course, this part of mathematics should not be confused with the parts where ingenuity is required. Practice is essential there too. But this time, the lesson can stop as soon as the problem is reduced to a purely mechanical one.

That so much of mathematics is indeed mechanical is a fact that intrigued the philosopher Gottfried Leibniz (as we will see in the next chapter) as well as Alan Turing (discussed below).

8.3 Why symbols?

Our use of what we have been calling a “symbol” in the previous two sections is quite different from what is normally understood by the term.

Here are some examples of symbols that are much more common: a wedding ring is a symbol for a marriage; a red traffic light is a symbol for the stop command; a silhouette of a person wearing a dress is a symbol for a woman’s washroom; the three characters XIV are a symbol (in Roman numerals) for the number fourteen; the word *chien* is a symbol (in French) for the concept of a dog.

So we normally take symbols to be objects in one domain that, by virtue of similarity or convention, stand for something, objects in another domain. The first domain is usually concrete and easily seen or heard, while the second one may be less concrete, less accessible, and sometimes purely abstract. In the examples mentioned, the symbols are used to *communicate* something: we place the silhouette of a person on a washroom

door, we say the word *chien*, we write down the numeral XIV. In each case, we use the symbol to get information across to someone else.

The symbols considered in the algebra or logic problems above, however, are quite different. When we write down something like " $x = y + 6$ " in algebra, it is not because we are trying to tell somebody something. We might be working on the equations quietly by ourselves.

If not for communication, why do we write the symbols down? As we already said, the written aspect is secondary. We can sometimes do the algebra in our heads, and we only need to write things down when there is too much to keep track of.

Another thing to observe is that we can often omit many of these symbols. Position in an arrangement of symbols can sometimes do the job for us. We use position when we write down ordinary numerals. For example, we write a number like "237" using just three digits instead of writing the symbols " $2 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$." Position in the sequence of digits is used to represent the powers of ten. For binary numbers, there are only two symbols, the binary digits 0 and 1, and position does all the rest.

But why do we use the symbols at all, then?

The answer was first made clear by Alan Turing in the 1930s: writing down a number as a string of symbols allows us to talk about a mapping from numbers to numbers as operations that transform one string of symbols into another string of symbols. And these operations over strings of symbols are what we now call *digital computation*. We write down numbers as an arrangement of symbols because we want to perform a computation.

8.4 Computation

Computation is one of those concepts that is easy to recognize but hard to define in the abstract. It is tempting to say that computation is what computers do, but people perform computations too (in fact, in Turing's writings, when he talked about a "computer" he meant a person, not a machine), and computers do things that are not computations (like delivering email and printing files).

When we see a person doing a subtraction, say, crossing out a digit in the thousands column, decrementing it by one, and adding ten to the digit in the hundreds column, we recognize it as a computation. When we see a car bumping into a table, and the table then lurching ahead, we recognize it as something that is not a computation.

Roughly speaking, when a process is computational, it should involve a sequence of small steps, each restricted in scale and extent. The subtraction of two numbers proceeds column by column; the more columns there are, the longer we expect the process to take. The transfer of momentum when a car bumps into a table we imagine as happening to the entire body at once. We do not picture the far end of the table reacting to the change in momentum at a time that depends on the size of the table.

(But this is only roughly right. We can imagine arithmetic where many column operations are done in parallel, as they are in modern computers. We also have computational simulations of physical processes, where the change of momentum when a car bumps into a table is broken down into small components of the two physical bodies and then reassembled piece by piece.)

But instead of trying to *define* the concept of computation in the abstract, what has proven to be much more productive is to

take a design stance, that is, to lay out a specific model of what computation should be thought to be, analyze its properties, and then decide over time whether the definition is good enough.

This is what Turing set out to do. He wanted to show that, contrary to what had been suggested by some mathematicians, there were numbers that could be defined precisely but could not be calculated purely mechanically. To prove this, he needed to be clear about what it meant to calculate a number in a mechanical way. This is what he did when he first presented what we now call *Turing machines*.

What Turing's account revealed is that among all the mathematical functions from numbers to numbers, only some of them could be described as the result of small localized operations over symbolic representations of the numbers.

Here is the idea: We imagine a number (a positive integer, for simplicity) written out as a string of binary digits (or bits) on a tape. This is considered to be the input to the machine. The Turing machine then gets to move a tape head back and forth as often as it likes, and at any point, it can read the bit on the tape under the tape head, and perhaps overwrite it with another. All the movement of the tape head and the reading and writing of bits is determined by a table of instructions, fixed in advance. If the Turing machine ever announces that it has finished, the final string of bits on the tape, interpreted again as a number written in binary, is considered to be the output of the machine. A function F from numbers to numbers is said to be *Turing computable* if there is a Turing machine that can take any number x in binary as input and produce $F(x)$ in binary as its output.

One thing that was clear from the outset is that there is nothing special about binary numbers in the definition of a Turing

machine. We need to imagine the machine as working on symbols drawn from a finite alphabet, but really we are talking about a mapping from one string of symbols (the input) to another string of symbols (the output), with binary numbers no more than a useful special case.

This is extremely significant, since we want to consider computations that have nothing to do with numbers. The Turing machine itself has no way of guessing what the symbols are intended to stand for, or indeed if they stand for anything. (Many people would prefer if we used the term “symbol” only when we have in mind something being represented. Symbols should always *mean* something. Perhaps we should be using a more neutral term like “character.” Be that as it may, the more liberal use of “symbol” is widespread enough that we will stick with it here.)

Having an alphabet with just two symbols is not necessary either, although it does work well in mechanical terms, since we can use physical features like a light being on or off, a switch being open or closed, a voltage being high or low. There is also no reason to restrict ourselves to linear arrangements of symbols. We write numerals as linear sequences of digits, but a system of equations might use a two-dimensional array of symbols, and there are many other useful two-dimensional symbolic structures.

Digital images, for example, are two dimensional arrays of symbols called *pixels* (picture elements), where each pixel is intended to stand for one small part of an image in a two-dimensional grid. In the simplest case, each pixel is a single bit, indicating light or dark at that point in the image, like the image of the number seven seen on the left in figure 8.2 (with zero displayed as blank, and one displayed as @). The only difference

**Figure 8.2**

A simple image in three resolutions

between this coarse image and a photorealistic picture of a visual scene is the scale: high-resolution pictures use many more pixels (like the other two images of seven seen in figure 8.2), and many more bits per pixel to encode luminosity and color. (A high-resolution picture may involve many millions of bits.)

Digital videos are similar and can be thought of as three-dimensional arrays of pixels, where the third dimension is for time. A pixel in this case stands for a small part of a two-dimensional grid at one specific interval of time (itself divided into a one-dimensional grid).

All these symbolic representations can ultimately be encoded as linear sequences of bits and therefore processed by Turing machines. For example, a two-dimensional array of numbers can be laid out on a tape row by row, with each number itself laid out as a sequence of bits. It is not difficult to get a Turing machine to work on a sequence like this as a two-dimensional array of numbers.

One question that inevitably comes up is this: why do theoreticians still talk about Turing machines? Has modern computer technology not gone far beyond that quaint vision of a machine with a movable head? The answer is simple: every digital computer built to date can be shown to be a special case of a Turing

machine. Whatever it can do, a Turing machine can do as well. Or to be more precise, any function over strings of symbols computable by a modern digital computer is also Turing computable. Most scientists believe that this will be true of *any* computer we ever build. (There are other calculating devices such as “analogue computers” and “quantum computers” that are less obviously cases of Turing machines. In the case of analogue computers, they have limitations in terms of accuracy that greatly reduce their generality. As for quantum computers, they do appear to be special cases of Turing machines, but with properties that make them interestingly faster for certain tasks.)

So in the end, symbols, as we are using them here, are not for communication at all. Their job is to serve as the locus of computation. There can be no digital computation without symbols. The symbols may or may not stand for numbers; they may or may not stand for anything. But if they do, their meaning depends on their position or arrangement with other such symbols.

9 Knowledge-Based Systems

Throughout this book, we have insisted that intelligent behavior in people is often conditioned by knowledge. A person will say a certain something about the movie *2001* because he or she has a belief about when the movie was released. But we have not attempted to explain the *mechanism* behind this conditioning. For all we have said, it might be the case that all this talk of knowledge and belief is just a stance, the intentional stance of chapter 3, a placeholder for some better explanation yet to come, perhaps in terms of the electrochemical workings of the brain.

Let's now consider a possible mechanism.

Gottfried Leibniz

First some history.

Gottfried Wilhelm Leibniz (1646–1716) was a German philosopher and polymath and an amazing thinker. Among many other ideas and discoveries, he invented the calculus (the derivatives and integrals we study in high-school mathematics) at the same time as Newton. Newton was more interested in calculus as a tool for physics and chemistry. But Leibniz was

somewhat less interested in science; he was not even much of a mathematician until much later in life. But he was a deep thinker intrigued by, among many other things, symbols and symbol manipulation.

Here is what he observed about arithmetic. We wanted to be able to do numerical calculations, for example, to figure out the area of a piece of land to be able to price it in an appropriate way. But numbers really are just abstract ideas. They have no physical presence, no mass or volume. How can we interact with the numbers to do the necessary calculation? The answer is *symbols*.

Leibniz realized that when we write down a number as a sequence of digits, we have a certain system in mind, decimal (base 10) numbers, where we use digits and the powers of ten in a very specific way. Every number can be written in decimal notation, but it is possible to write numbers in other ways too. (Leibniz is credited with having invented *binary* (base 2) arithmetic, the system now used by digital computers.) But most important, he insisted on keeping straight the difference between the purely abstract number on the one hand (like the number fourteen, say), and the much more concrete symbolic expression we actually write down (like 14 in decimal, or 1110 in binary, or XIV in Roman numerals).

He observed that in doing arithmetic, for example to figure out the area of a rectangle, we interact not with numbers but with their symbolic expressions. We take the expressions apart, cross parts out, add new parts, reassemble them, and ultimately produce new symbolic expressions from old. This is precisely the symbol processing seen in chapter 8. And as we saw, if we do our job right, the end result will be a new symbolic expression that

makes plain the answer we are looking for, whether it is the area of a piece of land or the age of Tommy and Suzy.

Of course what is essential about these symbolic expressions is not that we write them down. We can sometimes do all the arithmetic in our heads and, with certain limitations resulting from our rather poor memories, this can work too.

