5

APPLICATION

The Predicate 'High Risk'

In this and the next two chapters, we show how the framework outlined in Chapter 4 can be applied to particular outputs of AI systems. We'll take as our example the output from SmartCredit that resulted in Lucie being turned down for a mortgage. Recall from Chapter 1 that according to SmartCredit, *Lucie is high risk*. We'll split this statement into three parts and present a separate theory for each part:

(i) The statement is about Lucie, i.e. SmartCredit refers to Lucie. We want to figure out how SmartCredit can refer to Lucie.

(ii) The statement is about the property of *being high risk*. We want to figure out how SmartCredit can denote that property.

(iii) The statement predicates the property of being high risk to Lucie. On the assumption that SmartCredit has the capacities outlined in (i) and (ii), we next want to figure out how SmartCredit can attribute the property of being high risk to Lucie.

Making AI Intelligible: Philosophical Foundations. Herman Cappelen and Joshua Dever, Oxford University Press (2021).
 DOI: 10.1093/oso/9780192894724.003.0005

This chapter is about (ii), i.e. how SmartCredit can pick out the property of *being high risk*. Chapter 6 is about how SmartCredit refers to Lucie. Chapter 7 outlines a proposal for how SmartCredit can predicate *being high risk* to Lucie. Each chapter will make use of different externalist tools: for names we use ideas from Evans; for predicates, we use ideas from Kripke; for predication we use ideas from teleosemantics. The proposals all instantiate the general strategy we presented in Chapter 4.[1]

Our test case is relatively simple. There's an enormous amount more to be done even if we're completely successful with this test case—attributing content to AI outputs that don't have linguistic form, and especially attributing content to non-explicit AI internal states that don't appear as output (understanding SmartCredit not just as denying a loan, but doing so for some reason). However, even the simple test case will be challenging enough for now and so that's where we'll start. Once we succeed with these baby steps, it'll be time to move on to the more complex issues.

The Background Theory: Kripke-Style Externalism

Saul Kripke's series of lectures published as *Naming and Necessity* (Kripke 1980) outlines what has now become one of the leading theories of how language connects to the world. Similar and complementary views were developed at more or less the same time

[1] It might seem surprising that we use a variety of externalist theories that are often seen as competitors. It's not strange if, as we do, you endorse a form of metasemantic pluralism: There are many metasemantic mechanisms (and there could be even more than we currently know of).

by Hilary Putnam (1975) and Tyler Burge (1979). The distinctive feature of these views is a form of externalism: what grounds meaning and determines what sentences and expressions are about is external to the speaker's mind. If you look at just the speaker of 'John is in Paris', you won't find out what the expression 'Paris' is about. Moreover, it is not determined 'computationally': reference determination is not about how a particular symbol computationally integrates into the neural net calculations. It's not computational in the sense of being about the internal computational structure of the representational item itself. It's all about external relations to the world—looking at internal computation structure is just wrong-headed through and through. What determines that an utterance of 'Paris' is about Paris has to do with the history of use of that name. Here is how Kripke introduces his basic idea:

> Someone, let's say, a baby, is born; his parents call him by a certain name. They talk about him to their friends. Other people meet him. Through various sorts of talk the name is spread from link to link as if by a chain. A speaker who is on the far end of this chain, who has heard about, say Richard Feynman, in the marketplace or elsewhere, may be referring to Richard Feynman even though he can't remember from whom he first heard of Feynman or from whom he ever heard of Feynman. He knows that Feynman is a famous physicist. A certain passage of communication reaching ultimately to the man himself does reach the speaker. He then is referring to Feynman even though he can't identify him uniquely.
>
> (Kripke 1980: 91)

Note that on this view, a speaker can use a name to talk about something or someone even if that speaker has no ability to describe that thing correctly. It is not the speaker's beliefs about what 'Paris' denotes that determines what she denotes by 'Paris'.

