# 2

# ALFRED (THE DISMISSIVE SCEPTIC)

*Philosophers, Go Away!*

In the previous chapter, we outlined a range of interesting philosophical challenges that arise in connection with understanding, explaining, and using AI systems. We tried to make the case that philosophical insight into the nature of content (and the difference between a system having content and simply being evidence of some sort) is centrally important both for understanding AI and for deciding how we should integrate it into our lives.

When we first started work on this project, we got in touch with people in the AI community. We thought that our work on these issues should be informed by people working in the field—those who are actually developing ML systems. When we approached people, they were friendly enough. We had many helpful conversations that have improved this book. However, it soon became clear to us that the people working at the cutting edge of AI (and in particular those working for the leading corporations) considered us, at best, lunchtime entertainment—a bit like reading an interesting novel. There was a sense that no one really took these

*Making AI Intelligible: Philosophical Foundations*. Herman Cappelen and Joshua Dever, Oxford University Press (2021).
© Herman Cappelen and Joshua Dever. DOI: 10.1093/oso/9780192894724.003.0002

philosophical issues *seriously*. Here is a caricatured summary of the attitude we encountered:

> Look, these big-picture concerns just aren't where the action is. I don't really care whether this program I'm working on is 'really a malignant mole detector' in any deep or interesting sense. What I care about is that I'm able to build a program that plays a certain practical role. Right now I can build something that's pretty decent at the role. Of course there's a long way to go. That's why I spend my time thinking about how adding back propagation, or long short term memory, or exploding gradient dampening layers, or improved stochastic gradient descent algorithms, will lower certain kinds of error rates. If you have something actually helpful to say about a piece of mathematics that will let me lower error rates, or some mathematical observations about specific kinds of fragility or instability in the algorithms we're currently using, I'm happy to listen. But if not, I'm making things that are gradually working better and better, so go away.

We take this dismissive reaction very seriously and much of this book is an attempt to reply to it. We are not going to dismiss the dismissal. At the end of the book, we have not refuted it. There's something to it, but, we argue, it's an incomplete picture and we outline various ways in which it is unsatisfying.

It's worth noting that this pragmatic-sceptic's dismissal of philosophy has analogues in almost all practical and theoretical domains. Practising mathematicians don't worry much about the foundations of their disciplines (they don't care much about what numbers are, for example). Politicians don't care much about theories of justice (they don't spend much of their time reading Rawls, Nozick, or Cohen). Those making medical decisions with massive moral implications don't spend much time talking to moral philosophers. And so it goes. There's a very general

question about what kind of impact philosophical reflection can have. One way to read this book is as a case study of how philosophers should reply to that kind of anti-philosophical scepticism.

We should, however, note that not all those working in AI share Alfred's dismissive attitude towards increased reflection on the foundations of ML systems. In 2017, Google's Ali Rahimi gave a talk where he compared the current state of ML systems to a form of *alchemy*: programmers create systems that work, but they have no real, deep, understanding of *why* they work. They lack a foundational framework. Rahimi said:

> There's a self-congratulatory feeling in the air. We say things like 'machine learning is the new electricity.' I'd like to offer an alternative metaphor: machine learning has become alchemy.[1]

It's become alchemy because the ML systems work, but no one really understands *why* they work the way they do. Rahimi is not entirely dismissive of making things that work without an understanding of why it works: 'Alchemists invented metallurgy, ways to make medication, dying techniques for textiles, and our modern glass-making processes.' Sometimes, however, alchemy went wrong:

> …alchemists also believed they could transmute base metals into gold and that leeches were a fine way to cure diseases. To reach the sea change in our understanding of the universe that the physics and chemistry of the 1700s ushered in, most of the theories alchemists developed had to be abandoned.

[1] https://www.youtube.com/watch?v=x7psGHgatGM.

More generally, Rahimi worries that when we have ML systems that contribute to decision making that's crucial both to individuals and to societies as a whole, a foundational understanding would be preferable.[2]

> If you're building photo sharing services, alchemy is fine. But we're now building systems that govern health care and our participation in civil debate. I would like to live in a world whose systems are built on rigorous, reliable, verifiable knowledge, and not on alchemy.