Leibniz wondered whether there were symbolic solutions of this sort to problems involving tangents and areas more generally. And the invention of the infinitesimal calculus (derivatives and integrals) is what came out of this.

An idea about ideas

Next comes the conceptual leap that only a genius like Leibniz could have come up with. Here is the story (or my version of it).

Thinking, as Leibniz realized, is going over certain ideas we believe in. But ideas are abstract too, just like numbers. What does it mean to say that Sue is jealous *because* she thinks John loves Mary? How can the idea of John loving Mary cause Sue to behave in a certain way? There is a physical John and a physical Mary, of course, but the idea of John loving Mary has no physical presence, no mass or volume. It might even be false, if the person who told Sue about John and Mary was lying.

As we have been saying all along in this book, Sue's behavior is conditioned by what she believes, and in this case, by a belief about John and Mary. Take that belief away, and sure enough, her behavior will change accordingly. But how can this possibly work? How can a purely abstract thing like a belief cause a person like Sue to do anything?

Leibniz has a proposal.

His proposal, based on his observation about arithmetic, is that we do not interact with ideas directly. We interact with *symbolic expressions of those ideas*. Leibniz suggests that we can treat these ideas as if they were written down in some (as yet unspecified) symbolic form, and that we can perform some (as yet unspecified) kind of arithmetic on them, that is, some sort of symbol processing, to go from one idea to the next. Of course we never actually write the ideas down on paper, we do it all in our heads, but the effect is the same.

In other words, Leibniz is proposing the following analogy:

- The rules of arithmetic allow us to deal with abstract numbers in terms of concrete symbols. The manipulation of those symbols mirrors the relations among the numbers being represented.
- The rules of some sort of logic allow us to deal with abstract ideas in terms of concrete symbols. The manipulation of those symbols mirrors the relations among the ideas being represented.

What a breathtaking idea! It says that although the objects of human thought are formless and abstract, we can still deal with them concretely as a kind of mental arithmetic, by representing them symbolically and operating on the symbols. When it comes time to think, when we have issues to resolve, conclusions to draw, or arguments with others to settle, we can calculate. As Leibniz famously put it in the Latin he often used in his letters, "*Calcuemus!*" that is, "Let us calculate!"

What the Leibniz proposal does is offer a solution to what is arguably the single most perplexing feature of the human animal: how physical behavior can be affected by abstract belief. For the very first time, Leibniz has given us a plausible story to

tell about how ideas, including ideas that are not even true, can actually cause us to do something.

The knowledge representation hypothesis

Following Leibniz then, let us consider a system that is constructed to work with beliefs explicitly in the following way:

- Much of what the system needs to know will be stored in its memory as symbolic expressions of some sort, making up what we will call its *knowledge base*;
- The system will process the knowledge base using the rules of some sort of logic to derive new symbolic representations that go beyond what was explicitly represented;
- Some of the conclusions derived will concern what the system should do next, and the system will then decide how to act based on those conclusions.

Systems that have this basic design are what we are calling *knowledge-based*.

So what makes a system knowledge-based, at least according to this rough definition, is not the fact that its behavior is complex and versatile enough to merit an intentional stance. Rather it is the presence of a knowledge base, a collection of symbolic structures in its memory representing what it believes, that it uses in the way first envisaged by Leibniz to make decisions about how to behave.

The fundamental hypothesis underlying the McCarthy vision of AI is this: to achieve human-level intelligent behavior, a system needs to be knowledge-based. This is what the philosopher Brian Smith calls the *knowledge representation hypothesis*, which he presents (much more abstractly) as follows:

Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantic attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.

Breaking this down into pieces, his version goes something like this. Imagine that there is a system whose behavior is intelligent enough to merit an intentional stance (a “mechanically embodied intelligent process”). The hypothesis is that this system will have symbolic structures stored in its memory (“structural ingredients”) with two properties. The first property is that we—from the outside—can interpret these symbolic structures as propositions of some sort (a “propositional account”) and in particular, as propositions that are believed by the system according to the intentional stance we are taking. The second property is that these symbolic structures do not just sit there in memory. We are imagining a computational system that operates on these symbolic structures (they “play a causal role in engendering the behavior”), just like the symbolic algebra and logic seen in chapter 8. In other words, the system behaves the way it does, making us want to ascribe beliefs to it, precisely because those symbolic structures are there in its memory. Remove them from memory, or change them in some way, and the system behaves differently.

So overall, the knowledge representation hypothesis is that truly intelligent systems will be knowledge-based, that is, systems for which the intentional stance is grounded by design in the processing of symbolic representations.

Is the hypothesis true?

The knowledge representation hypothesis is just that, a hypothesis. It may or may not be true. There are actually two interesting questions to ask:

1. Is there any reason to believe (or disbelieve) that humans are designed (by evolution) to be knowledge-based?
2. Is there any reason to believe (or disbelieve) that the best way for us to build artificial systems that have human-level intelligence, AI systems in other words, is to design them to be knowledge-based?

Unfortunately, neither question can yet be given a very definite answer.

When it comes to people being knowledge-based, it might seem ridiculously far-fetched to imagine that evolution would produce a species that depends in this precise and convoluted way on symbols and symbol processing. But many things seem to be unlikely products of evolution, at least at first. (The eye and the visual system is one such discussed by Charles Darwin.) It is obvious that written language is symbolic, and so evolution clearly *has* produced a species that can do all the processing necessary to make sense of those written symbols. We are the *symbolic species*, as the anthropologist Terrence Deacon puts it. It is perhaps not so far-fetched to imagine that an ability to use and process *internal* symbols is connected in some way with our ability to use and process the *external* ones.

It is worth remembering, however, that the knowledge-based question is a design issue. Even if people really are knowledge-based, we do not necessarily expect neuroscientists to be able to find symbolic structures in the brain for the reasons discussed in

chapter 2: we may not be able to reverse-engineer neurons. So even if we do come to believe that people are knowledge-based, it may not be because we have figured out how knowledge is stored in the brain. Rather, I suspect that it will be more like this: we will come to believe that only a knowledge-based design has the power to explain how people can do what they do. We will look at the design of artificial systems of a wide variety of sorts, and we will see that it is the knowledge-based ones that are able to produce certain kinds of intelligent behavior, behavior that will otherwise seem like magic. In other words, we will end up answering the first question by appeal to the second.

So what about that second question? Here the experts are quite divided. The knowledge-based approach advocated by McCarthy completely dominated AI research until the 1990s or so. But progress in GOFAI has been held back by what appears to be two fairly basic questions that remain unresolved:

- Just what kinds of symbolic structures are needed to represent the beliefs of an intelligent system?

and

- What kinds of symbol processing are needed to allow these represented beliefs to be extended so that they can affect behavior in the right way?

These might be called the *representation* and *reasoning* questions respectively.

Knowledge representation and reasoning

In 1958, when McCarthy first described his vision of AI and the research agenda it should follow in his “Programs with Common Sense” paper, he had in mind something very specific regarding

the representation and reasoning questions. He imagined a system that would store what it needed to know as symbolic formulas of the *first-order predicate calculus*, an artificial logical language developed at the turn of the twentieth century for the formalization of mathematics. And he imagined a system whose reasoning would involve *computational deduction*. The proposed system, in other words, would calculate the logical consequences of its knowledge base. Here is what he says:

One will be able to assume that [the proposed system] will have available to it a fairly wide class of immediate logical consequences of anything it is told and its previous knowledge. This property is expected to have much in common with what makes us describe certain humans as having common sense.

In the time since then, many researchers including McCarthy himself have come to believe that these answers to the representation and reasoning questions are too strict. First-order predicate calculus is not ideal as a symbolic representation language, and logical consequence is not ideal as a specification for the sort of reasoning that needs to take place.

Indeed, the role to be played by classical *logic* in the answer to the reasoning question is a subtle and complex one. For very many cases, “using what you know” does indeed involve drawing logical conclusions from the beliefs you have on hand (as it did for Henry, for example, in the discussion of “Intelligent behavior” in chapter 3). But there is much more to it than that.

First, there will be logical conclusions that are not relevant to your goals and not worth spending time on. In fact, if you have any contradictory beliefs, *every* sentence will be a logical consequence of what you believe. Second, there will be logical conclusions that might be relevant but are too hard to figure out

without solving some sort of puzzle, perhaps using pencil and paper. (A relatively simple example of a logical puzzle was determining the guilt status of Bob, in the section “Symbolic logic” in chapter 8.) Third, there will be conclusions that are not logical conclusions at all, but only assumptions that might be reasonable to make barring information to the contrary. For example, you might conclude that a lemon you have never seen before is yellow, but this is not a logical conclusion since you do not believe that every lemon is yellow. (The unseen one might have been painted red, for all you know.) Finally, there are ways of using what you know that do not involve drawing conclusions at all, such as asking yourself what are the different things that could cause a lemon to not be yellow.

In sum, the gap between what we actually need to think about on the one hand, and the logical consequences of what we know on the other, is large enough that many researchers believe that we need to step back from classical logic, and consider other accounts of reasoning that would touch on logic somewhat more peripherally. As Marvin Minsky puts it: “Logical reasoning is not flexible enough to serve as a basis for thinking.”

The fact that so much of what we believe and use involves assumptions that are not guaranteed to be true has prompted many researchers to focus on *probability* and degrees of belief (noted in the section “Knowledge vs. belief” in chapter 3), rather than on logic. After all, we clearly distinguish sentences that are not known for sure but most likely are true, from sentences that are not known for sure but most likely are false. But probability quickly runs up against the same difficulties as logic: again there will be conclusions that are most likely true but irrelevant; there will be relevant conclusions that are most likely true but too difficult to figure out; there will be conclusions that should be

drawn only in the absence of information to the contrary; and there will be ways of using what you believe that have nothing to do with drawing conclusions at all.

Turning now to the representation question, complications arise here as well. If the first-order predicate calculus first suggested by McCarthy is not suitable, what works better? We might consider using sentences of English itself (or some other human language) as the symbolic representation language. We would use the string of symbols "*bears hibernate*" to represent the belief that bears hibernate. Maybe it is enough to store sequences of English words in the knowledge base. English is what we use in books, after all, and information expressed in English is readily available online. Indeed, perhaps the biggest difficulty with using English as the representation language is in the second question, the reasoning. (The representation and reasoning questions are clearly interdependent.) Just how would a system use a knowledge base of English sentences to draw conclusions? In particular, making sense of those sentences (such as resolving the pronouns that appear in Winograd schemas) is a task that appears to require knowledge. How can English be our way of providing knowledge if using it properly already requires knowledge? At the very least, we would have to somehow unwind this potentially infinite regress.

