4

If Computers Are So Smart, How Come They Can't Read?

SAMANTHA: So how can I help you?
THEODORE: Oh, it's just more that everything feels disorganized, that's all.
SAMANTHA: You mind if I look through your hard drive?
THEODORE: Um . . . okay.
SAMANTHA: Okay, let's start with your e-mails. You have several thousand e-mails regarding *LA Weekly,* but it looks like you haven't worked there in many years.
THEODORE: Oh, yeah. I think I was just saving those cause, well I thought maybe I wrote something funny in some of them. But . . .
SAMANTHA: Yeah, there are some funny ones. I'd say that there are about eighty-six that we should save, we can delete the rest.

—*HER* (2013), WRITTEN AND DIRECTED BY SPIKE JONZE

Wouldn't it be nice if machines could understand us as well as Samantha (the "operating system" voiced by Scarlett Johansson in the science fiction movie *Her*) understands Theodore? And if they could sort through our emails in an instant, pick out whatever we need, and filter out the rest?

If we could give computers one gift that they don't already have, it would be the gift of understanding language, not just so they could help organize our lives, but also so they could help humanity on some of our greatest challenges, like distilling vast scientific literatures, that individual humans can't possibly hope to keep up on.

In medicine, seven thousand papers are published every day. No doctor or researcher can possibly read them all, and that is a serious impediment to progress. Drug discovery gets delayed in part because lots of information is locked up in literature that nobody has time to read. New treatments sometimes don't get applied because doctors don't have time to read and discover them. AI programs that could automatically synthesize the vast medical literature would be a true revolution.

Computers that could read as well as PhD students but with the raw computational horsepower of Google would revolutionize science, too. We would expect advances in every field, from mathematics to climate science to material science. And it's not just science that would be transformed. Historians and biographers could instantly find out everything that has been written about an obscure person, place, or event. Writers could automatically check for plot inconsistencies, logical gaps, and anachronisms.

Even much simpler abilities could be enormously helpful. Currently the iPhone has a feature such that when you get an email message that sets up an appointment, you can click on it and the iPhone will add it to your calendar. That's really handy—when it works right. Often, it doesn't; the iPhone adds the appointment, not on the day you have in mind, but perhaps on some other day mentioned in the email. If you don't catch the mistake when the iPhone makes it, it can be a disaster.

Someday, when machines really can read, our descendants will wonder how we ever got by without synthetic readers, just as we wonder how earlier generations managed without electricity.

At TED, in early 2018, the noted futurist and inventor Ray Kurzweil, now working at Google, announced his latest project, Google Talk to Books, which promised to use natural language understanding to "provide an entirely new way to explore books." *Quartz* dutifully hyped it as "Google's astounding new search tool [that] will answer any question by reading thousands of books."

As usual, the first question to ask is "What does the program actually do?" The answer was that Google has indexed the sentences in 100,000 books, ranging from *Thriving at College* to *Beginning Programming for Dummies* to *The Gospel According to Tolkien,* and developed an efficient method for encoding the meanings of sentences as sets of numbers known as vectors. When you ask a question, it uses those vectors to find the twenty sentences in the database that have vectors that are most similar. The system has no idea what you are actually asking.

Just from knowing the input to the system it should be immediately obvious that the claim in the *Quartz* article that Talk to Books "will answer any question" can't be taken literally; 100,000 books may sound like a large number but it's a tiny fraction of the more than one hundred million that have been published. Given what we saw earlier about how much deep learning draws on correlation rather than genuine comprehension, it should come as no surprise that many of the answers were dubious. If you asked it some particular detail of a novel, for instance, you should reasonably expect a reliable answer. Yet when we asked "Where did Harry Potter meet Hermione Granger?" none of the twenty answers was from *Harry Potter and the Sorcerer's Stone,* and none addressed the question itself: where the meeting took place. When we asked "Were the Allies justified in continuing to blockade Germany after World War I?" it found no results that even mentioned the blockade. Answering "any question" is a wild exaggeration.

And when answers weren't spelled out directly in a phrase in the indexed text, things often ran amiss. When we asked "What were the seven Horcruxes in *Harry Potter*?" we didn't even get an answer with a list, perhaps because none of the many books that discuss Harry Potter enumerates the Horcruxes in a single list. When we asked "Who was the oldest Supreme Court justice in 1980?" the system failed, even though you as a human can go to any online list of the Supreme Court justices (for instance, in Wikipedia) and in a couple of minutes figure out that it was William Brennan. Talk to Books stumbled again, precisely because there was no sentence

spelling it out in full—"The oldest judge on the Supreme Court in 1980 was William Brennan"—in any of the 100,000 books, and it had no basis for making inferences that extended beyond the literal text.

The most telling problem, though, was that we got totally different answers depending on how we asked the question. If we asked Talk to Books, "Who betrayed his teacher for 30 pieces of silver?," a pretty famous incident in a pretty famous story, of the twenty answers, only six correctly identified Judas. (Curiously, nine of the answers had to do with the much more obscure story of Micah the Ephraimite, in Judges 17.) But things got even worse as we strayed from the exact wording of "pieces of silver." When we asked Talk to Books the slightly less specific "Who betrayed his teacher for 30 coins?" Judas turned up in only 10 percent of the answers. (The top-ranked answer was both irrelevant and uninformative: "It is not known who Jingwan's teacher was.") And when we again slightly reworded the question, this time changing "betrayed" to "sold out," yielding "Who sold out his teacher for 30 coins?" Judas disappeared from the top twenty results altogether.