Many in the AI community dismissed Rahimi's pleading for a deeper understanding. Facebook's Yann LeCun replied to Rahimi saying that the comparison to alchemy was not just insulting, but wrong:

> Ali complained about the lack of (theoretical) understanding of many methods that are currently used in ML, particularly in deep learning. Understanding (theoretical or otherwise) is a good thing…But another important goal is inventing new methods, new techniques, and yes, new tricks. In the history of science and technology, the engineering artifacts have almost always preceded the theoretical understanding: the lens and the telescope preceded optics theory, the steam engine preceded thermodynamics, the airplane preceded flight aerodynamics, radio and data communication preceded information theory, the computer preceded computer science.[3]

We will argue that Rahimi is right: the current state of ML systems really is a form of alchemy—and not just for the reasons Rahimi mentions. The one important reason is that the field lacks an

[2] Note: Rahimi's worry is not specifically about interpretability, but the same point applies.

[3] https://www2.isye.gatech.edu/~tzhao80/Yann_Response.pdf.

understanding of how to describe the content of what it has created (or how to describe what it has created as deprived of content). We are presented with AI as if it is something that can talk to us, tell us things, make suggestions, etc. However, the people making AI have no theory that justifies that contentful presentation of their product. They have given us no rational argument for that contentful presentation. They've just written some algorithms and they have no deeper understanding of what those pieces of mathematics really amount to or how they are properly translated into human language or affect human thoughts. If the view is that these programs have no content at all, then that too is a substantive claim that needs justification: What is content such that these systems don't have it?

So: welcome to the world of philosophy. It's a world where there's very little certainty. There are many alternative models, the models disagree, and there's no clear procedure for choosing between them. This is the kind of uncertainty that producers and consumers of AI will have to learn to live with. It's only after a refreshing bath in philosophical uncertainty that they will start to come to grips with what they have made.

## A Dialogue with Alfred (the Dismissive Sceptic)

**Alfred**: I appreciate the interest you philosophers have in these issues. It's important that a broad range of disciplines reflect on the nature of AI. However, my job is to make exactly the kinds of AI systems that you talk about in the introduction of this book and I don't get it. I just don't see that there's anything you philosophers can tell me about interpretation that will help me do my

job. We've made all these amazing advances, and we did it without you. I'm not doubting that there's some interesting meta-reflections around these issues, but that's just lunch entertainment for us. It makes no real difference to what we do, day to day. Issues about the nature of interpretation and the nature of content don't seem pressing to me in my professional life.

So, as a conversation starter, let me try this: philosophical theories of meaning and language make no difference to what we do. For professional purposes, we can ignore them.

**Philosopher**: I don't see how you can avoid those issues. What do you think is going on with SmartCredit, then? We give the software access to Lucie's social media accounts, and it spits out the number 550. But so far, that's just pixels on a screen. The output of the program is useless until we know that 550 *means* a high risk of default. We need to know how to look at a program and figure out what its outputs mean. That's absolutely central to our ability to make any use of these programs. We can't just ignore that issue, can we?

**Alfred**: Of course we say things like, 'That output of 550 means that Lucie is a high risk of default.' But that's just loose talk—we don't need to take it seriously. All that's really going on is this. SmartCredit is a very sophisticated tool. It takes in thousands of data points and sorts and weighs them using complicated and highly trained mathematical algorithms. In the end SmartCredit spits out some number or other. That number doesn't in itself *mean* anything. It's just a number—just the end product of millions of calculations. Of course the bank should then take that number into account when deciding whether to extend a loan to Lucie. But not because the number *means* that Lucie is a default

risk—rather, because the number is the output of a highly reliable piece of software.