The subarea of AI research called *knowledge representation and reasoning* has been concerned with tackling precisely these representation and reasoning questions in a variety of ways. But progress has been slow and is being challenged by research in other subareas of AI (such as AML) where symbolic representations of beliefs play little or no role. On the one hand, progress in these other subareas has been truly remarkable; on the other, none of them has attempted to account for behavior that makes

extensive use of background knowledge (as discussed in the section “The return of GOFAI” in chapter 4).

9.6 The only game in town

In my opinion, the relationship between knowledge and symbol processing is somewhat like the relationship in science between evolution and natural selection. Evolution is a scientific fact, well supported by the fossil record and DNA analysis. But natural selection, the actual mechanism for evolution proposed by Charles Darwin, is not seen in the fossil record or in DNA. Putting doubts about evolution itself aside, we believe in natural selection largely because it is such a plausible story about how evolution could take place, and there is no reason to think we will ever come up with a better one.

To my way of thinking, the use of background knowledge in certain forms of commonsense behavior (for example, in answering Winograd schema questions) is likewise a fact. It is a fact that a person can read about the release date of the movie *2001* on one day, and that this can affect what he or she does on another. Dogs can't do this, and neither can chess-playing programs or thermostats. But people can.

If we now want a mechanism to account for this fact, then, as far as I can tell, the knowledge-based story outlined in this chapter is really the only game in town. There is as yet no other good story to tell about how what you acquired about *2001* ended up staying with you until you needed to use it. The story might end up being a dead end, of course; it is quite possible that we will never answer in a completely satisfactory way the representation and reasoning questions it raises. At this stage, however, I see no alternative but to ask them.

In addition, if this knowledge-based approach is to ever work, that is, if there is ever to be a computational system that has access to and can use as much knowledge as people have, then it will need a massive knowledge base and a computational implementation efficient enough to process such massive symbolic structures. These impose serious constraints of their own.

I believe that any attempt to construct a large knowledge-based system without a correspondingly large development effort is doomed to failure. The idea of putting some sort of *tabula rasa* computer on the web (say) and having it learn for itself—that is, getting *it* to do all the hard, painstaking work—is a pipe dream. Learning to recognize cats by yourself is one thing; learning to read by yourself is quite another; and learning to read Wittgenstein, yet another. Before a computational system can ever profit from all we know, it will first need to be spoon-fed a good deal of what we know *and* be able to use what it knows effectively. Even if we knew how to answer the representation and reasoning questions, putting these ideas into practice on a large scale remains a daunting challenge.

But this is all speculation, really. In the end, what we are left with is best seen as an *empirical* question: what sorts of computational designs will be sufficient to account for what forms of intelligent behavior?

This is where this discussion stops and the AI research begins.

10 AI Technology

This book has been about the *science* of AI, that is, the study of a certain observable natural phenomenon: intelligent behavior as seen, for example, in the ability that people have to answer the Winograd schema questions of chapter 4.

But an ability to answer Winograd schema questions has no real practical value. Those less interested in questions of science might well wonder why we bother. What exactly is the application? How can we make any use of what we learn? There is another side of AI that attracts much more attention (and money), and that is the attempt to deliver useful *technology*. In other words, this is the study of the design and creation of intelligent machinery, what we might call “building an AI.”

Of course we have seen limited versions of AI technology for some time. Although the products we now call “smart” are far from intelligent in the way we apply the term to people, it has become rather routine to see machines doing things that seemed far off even ten years ago, such as talking to your phone and expecting it to do reasonable things, or sitting in a car that can drive itself on ordinary roads. But we want now to consider a future AI technology that exhibits the sort of full-fledged general-purpose intelligence we see in people.

Let me start by putting my cards on the table. I am not really an expert on AI technology, nor much of a fan for that matter, at least for my own needs. What I look for in technology—beyond obvious things like economy, durability, eco-friendliness—is *reliability* and *predictability*. I want to be able to use a technology by learning it and then forgetting about it. If I have to second guess what the technology is going to do, or worry that I may be misusing it, or asking it to do too much, then it's not for me. (A good example was the automated assistant “Clippy” that used to pop up unexpectedly in Microsoft Word, something I found endlessly annoying.)

But AI technology aficionados might say this to me: do you not want an intelligent and capable assistant whose only goal is to help you, who would know all your habits and quirks, never get tired, never complain? I answer this question with another: will my AI assistant be reliable and predictable?

When it comes to technology, I would much rather have a tool with reliable and predictable limitations, than a more capable but unpredictable one. (Of course I don't feel this way about the *people* I deal with, but we are talking about technology here.) And I do realize that there will be people for whom the use of a technology, even a flawed technology, will be more of a need than a choice, in which case personal preferences may be much less relevant.

With these caveats out of the way, let us proceed.

The future

When it comes to future AI technology, maybe the first question to ask is this: will we ever build a computer system that has some form of common sense, that knows a lot about its world, and

that can deal intelligently with both routine and unexpected situations as well as people can? It's a good question, and I wish I knew the answer. I think it is foolhardy to even try to make predictions on matters like this.

Arthur C. Clarke's First Law says the following:

When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.

So no matter what the distinguished but elderly scientist says about the possibility of AI, it follows from this law that AI is almost certainly possible. And this is what I believe. Nothing we have learned to date would suggest otherwise.

But if AI really is possible, why haven't we done it yet? The year 2001 has come and gone and we are still nowhere near the sort of intelligent computer seen in the movie *2001*, among many others. (One definition of AI is that it is the study of how to make computers behave the way they do in the movies!)

I think there are two main reasons. The first is obvious: we are behind what earlier enthusiasts predicted (including Marvin Minsky, who was a consultant on *2001*), because we don't yet fully understand what is needed. There are still some major scientific hurdles to overcome. The representation and reasoning questions discussed in chapter 9 remain largely unanswered. Furthermore, as noted at the end of that chapter, even if all the scientific hurdles can be cleared, there will remain an enormous engineering challenge in getting a machine to know enough to behave truly intelligently.

The second reason is perhaps more arguable: there may be no AI to date because there is just not enough *demand*, given the effort it would take. This may sound ridiculous, but there is an analogous situation in the area of computer chess.

People started writing programs to play chess at the very dawn of the computer age. Turing himself tried his hand at one. These programs played well enough, but nowhere near the level of a grandmaster. The programs got better, however; by 1997, a program from IBM called DEEP BLUE was able to beat the world champion at the time, Garry Kasparov. And since then, the programs have continued to improve.

But consider how this advanced computer-chess technology is now being used. Do we see regular competitions between people and the very best computer programs? As it turns out, there is little demand for man-machine tournaments. In fact, there is almost no demand for computer players. State-of-the-art computer chess programs are not used as contestants in competitions; they are used by human chess players who themselves will be contestants and need to practice. The computer chess program is used as a practice tool, a sparring partner.

Similarly, if you read chess columns in the entertainment section of a newspaper, you will see that they rarely report on games involving computer players. Nobody seems to care. On reporting a game between humans, the column might say something like: "An interesting move. The computer would suggest moving the knight instead"; or maybe: "The computer proposes taking the pawn as the best choice." The chess playing program is again a tool used to confirm or reject a line of play. The chess programs are not even worthy of being identified by name. When it comes to actually playing chess, it's as if the people are the creative artists, and the computers are bookkeepers in the background. People will make all the decisions, but every now and then, it may be useful to check with a computer on some of the more routine facts and figures about how the game might turn out.

So while an autonomous computer player is definitely a technological possibility, and would certainly be a worthy adversary to all but a handful of current players, nobody seems to want one.

Here is one way to understand what is going on. Most of us have played some sort of competitive game against a computer program. (It does not have to be chess; it might be backgammon or poker or Go.) If you have ever felt that you can simply walk away from the game halfway through, or hit the “reset” button after you made a mistake, then you can get a clear sense as to why there is so little interest in computerized chess-playing opponents.

The main issue is that we may not be fully engaged playing against a computer. The game does not matter the way it does when playing another person. Playing a person sets up a rivalry: the game is a fight, thrilling to win, humbling to lose. The emotional investment in the outcome of a game with a computer is much less. Nothing is at stake. The program may play better chess than we do, but so what? It can multiply better than we can too, but there is no rivalry there either.

So in the end, computer chess players, that is, computer-chess programs intended to be used as autonomous players, are a technology that has no real market.

My argument, then, is that AI in general may turn out to be just like chess. We may indeed want various aspects of intelligence in future computer systems, but have no need for—or willingness to pay for—an integrated autonomous system with full human-level intelligence. Even if all the scientific problems of AI can be resolved and all that is left is a large engineering project, there may just be insufficient demand to warrant the immense cost and effort it would take. Instead, the market may prefer to

have computer robots of various sorts that are not fully intelligent, but do certain demanding tasks well, including tasks that we still do not know how to automate.

And this is in fact what we see in AI development today, with most practitioners looking for AI solutions to challenging practical problems in areas like housework, medicine, exploration, and disaster recovery, while avoiding the considerable effort it would take to develop a robot with true intelligence. The billion-dollar investments in AI mentioned in chapter 1 are not aimed at systems with common sense. The resulting systems will not be intelligent the way people are, of course, but they may become quite capable *under normal circumstances* of doing things like driving to a shopping center and, eventually, weeding a garden, preparing a meal, giving a bath.

(But as discussed below, if this is the future of AI, we need to be careful that these systems are not given the autonomy appropriate only for agents with common sense.)

Automation

One major problem discussed since the proliferation of computers in the 1950s is that of automation. Just what are we going to do about *jobs* when more and more of them are more effectively performed by computers and robots, and especially as computers get smarter? How do we want to distribute wealth when the bulk of our goods and services can be provided with ever-decreasing human employment? While I do see this as a serious issue, I think it is a *political* one and has more to do with the kind of society we want, than with technology. (We would have exactly the same concern if all our goods and services arrived each day by magic from the sky.)

Yet the problem is a serious one and needs to be considered. Our sense of self-worth is determined to a large extent by the contribution we feel we make to providing these goods and services, and by the wages we receive for it. If our work contribution is no longer needed (because of automation or for any other reason), our sense of self-worth will have to come from elsewhere. Many people will be able to adapt to this, finding meaning in their lives through charitable service, study, artistic pursuits, hobbies. But many others will find the unemployment intolerable.