The further we moved from exactly matching a set of words, the more lost the system became.

The machine-reading systems of our dreams, when they arrive, would be able to answer essentially any reasonable question about what they've read. They would be able to put together information across multiple documents. And their answers wouldn't just consist of spitting back underlined passages, but of *synthesizing* information, whether that's lists of Horcruxes that never appeared in the same passage, or the sort of pithy encapsulations that you would expect of a lawyer assembling precedents across multiple cases, or a scientist formulating a theory that explains observations collected across multiple papers. Even a first grader can create a list of all the good guys and bad guys that appear in a series of children's books.

Just as a college student writing a term paper can bring together ideas from multiple sources, cross-validating them and reaching novel conclusions, so too should any machine that can read.

But before we can get machines to synthesize information rather than merely parroting it, we need something much simpler: machines that can reliably comprehend even basic texts.

That day isn't here yet, however excited some people seem to be about AI. To get a sense for why robust machine reading is actually still a fairly distant prospect, it helps to appreciate—in detail—what is required even to comprehend something relatively simple, like a children's story.

Suppose that you read the following passage from *Farmer Boy,* a children's book by Laura Ingalls Wilder (author of *Little House on the Prairie*). Almanzo, a nine-year-old boy, finds a wallet (then referred to as a "pocketbook") full of money dropped in the street. Almanzo's father guesses that the "pocketbook" (i.e., wallet) might belong to Mr. Thompson, and Almanzo finds Mr. Thompson at one of the stores in town.

> Almanzo turned to Mr. Thompson and asked, "Did you lose a pocketbook?"
>
> Mr. Thompson jumped. He slapped a hand to his pocket, and fairly shouted.
>
> "Yes, I have! Fifteen hundred dollars in it, too! What about it? What do you know about it?"
>
> "Is this it?" Almanzo asked.
>
> "Yes, yes, that's it!" Mr. Thompson said, snatching the pocketbook. He opened it and hurriedly counted the money. He counted all the bills over twice. . . .
>
> Then he breathed a long sigh of relief and said, "Well, this durn boy didn't steal any of it."

A good reading system should be able to answer questions like these:

Why did Mr. Thompson slap his pocket with his hand?
Before Almanzo spoke, did Mr. Thompson realize that he had lost his wallet?
What is Almanzo referring to when he asks "Is this it?"
Who almost lost $1,500?
Was all of the money still in the wallet?

All of these questions are easy for people to answer, but no AI yet devised could reliably handle queries like these. (Think about how troubled Google Talk to Books would have been by them.)*

At its core, each of these questions requires a reader (human or otherwise) to follow a chain of inferences that are only implicit in the story. Take the first question. Before Almanzo speaks, Mr. Thompson doesn't know he has lost the wallet and assumes that he has the wallet in his pocket. When Almanzo asks him whether he has lost a wallet, Thompson realizes he might in fact have lost his wallet. It is to test this possibility—the wallet might be lost—that Thompson slaps his pocket. Since the wallet isn't where he usually keeps it, Thompson concludes that he has lost his wallet.

When it comes to complex chains of reasoning current AI is at a loss. Such chains of reasoning often demand that the reader put together an impressive range of background knowledge about people and objects, and more generally about how the world works, and

* The Allen Institute for Artificial Intelligence has a website, ai2.org, where you can try out near-state-of-the-art models on tests like these. For example, on November 16, 2018, we entered the Almanzo story into the then leading model available on the site and asked *How much money was in the pocketbook?, What was in the pocketbook?, Who owns the pocketbook?,* and *Who found the pocketbook?* The first and third were answered correctly; the second received an incoherent answer ("counted the money"); and the last was simply wrong (Mr. Thompson, rather than Almanzo). Unreliable results like these are typical of the contemporary state of the art.

no current system has a broad enough fund of general knowledge to do this well.

Take some of the kinds of knowledge you probably drew on just now, automatically, without even being aware of it, as you digested the story of Almanzo and the wallet:

- People can drop things without realizing it. *This is an example of knowledge about the relation between events and people's mental states.*
- People often carry their wallets in their pockets. *This is an example of knowledge about how people typically use certain objects.*
- People often carry money in their wallets, and money is important to them, because it allows them to pay for things. *This is an example of knowledge about people, customs, and economics.*
- If people assume that something important to them is true, and they find out that it might not be true, then they often urgently try to verify it. *This is an example of knowledge about the kinds of things that are psychologically important to people.*
- You can often find out whether something is inside your pocket by feeling the outside of the pocket. *This is an example of how different types of knowledge may be combined. Here knowledge about how different objects (hands, pockets, wallets) interact with one another is combined with knowledge about how the senses work.*

The reasoning required for the other questions is equally rich. To answer the third question, "What is Almanzo referring to when he asks 'Is this it?,' " the reader has to understand something about language, as well as about people and objects, concluding that a reasonable antecedent to the words "this" and "it" could be the wallet, but (rather subtly) that "this" refers to the wallet that Almanzo is holding, while "it" refers to the wallet that Mr. Thompson has lost.

Happily, the two (what Almanzo is holding and what Mr. Thompson has lost) turn out to be the same.