**Philosopher**: Wait, I'm not sure I understand what you're proposing. Just recently I went to the doctor and he used a machine learning program called SkinVision to evaluate a mole on my back.[4] According to him, SkinVision said that the mole was likely to be malignant, so he scheduled surgery and removed it. Are you telling me that the doctor was wrong and that SkinVision didn't say anything about my mole? I guess then I had surgery for no reason. Or what about the case of Eric Loomis? Loomis was found guilty of participating in a drive-by shooting, and was sentenced to six years in prison in part because, according to the judge, the machine learning program COMPAS said that Loomis was a high risk to reoffend.[5] Are you telling me that the judge was wrong and that COMPAS didn't say anything about Loomis's recidivist risk? If that's right, surely it was a huge injustice to give Loomis more prison time. It looks like we're treating these programs *as if* they are saying things all over the place, and making many important and high-stakes decisions based on what we think they are saying. If that's all wrong, and the programs aren't really saying anything, don't we need to do some serious rethinking of all of this technology?

**Alfred**: I think you're making a mountain out of a molehill here. Again, it's just loose talk to say that *COMPAS says that Loomis is high risk* or to say that *SkinVision says that your mole is probably malignant*. But that doesn't mean we're taking important actions for no reason. *SkinVision* didn't say that your mole was probably

[4] See https://www.skinvision.com. **Alfred**: Wait, we can talk in footnotes?

[5] https://www.wired.com/2017/04/courts-using-ai-sentence-criminals-must-stop-now.

malignant, but your doctor did say that. He said it in a sloppy way—he used the words 'SkinVision says that your mole is probably malignant'—but we don't need to take his exact phrasing seriously. It's clearly just his way of telling you (himself) about your mole. And there's no worry about having a mole removed because your doctor says that it's probably malignant, is there? The same with COMPAS. COMPAS didn't say that Loomis was high risk—the judge did. Again, he said it in a sloppy way, but we all know what's going on. And there's nothing wrong with giving someone a severe sentence because a judge says that he's a high recidivism risk, is there? That kind of thing happens all the time.

**Philosopher**: That's helpful. So the idea is that all the meaning and content is in the people saying things in response to the programs, not in the programs themselves. That's why we don't need a theory of content for the programs. (Hopefully we can get a good theory of content for people—but in any case that's not a special problem for thinking about AI systems.) But I'm still worried about how this idea is going to be worked out. My doctor gives SkinVision a digital photograph of my mole, and it produces a printout that says 'Malignancy chance = 73%'. Then my doctor says that my mole is probably malignant. On your view, SkinVision didn't say anything, and its printout didn't have any content—all the saying and all the content is coming from the doctor. But it sure seems like quite a coincidence that there's such a nice match between what my doctor *really meant* and what the words printed by SkinVision *seemed to me* but (on your view) didn't really mean.

**Alfred**: Of course it's not a coincidence at all. The designers of SkinVision included a helpful user interface so that doctors would know what to say when they got the results of a SkinVision analysis. There's nothing essential about that—SkinVision could have

been designed so that it just outputs a graph of a function. But then doctors would have needed more training in how to use the program. It makes sense just to have the programmers add on some informative labelling of the outputs on the front end and save the doctors all that work.

**Philosopher**: 'Informative labelling'—I like that. You can't have informative labelling without information. Doesn't that then require that the outputs of SkinVision do mean something, do carry the information that (for example) the mole is probably malignant?

**Alfred**: Good point. OK, what I should have said was not that it's the *doctor* who's the one who's really saying something—rather, it's the *programmer* who's really saying something. When SkinVision prints out 'Malignancy chance = 73%', that's the programmer speaking. She's the one who is the source of the meaning of those words. They mean what they do because of her programming actions, not because of anything about the SkinVision program itself. SkinVision is then just a kind of indirect way for the programmer to say things. That's a bit weird, I admit, but there are lots of other forms of indirect announcement like that. When the programmer writes some code which, when run, prints 'Hello World', it's the programmer, not the program, who greets the world. SkinVision and other AI systems are just more complicated versions of the same thing. The doctor then *also* says that your mole is probably malignant, but that's just the doctor passing on what the programmer indirectly said to him.

**Philosopher**: That's an interesting idea. But I'm worried that it has a strange consequence. Suppose that the programmer of SkinVision had been in a perverse mood when programming the final user interface, and had set things up so that the mathematical

output that in fact leads to SkinVision printing 'Malignancy chance = 73%' instead caused SkinVision to print 'Subject is guilty of second degree murder'. Would that then mean that SkinVision, rather than a piece of medical software, was instead a bit of legal software, making announcements about guilt or innocence rather than malignant or benign statuses?