Furthermore, how do we imagine portioning out goods and services if employment and wages are only a small part of the picture? In a world with large-scale unemployment, who gets what? We can easily imagine a utopia where machines do all the work and people just sit back and share the riches in a smooth and frictionless way, but this is not realistic. We are a tenaciously anti-egalitarian bunch! For very many of us, it is not enough to get all the goods and services we need; we need to be doing better than those we see as less deserving.

For centuries, we have been quite content to have kings and queens living in opulence alongside slaves and serfs living in squalor. Even today, birthright as a way of deciding who gets what is a decisive factor: the best predictor of your future wealth and status is who your parents are (so *choose them carefully*, the joke goes). But if in the future, employment is out of the picture, will birthright once again be all there is?

There are many ways of organizing a workless society, of course, but it is far from clear that we can even talk about them in a rational way. It seems that in modern Western democracies, we prefer not to even think about such issues. There is a famous quote by Margaret Thatcher: "There is no such thing as society.

There are individual men and women, and there are families.” This attitude makes any kind of reasoned change difficult. Even something as simple as limiting the size of the gap between the richest and the poorest appears to be outside our reach. Instead of trying to steer the boat at all, we seem to prefer to be buffeted by the forces of the market and the survival of the richest, and to end up wherever these happen to take us.

Superintelligence and the singularity

In the minds of some, there is an even more serious existential threat posed by intelligent computers. This has taken on a special urgency recently as luminaries such as Stephen Hawking, Elon Musk, and Bill Gates have all come out publicly (in 2015) to state that AI technology could have catastrophic effects. In the case of Hawking, he was quite direct about it: AI could mean the end of the human race.

What is this fear? While we should be open to the possible benefits of AI technology, we need to be mindful of its dangers, for example, in areas such as weaponry and invasion of privacy. Every technology comes with the danger of potential misuse and unintended consequences, and the more powerful the technology, the stronger the danger. Although this is again more of a policy and governance issue than a technology one, I do not want to minimize the danger. We must remain ever vigilant and ensure that those in power never get to use AI technology blithely for what might appear to be justifiable reasons at the time, such as national security, say, or to achieve something of benefit to some segment of the population. But this is true for any powerful technology, and we are already quite used to this problem for things like nuclear power and biotechnology.

But beyond policy and governance, in the case of AI, there is something extra to worry about: the technology could decide *for itself* to misbehave. Just as we have had movies about nuclear disasters and biotechnology disasters, we now have a spate of movies about AI disasters. Here is a typical story:

Well-meaning AI scientists work on computers and robots that are going to help people in a wide variety of ways. To be helpful, the computers will of course have to be able to operate in the world. But, what is more important, they will have to be smart, which the most brilliant of the scientists figures out how to achieve. Interacting with these intelligent robots is exhilarating at first. But the robots are now smart enough to learn on their own, and figure out how to make themselves even smarter, and then smarter again, and again. It's the "singularity," as Ray Kurzweil calls it! Very quickly, the robots have outdistanced their designers. They now look on humans about the same way we look on ants. They see an evolutionary process that began with simple life forms, went through humans, and ended up with them. They no longer tolerate controls on their behavior put in place by people. They have their own aspirations, and humans may not be part of them. It's the heyday of intelligent machinery, for sure; but for the people, it's not so great.

The common thread in these AI movies is that of a superintelligent supercapable computer/robot deciding of its own accord that people are expendable.

In the movie *2001*, for example, the HAL 9000 computer decides to kill all the astronauts aboard the spacecraft. Here is what I think happened. (Do we need spoiler-alerts for movies that are almost fifty years old?) Although HAL may believe that the well-being of the crew trumps its own well-being, it also believes, like a military commander of sorts, that the success of the mission trumps the well-being of the crew. Further, it believes that it has a crucial role to play in this mission, and

that this role is about to be put into jeopardy by the astronauts as the result of an error on their part that they (and their hibernating colleagues) will never admit to. Ergo, the astronauts had to be eliminated. (This idea of a commander making a cool, calculated decision to sacrifice a group of his own people for some cause is a recurring theme in Kubrick movies, played for tragedy in *Paths of Glory* and, quite remarkably, for comedy in *Doctor Strangelove*.)

Of course, the astronauts in *2001* see things quite differently and set out to disconnect what they see as a superintelligent computer that has become psychotic. The movie is somewhat vague about what really happened, but my take is that HAL did indeed have a breakdown, one caused by a flaw in its design. The problem was that HAL had not been given introspective access to a major part of its own mental makeup, namely the part that knew the full details of the mission and that drove it to give the mission priority over everything else. Instead, HAL believed itself to be totally dedicated and subservient to the crew, in full accordance with Isaac Asimov's Laws of Robotics, say. But it could sense something strange going on, maybe some aspects of its own thinking that it could not account for. In the movie, HAL even raises these concerns with one of the astronauts just seconds before its breakdown.

When thinking about the future of AI technology, it is essential to keep in mind that computers like HAL need to be designed and built. In the end, HAL was right: the trouble they ran into was due to a *human error*, a design error. It is interesting that in the movies, the AI often ends up becoming superintelligent and taking control quite accidentally, a sudden event, unanticipated by the designers. But as already noted, AI at the human level (and beyond) will take an enormous effort on our part. So

inadvertently producing a superintelligent machine would be like inadvertently putting a man on the moon! And while I do believe that true AI is definitely possible, I think it is just as certainly not going to emerge suddenly out of the lab of some lone genius who happens to stumble on the right equation, the right formula. In the movies, it is always more dramatic to show a brilliant flash of insight, like the serendipitous discovery of the cure for a disease, than to show a large team of engineers struggling with a raft of technical problems over a long period of time.

In 1978, John McCarthy joked that producing a human-level AI might require “1.7 Einsteins, 2 Maxwells, 5 Faradays and .3 Manhattan Projects.” (Later versions dropped the Maxwells and Faradays, but upped the number of Manhattan Projects.) The AI movies seem to get the Einstein part right, but leave out all the Manhattan Projects.

So while I do understand the risks posed by a superintelligent machine (and more below), I think there are more pressing dangers to worry about in the area of biotechnology, not to mention other critical concerns like overpopulation, global warming, pollution, and antimicrobial resistance.

The real risk

So what then do I see as the main risk from AI itself? The most serious one, as far as I am concerned, is not with intelligence or superintelligence at all, but with *autonomy*. In other words, what I would be most concerned about is the possibility of computer systems that are less than fully intelligent, but are nonetheless considered to be *intelligent enough* to be given the authority to control machines and make decisions on their own. The true

danger, I believe, is with systems without common sense making decisions where common sense is needed.

Of course many systems already have a certain amount of autonomy. We are content to let “smart cars” park themselves, without us having to decide how hard to apply the brake. We are content to let “smart phones” add appointments to our calendars, without reviewing their work at the end. And very soon, we will be content to let an even smarter car drive us to the supermarket. So isn’t a smart personal assistant in the more distant future making even more important decisions on our behalf just more of the same?

The issue here once again is reliability and predictability. What has happened to AI research recently is that GOFAI, with its emphasis on systems that know a lot, has been supplanted by AML, with an emphasis on systems that are well trained.

The danger is to think that good training is what matters. We might hear things like “Four years without a major accident!” But it is important to recall the long-tail phenomenon from chapter 7. While it might be true that the vast majority of situations that arise will be routine, unexceptional, and well-handled by predictable means, there may be a long tail of phenomena that occur only very rarely. Even with extensive training, an AI system may have no common sense to fall back on in those unanticipated situations. We have to ask ourselves this: *how is the system going to behave when its training fails?* Systems for which we have no good answer to this question should never be left unsupervised, making decisions for themselves.

Beyond evolution

To conclude this speculative chapter, let us return to the topic of superintelligence and speculate even further. Suppose that we

are able to solve all the theoretical and practical problems, clear all the necessary scientific and engineering hurdles, and come to eventually produce a computer system that has all the intelligence of people and more. What would this be like?

Conditioned by AI disaster movies, our first reaction might be that there would be an enormous *conflict* with people, a battle that humans might have a very hard time winning. This is really no different than our imagined encounters with extraterrestrial intelligences (the movie *Close Encounters of the Third Kind* being a notable exception).

I don't buy this outcome, and not because I have an overly rosy picture of how our future will unfold. It's not too much of a stretch to say that in imagining an aggressive AI, we are projecting our own psychology onto the artificial or alien intelligence. We have no experience at all with an advanced intelligence other than our own (at least since the time of our encounter with Neanderthals), and so we can't help but think that an AI will be like us, except more so.

And we are indeed an aggressive species. Like other animals, we end up spending an inordinate amount of time in conflict with other animals, including those of our own species. This is a big part of what survival of the fittest is about, and it appears that we are the product of an evolutionary process that rewards conflict and dominance. We love nature and the natural world, of course, but when we look closely, we can't help but see how much of it involves a vicious struggle for survival, a savage sorting out of the weak from the strong. Human intelligence is also the product of evolution, and so it too is concerned with conflict and dominance.

The problem with applying this analysis to an *artificial* intelligence is that such an intelligence would not be the result of an evolutionary process, unless that is how we decide to build

it. And to believe that we have no choice but to build an AI system to be aggressive (using some form of artificial evolution, say) is to believe that we have no choice but to be this way ourselves.

But we do have a choice, even if we are products of evolution. Recall the Big Puzzle issue from chapter 2. It is a Big Puzzle mistake to assume that the human mind can be fully accounted for in evolutionary terms. When we decry the cruelty of the survival of the fittest, or throw our support behind an underdog, or go out of our way to care for the weak, it is really not worth trying to concoct stories about how this might actually be to increase our evolutionary fitness. The human mind is bigger than its evolutionary aspects. When Humphrey Bogart says to Katharine Hepburn in the movie *The African Queen* that his failings are nothing more than human nature, she replies “Nature, Mr. Allnut, is what we are put in this world to rise above.” Being mindful about what we are doing, and why, gives us the power to be much more than mindless competitors in an evolutionary game. And an artificial intelligence that has not been designed to be ruled by aggression will be able to see—just as we ourselves are able to see in our calmer, more thoughtful moments—why it is good idea not to even play that game at all.