Aboutness is determined by an external chain of communication. Here is Kripke's rough summary of his view:

> An initial 'baptism' takes place. Here the object may be named by ostension [. . .]. When the name is 'passed from link to link', the receiver of the name must, I think, intend when he learns it to use it with the same reference as the man from whom he heard it. If I hear the name 'Napoleon' and decide it would be a nice name for my pet aardvark, I do not satisfy this condition. (Kripke 1980: 96)

The structure of such a theory is relatively simple. It has three parts:

1. There's an **introductory, anchoring, event**, where an expression is 'hooked up' to some part of the world ('Paris' to Paris, 'Napoleon' to Napoleon, 'zebra' to zebras, 'chair' to chairs, etc.). Kripke suggested this could happen through a baptism (as in the example in the quotation above) or by a description being used to pick out the thing talked about (if we said, 'let "Alfred" be the name of the first person to buy a copy of this book').
2. Then there's **a chain of transmission** from person to person. This is what Kripke also calls a communicative chain. Kripke stipulates that this chain has to be reference preserving (more on that below).
3. Then there's a speaker using the expression at some point in the chain: She can use, say, 'zebra' to talk about zebras **by virtue of being part of a communicative chain** that started with zebras (e.g. that started with a baptism where a speaker said: call those kinds of animals 'zebras'.)

There are two notable and relevant features of this view:

1. Even if you knew everything that had ever happened to the speaker, you would not know what she is referring to (what she is talking about). That is determined by historical facts that are independent of that particular speaker (the beginning of the communicative chain, which could have started before the speaker was born). In particular: if you look inside the head of the speaker, there's no fact 'in there' that will tell you what she is talking about.
2. The speaker could be radically wrong about what she is talking about—she could be wrong about what her use of 'Napoleon' refers to (she need not know what she is talking about because she need not know what is at the origin of the communicative chain).

Of course, often things are more complicated and messy. And the Kripkean view has inspired mountains of theorizing, both defending and furthering his externalist view (as in Salmon 1986, Soames 2002 and more recently, in a textbook presentation: our 2018) and responding to it (as in the causal descriptivism of Lewis 1984 and Kroon 1987, or the two-dimensional semantics defended in works like Chalmers 2006).

While important, we don't think that we need to engage with this literature too much here—our aim is to get as far as we can with (and simply assuming the truth of) the key Kripkean insight and picture of reference. With that mind, let's consider how this can help us understand content attribution to ML systems.

Starting Thought: SmartCredit Expresses High Risk Contents Because of its Causal History

The externalist story we just outlined has two ways of describing the anchoring of the content of ML systems:

1. SmartCredit's history includes an anchoring event, anchoring on high risk.
2. SmartCredit is a link in a transmission chain leading back to high risk.

The obvious problem with this line of thought is that there's no simple way to apply Kripke's picture directly to SmartCredit. There's just nothing that looks like a standard anchoring event in SmartCredit's history. SmartCredit never points to anything (let alone to the property of high risk), it never descriptively singles out anything, it never has referential intentions. And SmartCredit can't be inheriting a semantic connection of high risk from elsewhere (from the programmers, perhaps) in the usual way, because there's nothing that looks like the standard transmissive link in SmartCredit's case. SmartCredit has no intention to use a term in the same way as those from whom it received the term. In sum:

- There is nothing like a Kripke-style baptism event.
- There is no intention to refer to high risk, no pointing at high risk, no descriptive identification of high risk.
- And there is nothing like Kripke-style proper transmission. In particular, there are no reference-preserving intentions.

So while—for reasons we spelled out above—externalism might seem plausible at first glance, the versions that are available off the shelf seem initially unpromising.

Anthropocentric Abstraction of 'Anchoring'

The problems above take this form: they point out disanalogies between the ways in which humans initiate and participate in communicative chains and the ways AI, on our proposal, would do so. There will obviously be very many such differences. Humans are animals that engage with the world in ways computers can't. We have all kinds of inter- and intra-personal experiences that computers lack. To be open to the idea that systems very different from humans can have content, we need to engage in what we call 'anthropocentric abstraction': the effort to find some more abstract description of the structure that leads to content attribution for humans—a description that moves away from the contingencies and limitations of our peculiarities.

There's a general structure to that process of anthropocentric abstraction:

- We start with a pattern instantiated by human peculiarities.
- There is then a hierarchy of degrees of abstractions: what we are looking for is a degree of abstraction that preserves sufficiently many important features of the original phenomenon. It's not too abstract and it's not too focused on specific details. Call that **'the abstractive sweet-spot'**.