**Alfred**: What? Of course not. Why would you even think that? SkinVision's whole training history shaped that neural network into a medical detector, not a legal detector. How would a perverse programmer implementing perverse output messages change that?

**Philosopher**: Well, doesn't it follow from what you said? If SkinVision itself isn't really saying anything, and it's just a tool for letting the programmer speak, then if the programmer chooses to have it produce the words 'Suspect is guilty of second degree murder', what's said (by the programmer, through the program) is that the suspect is guilty of second degree murder. And if the information conveyed is legal, rather than medical, then it looks like a piece of legal software.

**Alfred**: Not a very good piece of legal software! The guilt and innocence announcements it produces aren't going to have anything to do with whether the person is really guilty. You can't tell guilt or innocence from a photograph of a mole. And even if you could, SkinVision hasn't been trained to do so.

**Philosopher**: Agreed, it would be a terrible piece of legal software. But my point is just that that's what it would be, since its outputs mean what the programmer wants them to mean. I can see that in this case there's some plausibility to the claim that when the perversely programmed SkinVision prints 'Subject is guilty of second-degree murder', what's said is that the subject

is guilty of second-degree murder. (Whether it's SkinVision itself or the programmer who's saying this is less clear to me.) But I'm worried that that's a special feature of this example. In this particular case, the programmer has decided to put the program output in the form of words in a pre-existing language. It's thus very tempting to take that output to mean whatever those words mean in the language. In the same way, if a monkey banging on a keyboard happens to type out 'To be or not to be, that is the question', we might feel some inclination to say that the monkey has said something. But probably that feeling should be resisted, and we should just say that the *sentence* means something, and that the monkey has accidentally and meaninglessly produced it.

Consider another case. StopSignDetector is another machine learning neural net intended to be used in self-driving autonomous vehicles. The plan for StopSignDetector was, not surprisingly, to have it be a stop sign detector, processing digital images from a car camera to see if there is a stop sign ahead. But StopSignDetector doesn't print out 'There is a stop sign', or anything like that. There's just a little red light attached to the computer that blinks when the program reaches the right output state. As I understand your view, the blinking red light doesn't mean anything in itself, but is just a device for the programmer saying that there is a stop sign. That's because, I guess, the programmer intends the blinking red light to announce the presence of a stop sign. But now add in the perverse programmer. What if the programmer decides instead that the blinking red light should announce the presence of a giraffe—but doesn't change anything in the code of StopSignDetector. Does that mean that we end up with a very bad giraffe detector?

**Alfred**: I think all of this is getting much more complicated than it really needs to be. We speak sloppily as if these programs are saying things, producing outputs that somehow represent specific facts about the world. That's all just sloppy speech. In many cases, that sloppiness can be fixed up by taking us *really* to be talking about what the end user (like the doctor or the judge) is saying or what the original programmer is saying. But sure, I agree that in weird cases when end users or programmers have weird secret plans, that's not a good way to fix up our sloppy talk. But it's not that hard to find a different way, is it?

Think about your standard pocket calculator. You push the buttons '58 + 67' on the keyboard, and on the display it shows '125'. Does that mean that the calculator is *saying* that 58 plus 67 is 125? Surely not—there's no need for that kind of content talk. Of course, someone using the calculator might then say '58 + 67 = 125', and thereby mean (as people do) that 58 plus 67 is 125. And it's presumably not an *accident* that the calculator display looks the way it does—the original programmer of the calculator software chose that display format because of their plan that the calculator be a tool to announce arithmetic facts. But even if we discovered that the programmer had strange secret plans and the calculator user had strange secret interpretive ideas, it wouldn't matter. That's because in the end the calculator is just a tool for getting at mathematical results. So long as the calculator is working correctly, who really cares what anyone's communicative plans are, or what the calculator or anyone else is 'really saying'.

**Philosopher**: But I'm not sure a calculator is the right comparison for you. The programming of a calculator is a straightforward example of symbolic representational programming. If we look into the coding details of the calculator, we will indeed be able to

find the parts of the program that represent numerical values, and that represent the applications of various mathematical operations to those numerical values. Here, it looks entirely natural to me to say that the calculator display really does mean that 58 plus 67 is 125. None of the special features of (for example) SmartCredit that made its contents so obscure seems to be present in this case.