To cope with even a simple passage, one's knowledge of people, objects, and language must be deep, broad, and flexible; if circumstances are even slightly different, we need to adapt accordingly. We should not expect equal urgency from Mr. Thompson if Almanzo said that he had found Almanzo's grandmother's wallet. We find it plausible that Mr. Thompson could have lost his wallet without knowing it, but we would be surprised if he was unaware of having his wallet taken after he was mugged at knifepoint. Nobody has yet been able to figure out how to get a machine to reason in such flexible ways. We don't think this is impossible, and later we sketch some of the steps that would need to be taken, but the reality for now is that what is required vastly outstrips what any of us in the AI community have yet managed to accomplish. Google Talk to Books wouldn't even be close (nor would the readers from Microsoft and Alibaba that we mentioned at the very beginning of the book).

Fundamentally, there is a mismatch between what machines are good at doing now—classifying things into categories—and the sort of reasoning and real-world understanding that would be required in order to capture this mundane yet critical ability.

Virtually anything you might read poses similar challenges. There's nothing particularly special about the Wilder passage. Here's a brief example from *The New York Times,* April 25, 2017.

> Today would have been Ella Fitzgerald's 100th birthday.
>
> One New Yorker, Loren Schoenberg, played saxophone alongside the "First Lady of Song" in 1990, near the very end of her career. He compared her to "a vintage bottle of wine" . . .

Anyone can easily answer questions taken pretty directly from the text (*what instrument did Loren Schoenberg play?)*—but many

questions would demand a kind of inference that entirely eludes most current AI systems.

- Was Ella Fitzgerald alive in 1990?
- Was she alive in 1960?
- Was she alive in 1860?
- Did Loren Schoenberg ever meet Ella Fitzgerald?
- Does Schoenberg think that Fitzgerald was an alcoholic beverage?

"He compared her to 'a vintage bottle of wine.'"

Answering the first, second, and third questions involves reasoning that Ella was born on April 25, 1917, since April 25, 2017, was her 100th birthday, and then incorporating common knowledge such as facts that

- People are alive during their career, so she was alive in 1990.
- People are alive at all times between their birth and their death, and no times before their birth or after their death. So

> Fitzgerald must have been alive in 1960 and not yet alive in 1860.

Answering the fourth question involves reasoning that playing music alongside someone generally involves meeting them, and inferring that Fitzgerald is "the First Lady of Song," even though that identity is never quite made explicit.

Answering the fifth question requires reasoning about what sorts of things people typically envision when they make comparisons, and knowing that Ella Fitzgerald was a person and that people cannot turn into beverages.

Pick a random article in the newspaper, or a short story, or novel of any length, and you are sure to see something similar; skilled writers don't tell you everything, they tell you what you need to know, relying on shared knowledge to fill in the gaps. (Imagine how dull Wilder's story would be if she had to tell you that people keep their wallets in their pockets, and that people sometimes attempt to detect the presence or absence of small physical objects by reaching for them with their hands, through their pockets.)

In an earlier era, a bunch of AI researchers actually tried hard to solve these problems. Peter Norvig, now a director of research at Google, wrote a provocative doctoral thesis on the challenges in getting machines to understand stories. More famously, Roger Schank, then at Yale, came up with a series of insightful examples of how machines could use "scripts" to understand what happens when a customer goes to a restaurant. But understanding a story requires much more complex knowledge and many more forms of knowledge than scripts, and the problem of formulating and collecting all that knowledge was daunting. In time, the field gave up, and researchers started working on other, more approachable problems—such as web search and recommendation engines—none of which has brought us significantly closer to general AI.

Web search has of course nonetheless changed the world; it's one of AI's biggest success stories. Google Search, Bing, and others are amazingly powerful and fantastically useful pieces of engineering, powered by AI, that almost instantaneously find matches among billions of web documents.

What is perhaps surprising is that, while they are all powered by AI, they have almost nothing to do with the kind of automated, synthetic machine reading we have been calling for. We want machines that can understand what they are reading. Search engines don't.

Take Google Search. There are two basic ideas in the Google algorithm, one old, and one that Google pioneered. Neither depends on having the system comprehend documents. The first, older idea had been used in document-retrieval programs since the early 1960s, long before Google or the web: you match words in the query against words in the document. Want to search for recipes involving cardamom? No problem—just find all the websites containing the words "recipe" and "cardamom." No need to understand that cardamom is a spice, no need to understand what it smells like, or tastes like, nor to know anything about the history of how it is extracted from pods or which cuisines most often use it. Want to find instructions on building airplanes? Just match a few words like "model," "airplane," and "how to," and you will get lots of useful hits, even if the machine has no idea what an airplane actually is, let alone what lift and drag are or the reasons you would probably rather fly commercial than get a ride on a scale model.

The second, more innovative idea—the famous PageRank algorithm—was the idea that a program could use the collective wisdom of the web in judging which web pages were high quality by seeing which pages had gotten many links, particularly links from other high-quality pages. That insight catapulted Google above all the other web search engines of the time. But matching words does not have much to do with *understanding* texts, nor does counting links that are inbound from other pages.

The reason that Google Search does as well as it does *without*

any kind of sophisticated reading is that little precision is required. The search engine does not need to read deeply in order to discern whether some treatise on presidential powers leans to the left or the right; the user can figure that out. All Google Search has to figure out is whether a given document is about the right general topic. One can usually get a pretty good idea of the subject of a document just by looking at the words and short phrases that are in it. If it has "president" and "executive privilege," the user probably will be happy to have the link; if it's about the Kardashians, it's probably not relevant. If a document mentions "George," "Martha," and the "Battle of Yorktown," Google Search can guess that the document is about George Washington, even if it knows nothing about marriage or revolutionary wars.