This process of abstraction can be done bottom-up (starting with particular cases and working one's way up) or top-down (starting with a general theory and working one's way down to specific cases). Kripke's strategy is the former: he starts with cases that we all agree are instances of reference, and then he builds a very thin theoretical framework on top of that. What he in effect does is give a brief sketch of how we humans typically do it, based on reflection on a few cases. He does not start by an a priori articulation of a general condition that has to be imposed on reference and then looking for human behaviour that satisfies those conditions. We will use the Kripkean bottom-up approach. We start with the assumption that an ML system is, say, a high risk detector. We then explore its history and we ask: what in that history corresponds to what we find in the human case? Does any of it match—at an appropriately abstract level—some of the components of what Kripke finds in the human case?

Schematic AI-Suitable Kripke-Style Metasemantics

In order to de-anthropocentrize the Kripkean story that the analogue of anchoring for AI systems is to be found in their neural net training, very roughly, our proposal is this:

> SmartCredit's outputs express the property of high risk because the training of SmartCredit's neural network was done against the property of high risk, thereby anchoring the program to that property.

This is the coarse-grained answer, but the details will matter a great deal and they are in large part unsettled. Some of the relevant details involve how neural nets are trained:

- A generic initial neural net is given samples from a large pool of training cases.
- Each training case has been hand-coded ('high risk' versus 'low risk'), for example by the programmers.
- The AI's output for the training case is then compared to the hand coding using some scoring function evaluating how well the AI classified the training case.
- That score is then used to update the weightings of the node connections in the neural net.

Our suggestion is that SmartCredit's outputs express the property of high risk because SmartCredit was given training cases that were hand-coded for being high risk lendees or low risk lendees, and was then scored highly for categorizing cases hand-coded high risk as high risk and scored poorly for categorizing cases hand-coded high risk as low risk. Its net is then adjusted on the basis of that score. After some (indefinite) number of iterations of this process, SmartCredit's outputs gain representational content.

Here, in slogan form, is the proposal:

> **AI Anchoring**: SmartCredit is anchored in high risk via a scoring function that scores well for matching high risk hand coding and low for not matching.

We think that this is a plausible starting point, but it's just a schematic view right now, because there are a number of choice points that we'll encounter as we think through the details of hand coding, of scoring functions, and of updating procedures.

Complications and Choice Points

Hand coding choice points: We've been setting out a Kripke-inspired picture on which SmartCredit is anchored in the property of high risk because it's trained on a bunch of cases that have been hand-coded as high risk or low risk. But there's more than one thing we might mean by 'hand-coded as high risk or low risk'. To see this, consider two cases:

(C1) SmartCredit's training set was assembled by programmer Pat. Pat has gone back through old bank records, found numerous instances of people who did and did not default on their loans, and then put together files of the input data (at the time of loan application) for these people, together with a label of 'high risk' for the actual defaulters and 'low risk' for the actual non-defaulters. But Pat makes a few mistakes along the way. Among the thousands of test cases, there are three (for three individuals A, B, and C) that Pat marks with a 'high risk' label even though they didn't, in fact, default on their loans. When SmartCredit is trained on this data set, is SmartCredit being trained on a data set hand-coded for the property of being high risk, or on a data set hand-coded for the property of being high risk or one of A, B, and C?

(C2) Pat wanted more cases than were available in the bank's lending history, so created a number of additional fictional cases. Pat uses the best of her financial knowledge to design fictional cases of defaulters and non-defaulters, and then creates initial data sets suitable for those fictional cases, and labels those cases with 'high risk' and 'low risk' labels as appropriate.

> But, of course, there is no independent fact of the matter of whether these cases are genuinely high risk or low risk cases. When SmartCredit is trained on this data set, is SmartCredit being trained on a data set hand-coded for the property of being high risk, or on a data set hand-coded for the property P of being someone who Pat thinks would be high risk?

We can now consider various particular versions of the general Kripke-inspired metasemantics:

> (K1) SmartCredit is anchored to a property P if P is the property in fact shared by all the training cases hand-coded with the same label.
> (K2) SmartCredit is anchored to a property P if P is the property that the hand-coder intends to be indicating by marking training cases with a given label.

Consider hand coding. Suppose that some of the training cases are mislabelled in the hand coding—cases that are in fact high risk lenders are marked as low risk lenders. How will this affect what property the neural net is anchored in? (The *intended* property? Some disjunctive property?) Or suppose we don't use actual cases as training cases, but fictional cases, so that there is no independent fact of the matter about how the cases are *correctly* hand-coded. What effect will such training cases have on content fixation?