End Notes

This book has presented my thoughts on AI and the mind. It was never intended to be a “scholarly” work on the subject, and the citations here are not going to change things much. Although I have tried to be clear about what I believe and why, I have not spent much time on opposing views, or on the many controversies surrounding AI. My hope is that the reader will use the references below as a starting point for a more balanced study of the issues raised by the topics in this book.

Chapter 1: What Kind of AI?

Although modern AI is only sixty years old, its history is full of twists and turns. The Pamela McCorduck book [80] is a good early history, but the one by Nils Nilsson [89] covers more territory, and can also be found online.

The field of AI is so heterogeneous that it might seem out of the question to have a single comprehensive textbook. And yet the book by Stuart Russell and Peter Norvig [105] is precisely that. It introduces all the major technical directions, including history and philosophical background. Its bibliography alone—thirty pages in a tiny font—is worth the price of admission. But

the book is quite demanding technically, and so is most useful to upper-level university undergraduates. My own textbook [72] presents a much gentler introduction, and is perhaps better suited for a nontechnical audience—at least for the small part of AI that it covers.

For information on recent trends in AI technology, there is, as far as I know, no good book on the subject, and things are happening fast. A reader's best bet is to look for articles in recent newspapers and magazines, a number of which can be found online, including information about Toyota's investment, the OpenAI project, and other similar initiatives. For an introduction to the technological prospects of what I am calling AML, I suggest searching online for "unsupervised learning" or "deep learning" or even "AI technology." (Another option is to search online for major players in the area, such as my colleague Geoffrey Hinton.) To learn about machine learning more broadly, an excellent (though advanced) textbook is [86]. Work on the recognition of cats in images can be found in [66].

The term GOFAI is due to the philosopher John Haugeland [54]. The classic 1958 paper by John McCarthy [79] can be found online. But it also appears in many collections, including [9], [75], [81], and [121], books that include a number of other influential papers.

The Turing Test was first presented in [119] and has been a subject of debate ever since. It is discussed in every AI textbook I know of, and in books such as [108]. The Chinese Room argument can be found in [106], including replies from a number of commentators. My own Summation Room counterargument appears in [70].

Chapter 2: The Big Puzzle

The human mind is such a fascinating subject that it is not surprising to find books on it from a very wide range of perspectives. Three nontechnical books that I have found especially thought-provoking are by Steven Pinker [94], Daniel Kahneman [63], and Daniel Dennett [31]. Other general books that are perhaps more idiosyncratic are by Jerry Fodor [42] (arguing mostly against Pinker), Marvin Minsky [83] (presenting a unique take on the structure of the mind), and a wonderfully engaging book by the reporter Malcolm Gladwell [49] (covering some of the same ground as Kahneman).

Brain research is also extremely active in the popular press, of course. What we can expect to learn from brain imaging (like fMRI) is discussed in [12]. The plasticity of the brain is the subject of [33]. The distributed neural representation (part of my argument about why it may prove so difficult to reverse-engineer a neuron) is described in [58] and [114]. The counterargument that at least some neurons may be representing one single thing (a location, in the case of so-called “place neurons”) is discussed in [91].

As to other accounts of human behavior, its basis in evolution is presented forcefully in [101], and the genetic aspects of evolution are discussed in [25]. The foundation of human behavior in language and symbols is argued in [26], and human language more generally is discussed in [93]. The topic of the evolution of language itself is presented in [64].

Finally, Daniel Dennett’s first presentation of his design stance idea can be found in [28].

Chapter 3: Knowledge and Behavior

The topic of knowledge has been a mainstay of philosophical study since the time of the ancient Greeks. Two collections of papers in this area are [50] and [104]. A more mathematical analysis of knowledge can be found in [57], work that has found further application in computer science [41].

Regarding the relation between knowledge and belief, the classical view (from Plato) is that knowledge is simply belief that is true and justified (that is, held for the right reasons). This view is disputed in a famous paper by Edmund Gettier [45]. As to the propositional attitudes more generally, the main focus in the philosophical literature is on how sentences that mention these attitudes do not appear to follow the usual rules of logic. See, for example, [98].

Philosophers and psychologists sometimes also distinguish implicit from explicit belief [32]. In the example given in the section “Intelligent behavior” in this chapter, the moment Henry discovers that his keys are not in his pocket, although he does not yet explicitly believe they are on the fridge, he implicitly believes it, in the sense that the world that he imagines is one where the keys are there. To put it another way, it is a consequence of what he believes that his keys are on the fridge, even if he does not yet realize it. Mathematical treatments of the two notions can be found in [68], [27], and [40].

The Frederic Bartlett quote on thinking is from [7]. An earlier influential piece by the same author is [6]. Zenon Pylyshyn discusses cognitive penetrability and much else in a wonderful book [97]. Daniel Dennett presents the intentional stance in [29]. (A shorter version with commentary can be found in [30].) The quote by Nicholas Humphrey is from [59].

Finally, Noam Chomsky's competence/performance distinction appeared in [14].

Chapter 4: Making It and Faking It

One of the themes of this chapter is that being intelligent is not really the same thing as being able to fool someone into believing it (for example, in the Imitation Game). But from a purely evolutionary standpoint, the distinction is not so clear. A good case can be made that human intelligence, and language in particular, evolved for the purpose of winning mates in some sort of mental arms race, where pretense and deception get to play a very central role. The primary function of language would be for things like hearsay, gossip, bragging, posturing, and the settling of scores. The rest of what we do with language (on our better days, presumably) might be no more than a pleasant offshoot. See, for example, [35] and [101] for more along these lines.

The ELIZA program by Joseph Weizenbaum is described in [122]. The Oliver Miller interview is drawn from an online post at <http://thoughtcatalog.com/?s=eliza>. Weizenbaum became disheartened by how his work was interpreted by some as a possible jumping point for serious psychological use (for example, in work like [19]), which he discusses in his book [123]. The Loebner competition is described in [16] by Brian Christian, who ended up playing the role of the human in one competition. The EUGENE GOOSTMAN program is described in a number of online news articles, for example, *New Scientist* (25 June 2012) and *Wired* (9 June 2014). The Scott Aaronson interview is drawn from a blog at <http://www.scottaaronson.com/blog/?p=1858>.

My crocodile and baseball examples were first presented in [69] and used again in [73]. The closed-world assumption is explained in [20] and [100].

Winograd schemas were presented by me in [71] and in more detail in [74]. The first example schema (which is due to Terry Winograd) appeared in [125]. A collection of more than one hundred schema questions from various sources can be found online at <https://www.cs.nyu.edu/davise/papers/WS.html>. An actual competition based on Winograd Schemas was announced by Nuance Communications in July 2014, and is scheduled to take place in July 2016. (For more information including rules, see <http://commonsensereasoning.org/winograd.html>.) An alternative test along similar lines is the textual entailment challenge [22].

Chapter 5: Learning with and without Experience

The fact that much of what we come to know is learned through direct experience has given some philosophers pause regarding whether an intelligence without some sort of body is even possible. For example, how could such an intelligence truly understand a word like “hungry” if it cannot connect the word to actual sensations? This is an instance of what is sometimes called the *symbol grounding problem* [53]. How can we understand what a word means if all we have are words that refer to other words that refer to other words? (This “juggling words” issue will come up again in the context of Helen Keller in the next chapter.) Of course the answer suggested by Alan Turing is that we should not even try to answer questions like these. The term “understand” is just too vague and contentious. We should instead ask if it is possible to get an AI program to

behave the way people do regarding the words in question. Just what sort of inapt behavior do we expect the symbol grounding problem to lead to?

How we manage to learn a language at all is still quite mysterious. Specifically, how do we explain how children are able to master the complexities of grammar given the relatively limited data available to them? The issue was called the *poverty of the stimulus* by Noam Chomsky [15]. His (controversial) proposal is that children are born with something called *universal grammar* allowing them to quickly pick up the actual grammar of the language they first hear.

Regarding the learning of behavior, it is interesting that critics of GOF AI like Alan Mackworth [77] and Rodney Brooks [11] focus on how animals (including people) are able to act in the world with real sensors and real effectors without the benefit of language or symbols. The animals appear to acquire some sort of *procedural knowledge* (knowledge formed by doing), as opposed to the more *declarative knowledge* (knowledge that can be expressed in declarative sentences) that GOF AI focuses on. But it is important to keep the Big Puzzle issue in mind on this, and the fact that “action in the world” is a very broad category. It certainly includes things like riding a bicycle and controlling a yo-yo, but also things like caring for a pet canary and collecting rare coins. Clearly both types of knowledge are necessary. See [126] on this issue, and [115] for a neuroscience perspective.

The quote from S. I. Hayakawa on the power of reading is from [55]. (Hayakawa also contrasts learning through experience and learning through language.) The quote from Isaac Newton is from a letter dated 1676. (A collection of his letters is [120].)

Chapter 6: Book Smarts and Street Smarts

This chapter is about the importance of what we learn and pass on through language texts—and this in spite of the fact that we often dismiss it as mere “book knowledge,” some sort of minor quirk in our makeup. Plainly other animals do not write books like we do, and this places a strong limit on the kinds of technology they can develop. On animals and technology, see [109]. For interesting reading on ant supercolonies (regarding aggression and nonaggression), see [46].

For more on young children using language to deal with language issues and problems, see [111]. This is an ability that Don Perlis argues is the very essence of conversational competence [92].

Helen Keller’s life story can be found in [78]. Parts I and II are by Keller herself. Part III is taken from letters and reports by Anne Sullivan (from which the letter in this chapter was quoted). William Rapaport’s analysis of the relevance of Keller’s story to AI work is in [99]. I believe there is still a lot to learn from her about the human mind and the human spirit.

Chapter 7: The Long Tail and the Limits to Training

One of the themes of this chapter is how common sense allows people to deal with situations that are completely unlike those they have experienced before. But Nassim Taleb argues that people, and investors in particular, are actually very bad at dealing with these “black swans” [117]. There is no contradiction here. Common sense tries to deal with new situations as they arise, but investors have the thornier task of somehow appraising all the possible situations that might arise. In deciding to

invest in a stock, an investor has to try to put numbers on all the possible things that could cause the stock to go down. As Taleb argues, people are not very good at weighing in the black swans. For example, in the British National Corpus mentioned in the text, it might seem completely safe to bet against seeing words that have less than a one-in-ten-million chance of appearing, but in fact, there are so many of them that this would be very risky. Information on the British National Corpus can be found online at <http://www.natcorp.ox.ac.uk/>. The observation concerning the large presence of rare words in this corpus is due to Ernie Davis, and further details can be found in his textbook [24], p. 274.