Google is not always so superficial. Sometimes it manages to interpret queries and give back fully formed answers rather than just long lists of links. That's a little closer to reading, but only a little, because Google is generally only reading the queries, not the documents themselves. If you ask "What is the capital of Mississippi?" Google correctly parses your question and looks up the answer ("Jackson") in a table that's been constructed in advance. If you ask "How much is 1.36 euros in rupees," parsing is again correct and the system can, after consulting a different table (this time with exchange rates), correctly calculate that "1.36 euros = 110.14 Indian rupees."

For the most part, when Google returns an answer of this sort, it's usually reliable (the system presumably only does so when its indicators suggest the answers are likely to be correct). But it's still far from perfect, and the errors it makes give a good hint about what's going on. For example, in April 2018, we asked Google Search "Who is currently on the Supreme Court?" and got back the rather incomplete answer "John Roberts," just one member among nine. As a bonus, Google provided a list of seven other justices "people also search for": Anthony Kennedy, Samuel Alito, Clarence Thomas,

Stephen Breyer, Ruth Bader Ginsburg, and Antonin Scalia. All these people have of course been on the court, but Scalia was deceased. Scalia's successor, Neil Gorsuch, and recent appointees Elena Kagan and Sonia Sotomayor were absent from Google's list. It's almost as if Google had missed the word "currently" altogether.

Going back to our earlier point about synthesis, the ultimate machine-reading system would compile its answer by reading Google News and updating its list when there are changes; or at least by consulting Wikipedia (which humans update fairly regularly) and extracting the current judges. Google doesn't seem to be doing that. Instead, as best we can tell, it is simply looking at statistical regularities (Alito and Scalia come up in many searches for justices), rather than genuinely reading and comprehending its sources.

To take another example, we tried asking Google, "When was the first bridge ever built?" and got back the following at the top of the results:

> Iron and Steel bridges are used today and most of the worlds [*sic*] major rivers are crossed by this type. The picture shows the first iron bridge in the world. It was built in Telford in 1779 by Abraham Darby (the third) and was the first large structure in history to be constructed from iron.

The words "first" and "bridge" match our query, but the first bridge ever built wasn't iron, and "first iron bridge" doesn't equal "first bridge"; Google was off by thousands of years. And the fact is, more than a decade after they were introduced, searches in which Google reads the question and gives a direct answer still remain very much in the minority. When you get links rather than answers, it's generally a sign that Google is just relying on things like keywords and link-counting, rather than genuine comprehension.

Companies like Google and Amazon are of course constantly improving their products, and it's easy enough to hand-code a system to correctly list the current set of Supreme Court justices; small incremental improvements will continue. What we don't see on the

horizon is any *general* solution to the many kinds of challenges we have raised.

A few of years ago, we saw a clever Facebook meme: a picture of Barack Obama with the caption "Last year you told us you were 50 years old; now you say you are 51 years old. Which is it, Barack Obama?" Two different utterances, spoken at different times, can both be true. If you're human, you get the joke. If you are a machine doing little more than keyword matching, you are lost.

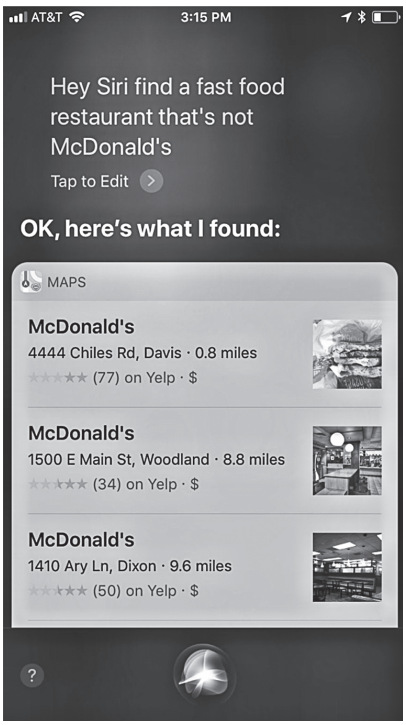
What about speech-driven "virtual assistants" such as Siri, Cortana, Google Assistant, and Alexa? On the plus side, they often take action rather than merely giving you lists of links; unlike Google Search, they have been designed from the beginning to interpret user queries not as collections of random keywords, but as actual questions. But after several years, all are hit-or-miss, effective in some domains and weak in others. For example, they are all pretty good at "factoid" questions—"Who won the World Series in 1957?"; each of them also has pockets of clear strength. Google Assistant is good at giving directions and buying movie tickets. Siri is good at giving directions and at making reservations. Alexa is good at math, pretty decent at telling prewritten jokes, and (not surprisingly) good at ordering things from Amazon.

But outside their particular areas of strength, you never know what to expect. Not long ago, the writer Mona Bushnell tried asking all four programs for directions to the nearest airport. Google Assistant gave her a list of travel agents. Siri gave her directions to a seaplane base. Cortana gave her a list of airline ticket websites, such as Expedia. On a recent drive that one of us took, Alexa scored 100 percent on questions like *Is Donald Trump a person?, Is an Audi a vehicle?,* and *Is an Edsel a vehicle?,* but bombed on questions like *Can an Audi use gas?, Can an Audi drive from New York to California?,* and *Is a shark a vehicle?*

Or take this example, sent to Gary recently on Twitter: a screenshot of someone's effort to ask Siri for "the nearest fast food restau-

rant that was not McDonald's." Siri dutifully came up with a list of three nearby restaurants, and all served fast food—but every one of them was a McDonald's; the word "not" had been entirely neglected.