**Alfred**: OK, fair enough. But I bet I could program up a machine learning pocket calculator if I really set my mind to it. I bet you haven't actually checked out the coding of your TI-Nspire—would you really change anything in how you used the calculator if you discovered that it had a neural network implementation?

**Philosopher**: Probably not. But that's because I would think that, whether neural network or not, the calculator's program was *about* mathematical operations. Remember, I'm not a sceptic about the role of content in these cases, you are. I'm happy to say that we don't need to worry about obscure communicative plans on the part of the programmer or the user, because I'm happy to say that the program itself means something. (Of course, I think it's a very hard question *why* it means something, and I think in some cases we might have a lot of trouble figuring out *what* it means.) So what's your view on this? Don't you need a view on what it means to say that the calculator is a 'tool for getting at mathematical results'? That looks an awful lot like a disguised claim about the contents of the calculator claims.

**Alfred**: That's got to be too fast. A hammer is a tool for pounding in nails, right? That's not a claim about the meaning or content of a hammer. That's just an observation about what hammers are useful for.

**Philosopher**: Agreed. But I think this overlooks an important distinction. A hammer isn't an informational tool. When we use a

hammer, we're not trying to learn anything—we're just trying to get something done (get some nails in some wood). It's not too surprising if we don't need any notion of content to explain that kind of tool. But a language is also a tool, isn't it? And to say *that* kind of tool, we need to talk about contents. That's because language is an informational communicative tool, a tool that we're using to learn things. So we need to say what sentences mean to see what we can learn from them. And SkinVision and COMPAS look like tools of the same sort. We're not trying to *do* something with those tools—all of the *doing* is by the doctor or the court. We're just trying to get some information out of the tools. And if we're going to get information out, we need a contentful interaction with the program.

**Alfred**: Good, that helps me see what I want to say. In the end, the tools I want to make are more like hammers than like languages. Consider an example. I want to build a self-driving car. I'm not trying to make a car that I'll learn something from—I just want a car that will do something for me. I want a car that I can get into and that will then take me to the right place. That's a big project, so I'm not trying to do it all at once. Along the way, I produce a machine learning image recognition program that will beep when there's a pedestrian in the road. For now, that can be a helpful signal to the driver. But eventually, I'll have that bit of programming integrated into a larger autonomous vehicle program. Once that's all done, all I care about is that the car won't in fact hit pedestrians. Whether the beeps from that one part of the program 'mean that there's a pedestrian in the road' makes no difference to me. Why would I care? I'm not trying to give anyone any information with that beeping; I'm just trying to make sure that the car doesn't crash.

**Philosopher**: I see the idea, but how does that help with other cases? Maybe we don't need to assign contents to the full self-driving car, but before the pedestrian detector is integrated into the full car, while it's being used to warn human drivers, don't we need its beeps to *mean* that there's a pedestrian in the road?

**Alfred**: I don't see why. I'm happy to think of the driver in the same way that I think of the self-driving car. I'm not interested in getting any particular *contents* to the driver. What I care about is that the driver swerves when the program beeps. So long as that happens, and the pedestrian isn't hit, I'm happy.

**Philosopher**: I see. So you're just thinking of the programs as little causal prods that push people into the right kind of activity. SkinVision just needs to cause doctors to perform surgeries; never mind what the doctors believe. COMPAS just needs to cause judges to issue severe sentences; never mind what judges might learn from COMPAS.

**Alfred**: Right. Sure, probably the best way to get doctors to perform surgeries under the right conditions is to get them to believe that people need surgeries under those conditions. But that's just an accidental feature of doctors making them different from nails. The thing that really matters is just that our program causally prompts the right things to happen.