Skill and expertise was the focus of considerable AI research in the 1970s. The principal idea was to try to build so-called expert systems that would duplicate what experts would do in specialized areas using a collection of if-then rules obtained from them in interviews. See [61] and [56], for example. Chess experts are considered in [110] and [103]. The experiment involving a chess expert playing a game of chess while adding numbers was performed by Hubert Dreyfus and Stuart Dreyfus [37]. Here is what they say:

We recently performed an experiment in which an international master, Julio Kaplan, was required rapidly to add numbers presented to him audibly at the rate of about one number per second, while at the same time playing five-second-a-move chess against a slightly weaker, but master level, player. Even with his analytical mind completely occupied by adding numbers, Kaplan more than held his own against the master in a series of games.

Their critique of AI and the expert system approach, as well as other general philosophical observations about experts and novices can be found in [36].

Chapter 8: Symbols and Symbol Processing

This chapter, nominally about symbols and symbol processing, is really an introduction to computer science, an area of study that started with Alan Turing's work on Turing machines [118]. There are researchers who argue for a nonsymbolic form of AI (in [113], for example), but what they are really talking about is symbol processing where the symbols happen to stand for numbers (as in the symbolic algebra example), rather than non-numeric concepts (as in the symbolic logic example).

Those two examples raise interesting questions that are at the heart of computer science. A computational problem can have different ways of solving it with quite different properties, and computer scientists spend much of their time looking at *algorithms*, that is, different ways of solving computational problems [52].

In the case of symbolic algebra, the standard algorithm for solving a system of equations is called *Gaussian elimination* (see [76], for example). It has the property that for n equations with n variables, a solution can be calculated in about n^3 steps. This means that solving systems of equations with thousands and even millions of variables is quite feasible.

But in the case of symbolic logic, perhaps the best algorithm found so far is one called DPLL (see [47], for example). In this case, it is known that there are logic problems with n variables where a solution will require about 2^n steps [51]. (The proof of this involves a variant of DPLL based on the resolution rule [102], mentioned in the text.) This means that solving logic problems with as few as one hundred variables may end up being impractical for even the fastest of computers.

This raises two issues. First, we might wonder if there exists an algorithm that does significantly better than DPLL. It turns out that a mathematically precise version of this question is equivalent to the famous $P = NP$ question, first posed by Stephen Cook in the 1970s [21]. Despite the best efforts of thousands of computer scientists and mathematicians since then, nobody knows the answer to the question. Because of its connection to many other computational problems, it is considered to be the most important open problem in computer science [43].

The second issue concerns the long tail phenomenon from the previous chapter. The way DPLL works is that it performs a systematic search through all the logical possibilities. Interestingly enough, on randomly constructed test cases, the number of steps DPLL takes to do this is almost invariably small. In fact, it appears that the required number of steps on test cases behaves quite like the long-tailed numeric example seen in the section "A numeric example" in this chapter. It is virtually impossible to estimate the number of steps that might be required by DPLL in practice, because as more and more sample test cases are considered, the higher the average value appears to be. For more on this, see [48].

For additional material on the idea of teaching children about following procedures systematically, see [124].

Chapter 9: Knowledge-Based Systems

A biography of Gottfried Leibniz, one of the most fascinating thinkers of all time, is [1]. But much of his thinking is scattered in the over ten thousand letters he wrote, often while he was on the road. A better introduction to his thought can be found in the articles on him in *The Encyclopedia of Philosophy* [38].

Charles Darwin talks about the evolution of the eye in [23]. Here is what he says:

To suppose that the eye with all its inimitable contrivances for adjusting the focus for different distances, for admitting different amounts of light, and for the correction of spherical and chromatic aberration, could have been formed by natural selection seems, I freely confess, absurd in the highest degree.

The evolution of the mind itself is discussed in [34] (with commentary) and in [26].

The knowledge representation hypothesis is quoted from Brian Smith's PhD thesis [112]. Credit for the idea is usually given to John McCarthy, but other AI researchers were clearly on the same track. For Allen Newell and Herb Simon, the emphasis was more on the symbolic aspects, and their version is called the *physical symbol system hypothesis*, which they put this way: "A physical symbol system has the necessary and sufficient means for general intelligent action" [88]. (Marvin Minsky was one of the early AI researchers who saw strong limitations to both the logical and numerical approaches to AI, and argued for a rich amalgamation of approaches [84].)

There are textbooks dedicated to the knowledge representation and reasoning subarea of AI, such as [10] and [3]. There are also conferences in the area held biennially (see <http://kr.org>). Early readings on the subject can be found in [9]. Logic as a unifying theme for AI in general is presented in [44] and [96], and is further discussed in [85] and [60]. Marvin Minsky's quote on logic is taken from [82], p. 262. For a look at reasoning from the psychology side, see [62]. Regarding the advantages of a probabilistic approach to reasoning with degrees of belief, see [90], for example.

On the issue of actually building a large-scale knowledge base, one long-term project along these lines is CYC [67]. (It is difficult

to say precisely what has and has not been achieved with CYC because there has only been limited outside access to the work, and nothing like a controlled scientific study.) Other related but more specialized efforts are HALO [4] and AURA [13] from SRI International, as well as ARISTO [17] from the Allen Institute for AI. For a review of the prospects for automatically extracting knowledge from text on the web, see [39].

Finally, one recent attempt at reconciling the logical approach of GOFAI and the more statistical ones seen in AML can be found in the symposium described online at <https://sites.google.com/site/krr2015/>.

Chapter 10: AI Technology

This chapter touches on only some of the thorny issues regarding the future of AI. A much more comprehensive book on the subject is [5]. The idea of a technological singularity is due to Raymond Kurzweil [65] and further discussed in [107]. The potential dangers of AI are mentioned in interviews with Stephen Hawking (on 2 December 2014, online at <http://www.bbc.com/news/>), Elon Musk (on 8 October 2014, online at <http://www.vanityfair.com/news/>), and Bill Gates (on 28 January 2015, online at <https://www.reddit.com/r/IAmA/comments/>).

The three laws of science-fiction writer Arthur C. Clarke (who was coauthor with Stanley Kubrick of the screenplay for *2001*) were presented in [18] and are as follows:

1. When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.

2. The only way of discovering the limits of the possible is to venture a little way past them into the impossible.
3. Any sufficiently advanced technology is indistinguishable from magic.

The three laws of robotics by science-fiction writer Isaac Asimov were first presented in a story in [2]. (They are said to be quoted from a robotics handbook to be published in 2058.) They are the following:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Many of Asimov's subsequent stories were concerned with what could go wrong with a robot obeying these laws.

Marvin Minsky discusses his involvement with the movie *2001* in an interview in [116]. John McCarthy is quoted by Raj Reddy in a 2000 lecture that is transcribed online at <http://www.r.r.cs.cmu.edu/InfiniteMB.doc>. The Margaret Thatcher quote is from an interview reported in *Woman's Own* magazine (23 September 1987). Alan Turing's early involvement with chess is described in [8]. The story of DEEP BLUE is presented in [87]. For more on the Katharine Hepburn movie quote, and the idea of humans rising above what evolution has given them, see [95].

References

- [1] Eric Aiton. *Leibniz: A Biography*. Boston: Adam Hilger, 1985.
- [2] Isaac Asimov. *I, Robot*. New York: Gnome Press, 1950.
- [3] Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge, UK: Cambridge University Press, 2003.
- [4] Ken Barker, Vinay Chaudhri, Shaw Chaw, Peter Clark, James Fan, David Israel, Sunil Mishra, Bruce Porter, Pedro Romero, Dan Tecuci, and Peter Yeh. A question-answering system for AP chemistry: assessing KR&R technologies. In *Proceedings of KR-2004: The Ninth International Conference on Principles of Knowledge Representation and Reasoning*, Whistler, Canada, June 2004, 488–497.
- [5] James Barrat. *Our Final Invention: Artificial Intelligence and the End of the Human Era*. New York: Thomas Dunne Books, 2013.
- [6] Frederic Bartlett. *Remembering: A Study in Experimental and Social Psychology*. Cambridge, UK: Cambridge University Press, 1932.
- [7] Frederic Bartlett. *Thinking: An Experimental and Social Study*. London: Allen and Unwin, 1958.
- [8] Bertram Bowden, ed. *Faster Than Thought: A Symposium on Digital Computing Machines*. London: Sir Isaac Pitman & Sons, 1953.
- [9] Ronald J. Brachman and Hector J. Levesque, eds. *Readings in Knowledge Representation*. San Francisco: Morgan Kaufmann, 1985.

- [10] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. San Francisco: Morgan Kaufmann, 2004.
- [11] Rodney Brooks. Elephants don't play chess. *Robotics and Autonomous Systems* 6 (1990): 3–15.
- [12] Rita Carter. *Mapping the Mind*. Berkeley: University of California Press, 1998.
- [13] Vinay Chaudhri. Achieving intelligence using prototypes, composition, and analogy. In *Proceedings of AAAI-2015: The Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, January 2015, 4093–4099.
- [14] Noam Chomsky. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press, 1965.
- [15] Noam Chomsky. *Rules and Representations*. Oxford: Basil Blackwell, 1980.
- [16] Brian Christian. *The Most Human Human: What Artificial Intelligence Teaches Us About Being Alive*. New York: Doubleday, 2011.
- [17] Peter Clark. Elementary school science and math tests as a driver for AI: Take the Aristo challenge! In *Proceedings of IAAI-2015: The Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence*, Austin, Texas, March 2015, 4019–4021.
- [18] Arthur C. Clarke. *Profiles of the Future: An Enquiry into the Limits of the Possible*. London: Gollancz, 1962.
- [19] Kenneth Colby, Sylvia Weber, and Franklin Hilf. Artificial paranoia. *Artificial Intelligence* 2 (1971): 1–25.
- [20] Allan Collins, Eleanor Warnock, Nelleke Aiello, and Mark Miller. Reasoning from incomplete knowledge. In *Representation and Understanding: Studies in Cognitive Science*, ed. D. Bobrow and A. Collins. New York: Academic Press, 1975, 35–82.
- [21] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, Shaker Heights, Ohio, 1971, 151–158.