WolframAlpha, introduced in 2009 to much hype as "the world's first computational knowledge engine," is no better. It has enormous built-in databases of all kinds of scientific, technological, mathematical, census, and sociological information, and a collection of techniques for using this information to answer questions, but its capacity to put all that information together is still spotty.



Misunderstanding "Find a fast food restaurant that is not McDonald's"

Its strength is mathematical questions like "What is the weight of a cubic foot of gold?" "How far is Biloxi, Mississippi, from Kolkata?" and "What is the volume of an icosahedron with edge length of 2.3 meters?" ("547 kg," "8781 miles," and "26.5 m^3," respectively.)

But the limits of its understanding aren't hard to reach. If you ask "How far is the border of Mexico from San Diego?" you get "1144 miles," which is totally wrong. WolframAlpha ignores the word "border," and instead returns the distance from San Diego to the geographic center of Mexico. If you slightly rephrase the question about icosahedron volume by replacing the words "with edge length 2.3 meters" with "whose edges are 2.3 meters long," WolframAlpha no longer recognizes that the question is about volume; all you get back is generic information that icosahedrons have 30 edges, 20 vertices, and 12 faces, without any mention of volume. WolframAlpha

Misunderstanding "How far is the border of Mexico from San Diego?"

can tell you when Ella Fitzgerald was born and when she died; but if you ask it "Was Ella Fitzgerald alive in 1960?," it wrongly interprets the question as "Is Ella Fitzgerald alive?" and answers "No."

But wait, you say, what about Watson, which was so good at answering questions that it beat two human champions at *Jeopardy!* True, but unfortunately Watson is not nearly as generally powerful as it might seem. Almost 95 percent of *Jeopardy!* answers, as it turns out, are titles of Wikipedia pages. Winning at *Jeopardy!* is often just a matter of finding the right article. It's a long way from that sort of intelligent information retrieval to a system that can genuinely think and reason. Thus far, IBM hasn't even turned Watson into a robust virtual assistant. When we looked recently on IBM's web page for such a thing, all we could find was a dated demo of Watson Assistant that was focused narrowly on simulated cars, in no way on a par with the more versatile offerings from Apple, Google, Microsoft, or Amazon.

Virtual assistants like Siri and Alexa are, to be sure, starting to become useful, but they have a long way to go. And, critically, just

as with Google Search, there is precious little synthesis going on. As far as we can tell, very few of them ever attempt to put together information in flexible ways from multiple sources, or even from a single source with multiple sentences, the way you did earlier when you read about Almanzo and about Ella Fitzgerald.

The truth is that *no* current AI system can duplicate what you did in those instances, integrating a series of sentences and reconstructing both what is said and what is not. If you are following what we are saying, you are human, not a machine. Someday you might be able to ask Alexa to compare *The Wall Street Journal*'s coverage of the president with *The Washington Post*'s coverage, or ask if your family doctor might have missed anything in your latest charts, but for now that's just fantasy. Better stick to asking Alexa for the weather.

What are we left with? A hodgepodge of virtual assistants, often useful, never fully reliable—not one of which can do what we humans do every time we read a book. Six decades into the history of AI, computers are still functionally illiterate.

Deep learning is not going to solve this problem, nor is the closely associated trend of "end-to-end" learning, in which an AI is trained to convert inputs directly into outputs, without any intermediate subsystems. For instance, whereas a traditional approach to driving would break things into subsystems like perception, prediction, and decision-making (perhaps using deep learning as an element in some of those subsystems), an end-to-end system would dispense with the subsystems and instead build a car-driving system that takes in camera images as input and returns, as its outputs, adjustments for acceleration and steering—without any intermediate subsystems for determining where different objects are and how they are moving, what the other drivers can be expected to do and not do, and so forth.

When it works, it can be very effective, and more straightforward to implement than more structured alternatives; end-to-end systems

often require comparatively little human labor. And sometimes they are the best available solution. As the *New York Times Magazine* article on Google Translate made clear, end-to-end deep learning systems have greatly improved the state of the art in machine translation, superseding earlier approaches. Nowadays, if you want to build a program to, say, translate between French and English, you would begin by collecting an enormous corpus of documents that exist in both French and English versions, called *bitexts* (pronounced "bye-texts"), like the proceedings of the Canadian parliament, which by law must be published in both languages. From data like that, Google Translate can automatically learn the correspondences between the English words and phrases and their French counterparts, without any prior knowledge about French or English, or any prior knowledge about the intricacies of French grammar. Even skeptics like us are amazed.

The trouble is, one size doesn't fit all. Machine translation turns out to be an unusually good fit for end-to-end methods, partly because of the ready availability of large amounts of relevant data, and partly because there is generally a more-or-less clear correspondence between the English words and the French words. (Most of the time, the right French word is one of the options that you would find in a French-English dictionary, and most of the time the relation between the order of words in the two languages follows fairly standard patterns.) But many other aspects of language understanding are a much poorer fit.

Answering questions is much more open-ended, in part because the words in the correct answer to a question may bear no obvious relation to the words of the text. Meanwhile there is no database of questions and answers of a size comparable to the French-English parliamentary proceedings. Even if there were, the universe of questions and answers is so vast that any database would be but a tiny sample of all the possibilities. As explained earlier, this poses serious problems for deep learning: the further a deep learning system has to veer from its training set, the more trouble it gets into.