**Philosopher**: I'm not sure this 'it's all just causal prods' idea is going to be as easy to work out as you seem to think. You said you just wanted 'a car that I can get into and that will then take me to the right place'. But where did this notion of 'right place' come from? That requires that the car takes you where you want to go, and that then requires that you are able to *tell* the car where to go. But doesn't that still require a contentful interaction with the program? Maybe it's on the input side rather than the output side,

but the issues seem to me to be the same—I need to be able to do something to the program that I can count on putting the program into the right state. I need to be confident that when I tell the self-driving car to take me to the airport, its subsequent driving will be guided by the content of what I told it.

**Alfred**: I'm tempted to say that the problem you're pointing out is just another artefact of our being only part-way through the overall project. I already agreed that *for now* I want the pedestrian detector's beeps to be understood by human drivers as signalling that there is a pedestrian in the road. We're talking about understanding and meaning here because the programming project isn't finished yet, so we can't just let the car do its own self-driving business. But the same is true for the need to give the car directions. Down the road, the goal should be a car that you don't need to give directions to. The car will figure out how to deal with pedestrians in the road; it will also figure out how to deal with a passenger in the car. Maybe it will access your calendar and determine where you ought to be and automatically take you there.

**Philosopher**: Wait, 'figure out'? 'Determine'? 'Access your calendar'? That all looks like content-based talk.

**Alfred**: Sure, but it's all dispensable in the same way. When I say that the car will figure out how to deal with pedestrians in the road, I just mean it won't hit pedestrians in the road. When I say the car will figure out how to deal with a passenger in the car, I just mean that it will take that passenger to a location where the passenger ought to be. And so on.

**Philosopher**: I'm not sure I like the vision of the AI future you're sketching here. These days when I get in the car and drive somewhere, I have plans and reasons for what I'm doing and I perform a bunch of deliberate intentional actions in pursuit of my

goals. Your self-driving car takes that all away from me. I don't need any plans, or any reasons for going anywhere. I just get in the car, and the car takes me somewhere that will work out well for me. It feels like a Wall-E future, with all of us passive passengers on the Axiom. It's important to us that we have reasoned engagement with the world—aren't you proposing to shrink that reasoned engagement down to nothing, by embedding us in a network of devices that just causally push us around to where we ought to be?

**Alfred**: Well, as long as you're really getting where you ought to be, is it really that bad? We're surrounded by lots of systems and devices that take care of our needs without our reasoned engagement. When you're exposed to germs, your immune system just takes care of it for you—it causally pushes bits of your body into the right places without any intervention by you. Things wouldn't be any better if you had to reason your way through a viral infection, would they?

**Philosopher**: Fair enough, although just because something is good in some places doesn't mean it's good everywhere. But surely there's also a real issue about whether we can count on the self-driving car taking us where we *ought* to be. What's our source of confidence in that 'ought'? Either we're just building into the program what the right final goals are (get us where our calendar says we ought to be), in which case it looks like we still need content tracking with the program. Or we've got the kind of advanced AI that has the ability to reshape the categories it's been trained to track, in which case, if there's no notion of content of the program's reshaped categories, I'm not sure why we should be confident that what it's doing is in any sense getting us where we *ought* to be.

**Alfred**: Look, all of this is getting extremely speculative. Forget about this utopian/dystopian picture in which our AI systems just shepherd us through the world. Remember, I've already observed that you can think of human users now as being like the eventual self-driving car. I don't care whether the car *knows* that there's a pedestrian in the road and *takes that into account*. All I care about is that when the pedestrian detector beeps, the car changes course. And similarly for the human user. I don't care whether the human user *knows* that there's a pedestrian in the road and *takes that into account*. All I care about is that when the pedestrian detector beeps, the driver changes course. Who cares what the underlying mechanism is by which that happens?

**Philosopher**: There's a sense in which I agree with all of that. Forget about programs entirely, and just think about people. There's some sense in which all of the content talk we go in for may be optional. Maybe we can stop thinking about other people as creatures having beliefs and desires and plans with contents and making claims with contents, and just think about them as lumbering obstacles to be manipulated and manoeuvred around. But surely *something* is gained by instead thinking about people as bearers of content. If we've at least reached the point, then, of saying that content talk for AI systems is exactly as dispensable as content talk for people, I think we've got enough to motivate some careful thinking about how to make that content talk work out in the AI case.

**Alfred**: Fair enough. Let's at least see what you've got.