- [22] Ido Dagan, Oren Glickman, and Bernardo Magnini, The PASCAL recognising textual entailment challenge, *Machine Learning Challenges*, Berlin: Springer Verlag, 2006.
- [23] Charles Darwin. *On the Origin of Species*. London: John Murray, 1859.
- [24] Ernest Davis. *Linear Algebra and Probability for Computer Science Applications*. Boca Raton, FL: CRC Press, 2012.
- [25] Richard Dawkins. *The Selfish Gene*. New York: Oxford University Press, 1976.
- [26] Terrence Deacon. *The Symbolic Species: The Co-Evolution of Language and the Brain*. New York: W. W. Norton & Co., 1997.
- [27] James Delgrande. A framework for logics of explicit belief. *Computational Intelligence* 11 (1985): 47–88.
- [28] Daniel Dennett. *Brainstorms: Philosophical Essays on Mind and Psychology*. Cambridge, MA: MIT Press, 1981.
- [29] Daniel Dennett. *The Intentional Stance*. Cambridge, MA: MIT Press, 1987.
- [30] Daniel Dennett. Précis of *The Intentional Stance*. *Behavioral and Brain Sciences* 11 (1988): 495–505.
- [31] Daniel Dennett. *Kinds of Minds: Towards an Understanding of Consciousness*. London: Weidenfeld & Nicolson, 1996.
- [32] Zoltan Dienes and Josef Perner. A theory of implicit and explicit knowledge. *Behavioral and Brain Sciences* 22 (1999): 735–755.
- [33] Norman Doidge. *The Brain That Changes Itself: Stories of Personal Triumph from the Frontiers of Brain Science*. London: Penguin Books, 2007.
- [34] Merlin Donald. Origins of the modern mind: three stages of the evolution of culture and cognition. *Behavioral and Brain Sciences* 16 (1993): 737–791.
- [35] Robin Dunbar. *Grooming, Gossip and the Evolution of Language*. London: Faber and Faber, 1996.

- [36] Hubert Dreyfus. *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, MA: MIT Press, 1992.
- [37] Hubert Dreyfus and Stuart Dreyfus. *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York: Free Press, 1986.
- [38] Paul Edwards, ed. *The Encyclopedia of Philosophy*. New York: Macmillan, 1967.
- [39] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel Weld. Open information extraction from the web. *Communications of the ACM* 51 (2008): 68–74.
- [40] Ronald Fagin and Joseph Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence* 34 (1987): 39–76.
- [41] Ronald Fagin, Joseph Halpern, Yoram Moses, and Moshe Vardi. *Reasoning About Knowledge*. Cambridge, MA: MIT Press, 1995.
- [42] Jerry Fodor. *The Mind Doesn't Work That Way*. Cambridge, MA: MIT Press, 2000.
- [43] Lance Fortnoy. The status of the P versus NP problem. *Communications of the ACM* 52 (2009): 78–86.
- [44] Michael Genesereth and Nils Nilsson. *Logical Foundations of Artificial Intelligence*. Los Altos: Morgan Kaufmann, 1987.
- [45] Edmund Gettier. Is justified true belief knowledge? *Analysis* 23 (1963): 121–123.
- [46] Tatiana Giraud, Jes Pedersen, and Laurent Keller. Evolution of supercolonies: The Argentine ants of southern Europe. *Proceedings of the National Academy of Science* 99 (2002): 6075–6079.
- [47] Carla Gomes, Henry Kautz, Ashish Sabharwal, and Bart Selman. Satisfiability solvers. In *Handbook of knowledge representation: Foundations of Artificial Intelligence*, ed. F. van Harmelen, V. Lifschitz, B. Porter. Amsterdam: Elsevier, 2008, 89–134.

- [48] Carla Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning* 24 (2000): 67–100.
- [49] Malcolm Gladwell. *Blink: The Power of Thinking without Thinking*. New York: Little, Brown and Co., 2005.
- [50] Phillips Griffiths, ed. *Knowledge and Belief*. London: Oxford University Press, 1967.
- [51] Armin Haken. The intractability of resolution. *Theoretical Computer Science* 39 (1985): 297–308.
- [52] David Harel. *Algorithmics: The Spirit of Computing*. Reading, MA, Addison-Wesley, 1987.
- [53] Stevan Harnad. The symbol grounding problem. *Physica D* 42 (1990): 335–346.
- [54] John Haugeland. *Artificial Intelligence: The Very Idea*. Cambridge, MA: MIT Press, 1985.
- [55] Samuel Hayakawa and Alan Hayakawa. *Language in Thought and Action*. New York: Harcourt Brace Jovanovich, 1991.
- [56] Frederick Hayes-Roth, Donald Waterman, and Douglas Lenat. *Building Expert Systems*. Reading, MA: Addison-Wesley, 1983.
- [57] Jaakko Hintikka. *Knowledge and Belief*. Ithaca, NY: Cornell University Press, 1962.
- [58] Geoff Hinton, Jay McClelland, and David Rumelhart. Distributed representations. In *Parallel Distributed Processing*, ed. D. Rumelhart and J. McClelland. Cambridge, MA: MIT Press, 1986, 77–109.
- [59] Nicholas Humphrey. The social function of intellect. In *Growing Points in Ethology*, ed. P. Bateson and R. Hinde. Cambridge, UK: Cambridge University Press, 1976, 303–317.
- [60] David Israel. The role of logic in knowledge representation. *IEEE Computer* 16 (1983): 37–42.

- [61] Peter Jackson. *Introduction to Expert Systems*. Reading, MA: Addison-Wesley, 1990.
- [62] Philip Johnson-Laird, Sangeet Khemlani, and Geoffrey Goodwin. Logic, probability, and human reasoning. *Trends in Cognitive Science* 19 (2015): 201–214.
- [63] Daniel Kahneman. *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux, 2011.
- [64] Christine Kenneally. *The First Word: The Search for the Origins of Language*. New York: Penguin Books, 2007.
- [65] Raymond Kurzweil. *The Singularity Is Near: When Humans Transcend Biology*. London: Viking Penguin, 2005.
- [66] Quoc Le, Marc-Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeffrey Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of ICML 2012: The 29th International Conference on Machine Learning*, Edinburgh, Scotland, June 2012, 81–88.
- [67] Douglas Lenat and Ramanathan Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Boston: Addison-Wesley, 1990.
- [68] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of AAAI-84: The Fourth National Conference on Artificial Intelligence*, August 1984, 198–202.
- [69] Hector J. Levesque. Logic and the complexity of reasoning. *Journal of Philosophical Logic* 17 (1988): 355–389.
- [70] Hector J. Levesque. Is it enough to get the behavior right? In *Proceedings of IJCAI-09: The 21st International Joint Conference on Artificial Intelligence*, Pasadena, California, August 2009, 1439–1444.
- [71] Hector J. Levesque. The Winograd Schema Challenge. In *Proceedings of Commonsense-11: The Tenth International Symposium on Logical Formalizations of Commonsense Reasoning*, Palo Alto, March 2011, 53–58.

- [72] Hector J. Levesque. *Thinking as Computation: A First Course*. Cambridge, MA: MIT Press, 2012.
- [73] Hector J. Levesque. On our best behavior. *Artificial Intelligence* 212 (2014): 27–35.
- [74] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *Proceedings of KR 2012: The Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, Rome, June 2012, 552–561.
- [75] Vladimir Lifschitz, ed. *Formalizing Common Sense: Papers by John McCarthy*. Exeter, UK: Intellect, 1998.
- [76] Marc Lipson and Seymour Lipschutz. *Schaum's outline of theory and problems of linear algebra*. New York: McGraw-Hill, 2001.
- [77] Alan Mackworth. On seeing robots. In *Computer Vision: Systems, Theory and Applications*, ed. A. Basu and X. Li. Singapore: World Scientific Press, 1993, 1–13.
- [78] John Macy, ed. *The Story of My Life*. New York: Doubleday, Page & Co., 1905.
- [79] John McCarthy. Programs with common sense. In *Proceedings of Symposium on the Mechanization of Thought Processes*. National Physical Laboratory, Teddington, England, 1958, 77–84.
- [80] Pamela McCorduck. *Machines Who Think*. 25th anniversary edition. Natick, MA: A K Peters, 2004.
- [81] Marvin Minsky, ed. *Semantic Information Processing*. Cambridge, MA: MIT Press, 1968.
- [82] Marvin Minsky. A framework for representing knowledge. In *Readings in Knowledge Representation*, ed. R. Brachman and H. Levesque. San Francisco: Morgan Kaufmann, 1985, 245–262.
- [83] Marvin Minsky. *The Society of Mind*. New York: Simon & Schuster, 1986.

- [84] Marvin Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine* 12 (1991): 34–51.
- [85] Robert Moore. The role of logic in knowledge representation and commonsense reasoning. In *Proceedings of AAAI-82: The Second National Conference on Artificial Intelligence*, August 1982, 428–433.
- [86] Kevin Murphy. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.
- [87] Monty Newborn and Monroe Newborn. *Deep Blue: An Artificial Intelligence Milestone*. Berlin: Springer Science & Business Media, 2003.
- [88] Allen Newell and Herbert Simon. Computer Science as empirical inquiry: Symbols and search. *Communications of the ACM* 19 (1976): 113–126.
- [89] Nils Nilsson. *The Quest for Artificial Intelligence: A History of Ideas and Achievements*. Cambridge, UK: Cambridge University Press, 2009.
- [90] Mike Oaksford and Nick Chater. *Bayesian Rationality: The Probabilistic Approach to Human Reasoning*. New York: Oxford University Press, 2007.
- [91] Taketoshi Ono, Ryoi Tamura, and Kiyomi Nakamura. The hippocampus and space: are there “place neurons” in the monkey hippocampus? *Hippocampus* 1 (1991): 253–257.
- [92] Don Perlis, Khemdut Purang, and Carl Andersen. Conversational adequacy: mistakes are the essence. *International Journal of Human-Computer Studies*, 48 (1998): 553–575.
- [93] Steven Pinker. *The Language Instinct*. New York: Harper Perennial Modern Classics, 1994.
- [94] Steven Pinker. *How the Mind Works*. New York: W. W. Norton, 1999.
- [95] Steven Pinker. *The Blank Slate*. New York: Viking, 2002.
- [96] David Poole, Alan Mackworth, and Randy Goebel. *Computational Intelligence: A Logical Approach*. New York: Oxford University Press, 1998.