And, truth be told, even in machine translation, end-to-end

approaches still have their limits. They are often (but not always) fine for getting the gist, but matching words and phrases and so forth is not always enough. When getting the right translation hinges on a deeper understanding, the systems break down. If you give Google Translate the French sentence *"Je mange un avocat pour le déjeuner,"* which actually means "I eat an avocado for lunch," the translation you get is "I eat a lawyer for lunch." The French word *avocat* means both "avocado" and "lawyer," and, since people write much more often about lawyers than avocados (particularly in the proceedings of the Canadian parliament), Google Translate goes with the more frequent meaning, sacrificing sense for statistics.

In a wonderful article in *The Atlantic,* Douglas Hofstadter described the limitations of Google Translate:

> We humans know all sorts of things about couples, houses, personal possessions, pride, rivalry, jealousy, privacy, and many other intangibles that lead to such quirks as a married couple having towels embroidered "his" and "hers." Google Translate isn't familiar with such situations. Google Translate isn't familiar with situations, period. It's familiar solely with strings composed of words composed of letters. It's all about ultra-rapid processing of pieces of text, not about thinking or imagining or remembering or understanding. It doesn't even know that words stand for things.

For all the progress that's been made, most of the world's written knowledge remains fundamentally inaccessible, even if it is digital and online, because it is in a form that machines don't understand. Electronic health records, for example, are filled with what is often called *unstructured text,* things like doctor's notes, emails, news articles, and word-processing documents that don't fit neatly into a table. A true machine-reading system would be able to dive in, scouring doctor's notes for important information that is captured in blood tests and admission records. But the problem is so far beyond

what current AI can do that many doctors' notes are never read in detail. AI tools for radiology are starting to be explored; they are able to look at images and to distinguish tumors from healthy tissue, but we have no way yet to automate another part of what a real radiologist does, which is to connect images with patient histories.

The ability to understand unstructured text is for now a significant bottleneck in a huge range of potential commercial applications of AI. We can't yet automate the process of reading legal contracts, scientific articles, or financial reports, because each consists in part of the kind of text that AI still can't grasp. Although current tools automatically pull some basic information out of even the most difficult text, a large part of the content is typically left behind. Fancier and fancier versions of text matching and link counting help—a little—but they simply don't get us to programs that can genuinely read and understand.

Of course, the situation is no better for spoken language understanding (sometimes called dialogue understanding). Even greater challenges would arise for a computerized doctor's assistant that tried to translate speech into medical notes (so that doctors could spend more time with patients and less on their laptops). Consider this simple bit of dialogue, sent to us by Dr. Vik Moharir:

> DOCTOR: Do you get chest pain with any sort of exertion?
> PATIENT: Well I was cutting the yard last week and I felt like an elephant was sitting on me. [Pointing to chest]

To a person, it's obvious that the answer to the doctor's question is "yes"; cutting the yard is in the taxonomy of exertions, and we infer that the patient experienced pain from our knowledge that elephants are heavy and being crushed by heavy things is painful. We also automatically infer that the word "felt" is being used figuratively rather than literally, given the amount of damage an actual elephant would inflict. To a machine, unless there's been a lot of specific talk of elephants before, it's probably just some rambling about large mammals and yard work.

How did we get into this mess?

Deep learning is very effective at learning correlations, such as correlations between images or sounds and labels. But deep learning struggles when it comes to understanding how objects like sentences relate to their parts (like words and phrases). Why? It's missing what linguists call compositionality: a way of constructing the meaning of a complex sentence from the meaning of its parts. For example, in the sentence *The moon is 240,000 miles from the earth,* the word *moon* means one specific astronomical object, *earth* means another, *mile* means a unit of distance, *240,000* means a number, and then, by virtue of the way that phrases and sentences work compositionally in English, *240,000 miles* means a particular length, and the sentence *The moon is 240,000 miles from the earth* asserts that the distance between the two heavenly bodies is that particular length.

Surprisingly, deep learning doesn't really have any direct way of handling compositionality; it just has lots and lots of isolated bits of information known as features, without any structure. It can learn that dogs have tails and legs, but it doesn't know how they relate to the life cycle of a dog. Deep learning doesn't recognize a dog as an animal composed of parts like a head, a tail, and four legs, or even what an animal is, let alone what a head is, and how the concept of head varies across frogs, dogs, and people, different in details yet bearing a common relation to bodies. Nor does deep learning recognize that a sentence like *The moon is 240,000 miles from the earth* contains phrases that refer to two heavenly bodies and a length.

To take another example, when we asked Google Translate to translate "The electrician whom we called to fix the telephone works on Sundays" into French, the answer we got was *L'électricien que nous avons appelé pour réparer le téléphone fonctionne le dimanche*. If you know French, you know that's not quite right. In particular, the word *works* has two translations in French: *travaille,* which means *labors,* and *fonctionne,* which means *functions properly*. Google has used the word *fonctionne,* rather than *travaille,* not

grasping, as a human would, that "works on Sundays" is something that in context refers to the electrician, and that if you are talking about a person working you should be using the verb *travaille*. In grammatical terms, the subject of the verb *works* here is *electrician*, not *telephone*. The meaning of the sentence as a whole is a function of how the parts are put together, and Google doesn't really get that. Its success in many cases fools us into thinking that the system understands more than it really does, but the truth (once again illustrating the illusory progress gap) is that there is very little depth to its translations.*

A related and no less critical issue is that deep learning has no good way to incorporate background knowledge, which is something that we saw earlier, in chapter 3. If you are learning to relate an image to a label, it doesn't matter how you do it. As long as it works, nobody cares about the internal details of the system because all that matters is that you get the right label for a given image. The whole task is often relatively isolated from most of the rest of what you know.

Language is almost never like that. Virtually every sentence that we encounter requires that we make inferences about how a broad range of background knowledge interrelates with what we read. Deep learning lacks a direct way of representing that knowledge, let

* When we first wrote this sentence, in August 2018, Google Translate made the mistake that we describe. By the time we edited the draft, in March 2019, Google Translate managed to get this particular example correct. However, the fix was fragile: if you left off the period at the end or put the sentence in parentheses, or changed the sentence to "The engineer whom we called to fix the telephone works on Sundays," Google Translate reverted to its old mistake of using *fonctionne* rather than *travaille*. Because the system's behavior frequently varies, possibly from day to day, perhaps as a function of changes in the exact composition of the training data set, it is hard to guarantee that any particular sentence will or won't work from one day to the next. So long as the basic nature of the algorithm remains the same, though, the general issues we describe are likely to continue to arise.

alone performing inferences over it in the context of understanding a sentence.

And, finally, deep learning is about static translations, from an input to a label (a picture of a cat to the label *cat*), but reading is a *dynamic* process. When you use statistics to translate a story that begins *Je mange une pomme* to *I eat an apple,* you don't need to know what either sentence means, if you can recognize that in previous bitexts *je* has been matched with *I*, *mange* with *eat*, *une* with *an,* and *pomme* with *apple*.

Most of the time a machine-translation program can come up with something useful, just churning through one sentence at a time, without understanding the meaning of the passage as a whole.

When you read a story or an essay, you're doing something completely different. Your goal isn't to construct a collection of statistically plausible matches; it's to reconstruct a world that an author has tried to share with you. When you read the Almanzo story, you might first of all decide that the story contains three main characters (Almanzo, his father, and Mr. Thompson), and then you start filling in some of the details about those characters (Almanzo is a boy, his father is an adult, etc.), and you also start to try to determine some of the events that took place (Almanzo found a wallet, Almanzo asked Mr. Thompson if the wallet belonged to him, and so forth). You do something similar (largely unconsciously) every time you walk into a room, or watch a movie, or read a story. You decide what entities are there, what their relationship is to one another, and so on.

In the language of cognitive psychology, what you do when you read any text is to build up a *cognitive model* of the meaning of what the text is saying. This can be as simple as compiling what Daniel Kahneman and the late Anne Treisman called an object file—a record of an individual object and its properties—or as complex as a complete understanding of a complicated scenario.

As you read the passage from *Farmer Boy,* you gradually build up a *mental representation*—internal to your brain—of all the peo-

ple and the objects and the incidents of the story and the relations among them: Almanzo, the wallet, and Mr. Thompson and also the events of Almanzo speaking to Mr. Thompson, and Mr. Thompson shouting and slapping his pocket, and Mr. Thompson snatching the wallet from Almanzo and so on. It's only after you've read the text and constructed the cognitive model that you do whatever you do with the narrative—answer questions about it, translate it into Russian, summarize it, parody it, illustrate it, or just remember it for later.

Google Translate, poster child of narrow AI that it is, sidesteps the whole process of building and using a cognitive model; it never has to reason or keep track of anything; it does what it does reasonably well, but it covers only the tiniest slice of what reading is really about. It never builds a cognitive model of the story, because it can't. You can't ask a deep learning system "what would have happened if Mr. Thompson had felt for his wallet and found a bulge where he expected to find his wallet," because it's not even part of the paradigm.

Statistics are no substitute for real-world understanding. The problem is not just that there is a random error here or there, it is that there is a fundamental mismatch between the kind of statistical analysis that suffices for translation and the cognitive model construction that would be required if systems were to actually comprehend what they are trying to read.

One surprisingly hard challenge for deep learning (though not for classical AI approaches) is just understanding the word *not*. Remember Siri's fail with "Find a fast food restaurant that is not McDonald's"? The person posing the query presumably wanted to get to an answer like "The Burger King at 321 Elm Street, the Wendy's at 57 Main Street, and the IHOP at 523 Spring Street." But there is nothing about Wendy's, Burger King, or IHOP that is particularly associated with the word *not,* and it doesn't happen all that

frequently that someone refers to any of them as *not McDonald's,* so brute statistics don't help the way they would with relating *king* and *queen*. One can imagine some statistical tricks to solve this particular issue (identifying restaurants), but a full treatment of all the ways in which *not* can be used is way outside the scope of current approaches.

What the field really need is a foundation of traditional computational operations, the kind of stuff that databases and classical AI are built out of: building a list (fast food restaurants in a certain neighborhood) and then excluding elements that belong on another list (the list of various McDonald's franchises).

But deep learning has been built around avoiding exactly those kinds of computations in the first place. Lists are basic and ubiquitous in computer programs and have been around for over five decades (the first major AI programming language, LISP, was literally built around lists) and yet they are not even part of the fabric of deep learning. Understanding a query with the word *not* in it thus becomes an exercise in driving square pegs into round holes.