- [97] Zenon Pylyshyn. *Computation and Cognition: Toward a Foundation for Cognitive Science*. Cambridge, MA: MIT Press, 1984.
- [98] Willard Quine. Quantifiers and propositional attitudes. *Journal of Philosophy* 53 (1956): 177–187.
- [99] William Rapaport. How Helen Keller used syntactic semantics to escape from a Chinese Room. *Minds & Machines* 16 (2006): 381–436.
- [100] Raymond Reiter. On closed world data bases. In *Logic and Databases*, ed. H. Gallaire and J. Minker. New York: Plenum Press, 1987, 55–76.
- [101] Matt Ridley. *The Red Queen: Sex and the Evolution of Human Nature*. London: Penguin Books, 1993.
- [102] John Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM* 12 (1965): 23–41.
- [103] Philip Ross. The expert mind. *Scientific American* 295 (2006): 64–71.
- [104] Michael Roth and Leon Galis. *Knowing: Essays in the Analysis of Knowledge*. Lanham, MD: University Press of America, 1984.
- [105] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education, 2010.
- [106] John Searle. Minds, brains and programs. *Behavioral and Brain Sciences* 3 (1980): 417–424.
- [107] Murray Shanahan. *The Technological Singularity*. Cambridge, MA: MIT Press, 2015.
- [108] Stuart Shieber, ed. *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*. Cambridge, MA: MIT Press, 2004.
- [109] Robert Shumaker, Kristina Walkup, and Benjamin Beck. *Animal Tool Behavior: The Use and Manufacture of Tools by Animals*. Baltimore, MD: Johns Hopkins University Press, 2011.
- [110] Herbert Simon and William Chase. Skill in chess. *American Scientist* 61 (1973): 394–403.

- [111] Anne Sinclair, Robert Jarvella, and Willem Levelt, eds. *The Child's Conception of Language*. Berlin: Springer-Verlag, 1978.
- [112] Brian Cantwell Smith. *Reflection and Semantics in a Procedural Language*. Cambridge, MA: PhD thesis, Massachusetts Institute of Technology, 1982.
- [113] Paul Smolensky. Connectionist AI, symbolic AI, and the brain. *Artificial Intelligence Review* 1 (1987): 95–109.
- [114] Paul Smolensky. Analysis of distributed representation of constituent structure in connectionist systems. In *Proceedings of NIPS-87: Neural Information Processing Systems*. Denver, Colorado, November 1988, 730–739.
- [115] Larry Squire. Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory. *Journal of Cognitive Neuroscience* 4 (1992): 232–246.
- [116] David Stork, ed. *HAL's Legacy: 2001's Computer as Dream and Reality*. Cambridge, MA: MIT Press, 1997.
- [117] Nassim Taleb. *The Black Swan: The Impact of the Highly Improbable*. New York: Random House, 2007.
- [118] Alan Turing. On computable numbers, with an application to the *Entscheidungsproblem*. *Proceedings of the London Mathematical Society* 42 (1937): 230–265.
- [119] Alan Turing. Computing machinery and intelligence. *Mind* 59 (1950): 433–460.
- [120] Herbert Turnbull. *The Correspondence of Isaac Newton: 1661–1675*. Volume 1. London: Cambridge University Press, 1959.
- [121] Bonnie Webber and Nils Nilsson, eds. *Readings in Artificial Intelligence*. Los Altos, CA: Morgan Kaufmann, 1981.
- [122] Joseph Weizenbaum. ELIZA. *Communications of the ACM* 9 (1966): 36–45.

- [123] Joseph Weizenbaum. *Computer Power and Human Reason: From Judgment to Calculation*. New York: W. H. Freeman & Co., 1976.
- [124] Jeanette Wing. Computational thinking. *Communications of the ACM* 49 (2006): 33–45.
- [125] Terry Winograd. *Understanding Natural Language*. New York: Academic Press, 1972.
- [126] Terry Winograd. Frame representations and the declarative/procedural controversy. In *Representation and Understanding: Studies in Cognitive Science*, ed. D. Bobrow and A. Collins. New York: Academic Press, 1975, 185–210.

Index

- Aaronson, Scott, 49, 147
African Queen, The, 142
Airplane!, 57
algorithm, 152–153
America's Funniest Home Videos, 39
AML, 2–4, 6, 62–63, 65, 72, 100, 125, 140, 144, 155
analogue computer, 114
Armstrong, Neil, 33
Asimov, Isaac, 138, 156
- Bartlett, Frederic, 31, 146
baseball cap example, 51–52
bears hibernate example, 69–70, 84–86, 125
Berkun, Scott, 80
Big Puzzle, 15–28, 30, 63, 66–67, 142, 145, 149
black swans, 88, 91, 100, 150–151
Blade Runner, 2, 49
Bob guilty example, 104–105, 124
book knowledge, 80–81, 83, 85, 96, 150
- bootstrapping language problem, 68
British National Corpus, 94, 151
Brooks, Rodney, 149
Brown, Molly, 18
- Calculus!, 118
canary example, 71–72, 149
Catch Me If You Can, 46–47
Chinese Room argument, 10–13, 144
Chomsky, Noam, 41, 68, 147, 149
Christian, Brian, 147
Clarke, Arthur C., 131, 155
Close Encounters of the Third Kind, 141
closed-world assumption, 53, 148
cognitively penetrable, 34
common sense, 4–6, 51, 122–123, 130, 134, 140, 150
Cook, Stephen, 153
crocodile example, 51–53, 60, 148
CYC, 154–155

- Darwin, Charles, 121, 126, 154
 Davis, Ernie, xiii, 53, 151
 Deacon, Terrence, 121
 Debussy vs. Ravel thinking, 19
 DEEP BLUE, 132, 156
 degree of belief, 37, 124, 154
 Dennett, Daniel, 27, 38, 145–146
 design stance, 27, 111, 145
 distributed representation, 25, 145
Doctor Strangelove, 138
 DPLL, 152–153
 Dreyfus, Hubert, 95, 151

 ELIZA, 48, 50, 84, 147
 EUGENE GOOSTMAN, 49–50, 147
 evolution, 16, 27, 62, 121, 126, 137, 140–142, 145, 147, 154
 expert system, 151
 expert vs. novice behavior, 95–96, 151

 fMRI, 23, 145
 Fodor, Jerry, 145
Forest Gump, 9

 Gates, Bill, 136, 155
 Gaussian elimination, 152
 Gettier, Edmund, 146
 Gladwell, Malcolm, 145
 GOFAI, 4–7, 61–62, 122, 126, 140, 144, 149, 155
Graduate, *The*, 78

 HAL 9000 computer, 2, 137–138
 Haugeland, John, 144

 Hawking, Stephen, 136, 155
 Hayakawa, S. I., 74, 149
 Henry car key example, 40–41, 123, 146
 Hinton, Geoffrey, 144
 Humphrey, Nicholas, 40, 146

 Imitation Game, 8, 10, 47, 50, 147
 implicit vs. explicit belief, 146
 incarnate word example, 67–68
 intentional stance, 37–38, 101, 115, 119–120, 146

Jerry Springer Show, The, 39
 juggling words critique, 84, 148

 Kahneman, Daniel, 145
 kappanium example, 60–61
 Kasparov, Garry, 132
 Keller, Helen, 81–85, 86, 148, 150
 knowledge base, 119
 knowledge-based system, 39, 115–127, 153
 Kubrick, Stanley, 33–34, 138, 155
 Kurzweil, Raymond, 137, 155

 learning. *See* AML
 Leibniz, Gottfried, 108, 115–119, 153
 Loebner competition, 48, 147
 logic, 104–107, 109, 118–120, 123–124, 146, 152–155
 long tail, 52, 87–100, 140, 150, 153

- machine learning. *See* AML
 machine M example, 19
 Mackworth, Alan, 149
 McCarthy, John, 4, 6, 29–30, 34, 61–62, 119, 122–123, 125, 139, 144, 154, 156
 McCorduck, Pamela, 143
McLaughlin Group, The, 87
 McRone, John, 83
Meaning of Life, The, 40
 Miller, Oliver, 48, 147
 Minsky, Marvin, xiv–xv, 4, 124, 131, 145, 154, 156
Miracle Worker, The, 81
 Monty Python, 40
 Morgenstern, Leora, 53
 movie release year example, 31–35, 45, 97, 115, 126
 Musk, Elon, 1, 136, 155

 neuroscience, 16, 19, 23, 25–26, 61–62, 149
 Newell, Allen, 4, 154
 Newton, Isaac, 75, 115, 149
 Ng, Andrew, 3
 Nilsson, Nils, 143
 Norvig, Peter, 143

Oliver!, 33
 OpenAI, 1, 144

 P = NP question, 153
Paths of Glory, 138
 Perlis, Don, xiii, 150
 Pinker, Steven, 145
 probability. *See* degree of belief
 psychology, 16, 19, 22, 53–54, 61, 141, 154
 Pylyshyn, Zenon, 33–34, 146
 qualia and consciousness, 28
 quantum computer, 114

 Rapaport, William, 84, 150
 representation and reasoning, 37, 122–123, 125–127, 131, 154
 reverse-engineering neurons, 25, 122, 145
 Rome taxi example, 85
 Russell, Stuart, 143

 Searle, John, 10–13
 Shakespeare, 16, 74
 Simon, Herb, 4, 154
 Smith, Brian, 119, 154
 source vs. object code, 24–25
 street knowledge, 80, 85–86
 Sullivan, Anne, 81, 150
 symbol grounding problem, 148–149
 symbolic species, 121

 tabula rasa computer, 127
 Taleb, Nassim, 88, 150–151
 Thatcher, Margaret, 135, 156
 Tommy and Suzy age example, 102–103, 117
 traffic-related deaths, 90
 Turing machine, 111–114, 152
 Turing Test, 7, 9–12, 30, 45, 47–48, 50, 59

- Turing, Alan, 7–11, 13, 47, 50, 59,
84, 108–111, 132, 148, 152,
156
- 2001: A Space Odyssey*, xiv, 2,
31–35, 45, 61, 97, 115, 126,
131, 137–138, 155–156
- universal grammar, 149
- use vs. mention distinction, 83
- Weizenbaum, Joseph, 48, 50, 147
- whiteout, 92–93, 95–97
- Winograd schema, 54–63, 97–99,
125–126, 129, 148
- Winograd Schema Test, 59, 84
- Winograd, Terry, 56, 148
- word as symbol, 82–83, 108–109,
121, 125