And then there is the problem of ambiguity. Human languages are shot through and through with ambiguities. Words have multiple meanings: *work* (as a verb) can mean either *labors* or *functions correctly; bat* (as a noun) can mean either a flying mammal or a wooden club used in baseball. And those are comparatively clear-cut; listing all the different meanings of words like *in* or *take* fills many columns of a good dictionary. Indeed, most words except very technical ones have multiple meanings. And the grammatical structure of phrases is often ambiguous, too. Does the sentence *People can fish* mean that people are able to go fishing or that (as in Steinbeck's *Cannery Row*) people pack sardines and tuna fish into cans? Words like pronouns often introduce further ambiguities. If you say *Sam couldn't lift Harry because he was too heavy*, then, in principle, *he* could be either Sam or Harry.

What's amazing about us human readers is that 99 percent of the time, we don't even notice these ambiguities. Rather than getting confused, we quickly and with little conscious effort home in on the right way to interpret them, if there is one.*

Suppose you hear the sentence *Elsie tried to reach her aunt on the phone, but she didn't answer.* Although the sentence is logically ambiguous, there is no confusion about what it means. It does not ever consciously occur to you to wonder whether *tried* means *held court proceedings* (as in *The criminal court tried Abe Ginsburg for theft*), or whether *reach* means *physically arrive at a destination* (as in *The boat reached the shore*) or whether *on the phone* means the aunt was balanced precariously on top of the telephone (as in *a clump of dust on the phone*), or whether the word *she* in the phrase *she didn't answer* refers to Elsie herself (as it would if the sentence ended with *but she didn't get an answer*). Instead, you immediately zero in on the correct interpretation.

Now try getting a machine to do all that. In some cases, simple statistics can help. The word *tried* means *attempted* much more frequently than it means *held a court proceeding*. The phrase *on the phone* means *using the phone for communication* more frequently than it means *physically on top of the phone,* though there are exceptions. When the verb *reach* is followed by a person and the word *phone* is nearby in the sentence, it probably means *successfully established communication*.

But in many cases statistics won't get you to the right solution. Instead, there is often no way to resolve a given ambiguity without actually understanding what is going on. In the sentence that reads *Elsie tried to reach her aunt on the phone, but she didn't answer,*

* Not every ambiguity can be resolved without further information. If someone walks into the room and says *Guess what, I just saw a bat in the garage,* you really can't know whether they are taking about a flying animal or a piece of sports equipment. Until you get more context, there is nothing more that can be done, and it would not be fair to ask AI to read minds, either.

what matters is background knowledge* together with reasoning. Background knowledge makes it obvious to the reader that Elsie wouldn't answer her own phone call. Logic tells you that it must therefore be her aunt. Nobody has to teach us how to do this sort of inference in school, because we know how to do it instinctively; it follows naturally from how we interpret the world in the first place. Deep learning can't even begin to tackle this sort of problem.

Sadly, though, nothing else has really worked so far either. Classical AI techniques, of the sort that were common long before deep learning become popular, are much better at compositionality, and are a useful tool for building cognitive models, but thus far they haven't been nearly as good as deep learning at learning from data, and language is too complex to encode everything you would need strictly by hand. Classical AI systems often use templates. For example, the template [*PLACE1 is DISTANCE from PLACE2*] could be matched

* Putting this together actually requires two kinds of background knowledge. First, you need to know how telephone calls work: one person initiates the call, the other person may or may not answer; the communication is successful (the caller reaches the callee) only if the second person does answer. Second, you have to use a rule, often associated with Oxford philosopher H. P. Grice, that when people say or write things, they try to give you new information, not old information. In this case, since the sentence already said that Elsie made the call, there is no point in saying that she didn't answer it; the caller is never the person who answers a call. What is useful information is that the aunt didn't answer.

This example, by the way, is drawn from one of the most challenging tests for machines that is currently available, known as Winograd Schemas: pairs of sentences (like *Elsie tried to reach her aunt on the phone, but she didn't answer* vs. *Elsie tried to reach her aunt on the phone but she didn't get an answer*) that, at least for humans, can only be understood by making use of background knowledge. Ernie has played a central role in putting these together, along with Hector Levesque and Leora Morgenstern, and assembled a collection of Winograd Schemas online: http://www.cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html.

against the sentence *The moon is 240,000 miles from the earth,* and used to identify this as a sentence that specifies the distance between two places. However, each template must be hand-coded, and the minute you encounter a new sentence that differs from what comes before (say, *The moon lies roughly 240,000 miles away from the earth,* or *The moon orbits the earth at a distance of 240,000 miles*), the system starts to break down. And templates by themselves do almost nothing toward helping resolve the jigsaw puzzles of integrating knowledge of language with knowledge of the world in order to resolve ambiguity.

So far, the field of natural language understanding has fallen between two stools: one, deep learning, is fabulous at learning but poor at compositionality and the construction of cognitive models; the other, classical AI, incorporates compositionality and the construction of cognitive models, but is mediocre at best at learning.

And both are missing the main thing we have been building toward throughout this chapter: common sense.

You can't build reliable cognitive models of complex texts unless you know a lot about how the world works, about people, and places, and objects and how they interact. Without this, the vast majority of what you would read would make no sense at all. The real reason computers can't read is that they lack even a basic understanding of how the world works.

Unfortunately acquiring common sense is much harder than one might think. And, as we will see, the need for getting machines to acquire common sense is also far more pervasive than one might have imagined. If it's a pressing issue for language, it's arguably even more pressing for robotics.