Scoring choice points: When SmartCredit is being trained—and thus, on our Kripke-inspired picture, hopefully being anchored to the property of high risk—its outputs for a test set are compared to hand-coded evaluations of the test set. We then want to give SmartCredit feedback based on how well it did at

categorizing the test cases. But there are many notions of 'how well' that could be used here.

Consider a complication that we've been sweeping under the rug. SmartCredit, like many AI classifiers, doesn't produce *binary* classification judgements. When given Lucie's data, it doesn't just report that Lucie is high risk or report that Lucie is low risk. Instead, it assigns probabilities that Lucie is in each category. So SmartCredit might report that Lucie is 0.8 likely to be high risk and 0.2 likely to be low risk. Now suppose that SmartCredit produces probabilistic outputs like this for thousands of cases. For each of these cases, we also have hand-coded evaluations of whether the person is genuinely high or low risk. Now we want to assess how well SmartCredit did. That can't just be a count of how many cases SmartCredit got right and how many SmartCredit got wrong—the 'how well' assessment needs to take SmartCredit's probabilities into account.

What we need is a *scoring function*. But there are many ways to design a plausible scoring function, and different scoring functions are in fact used in different AI applications. Let's consider briefly two scoring functions. One is the **Brier score**. To obtain SmartCredit's Brier score, for each case we take the difference between SmartCredit's assigned probability of being high risk and the actual 'probability of being high risk' (1 if the case is hand-coded as high risk; 0 if it is hand-coded as low risk). We then square each of those differences and add them. So for a test set S, SmartCredit's Brier score is:

$$1/|S| * \left(\sum_{(i \text{ in } S)} \left(L_i - A_i\right)^2\right)$$

where L_i is SmartCredit's probability that the ith case is high risk and A_i is the actual probability that the ith case is high risk.

Lower Brier scores indicate better accuracy; higher Brier scores indicate worse accuracy.

Another scoring function is the **log-loss score**. To get SmartCredit's log-loss score, take SmartCredit's assigned probability that a given case is high risk, and then take either the logarithm of that probability (if the case is hand-coded as high risk) or the logarithm of 1 minus that probability (if the case is hand-coded as low risk). So for a test set S, SmartCredit's log-loss score is:

$$-1/|S| * \left(\sum_{(i \text{ in } S)} \left(A_i * \log\left(L_i\right) + \left(1 - A_i\right) * \log\left(1 - L_i\right)\right)\right)$$

Again, lower log-loss scores indicate better accuracy.

Brier scores and log-loss scores won't in general be the same, and so training a program to minimize the Brier score won't in general produce the same behaviour as training a program to minimize the log-loss score. For example, the log-loss score punishes large probability errors more severely than does the Brier score. Consider the accuracy penalties, for both scoring functions, of outputting probabilities of either 0.01 or 0.001 for a case that is in fact high risk:

Brier score:
Output = 0.01: Brier score = $(1\text{-}0.01)^2 + (0\text{-}0.99)^2 = 1.9602$
Output = 0.001: Brier score = $(1\text{-}.001)^2 + (0\text{-}0.999)^2 = 1.996002$
Log-loss score:
Output = 0.01: Log-loss score = -log(0.01) = 2
Output = 0.001: Log-loss score = -log(0.001) = 3

The Brier score increases the accuracy penalty for the second case by less than 2 per cent, while the log-loss score increases the accuracy penalty for that case by 50 per cent. (Brier scores for individual cases are bounded at 2 while log-loss scores are unbounded, so in extreme cases the penalty increase for the Brier score tends to 0 while the penalty increase for the log-loss score increases arbitrarily.) So an AI system trained using a log-loss scoring function is, compared to a system trained using a Brier score, made more likely to avoid extreme probability errors.

Of course, if SmartCredit is getting everything right, it doesn't matter which scoring function is used. But no AI system is going to get every judgement right. Just for a toy case, let's suppose that the financial prospects of bitcoin speculators are particularly difficult for SmartCredit to evaluate. (For whatever reason, the kinds of correlations between social media footprint and creditworthiness that SmartCredit relies on are much less robust among bitcoin speculators than among the general population.) So SmartCredit's assigned probabilities for bitcoin speculator test cases tend to produce extreme errors—SmartCredit is often highly confident that a genuinely risky bitcoin speculator is low risk, or vice versa.

Now consider two properties to which SmartCredit might be anchored, Kripke-style: (i) being high risk, or (ii) being a bitcoin speculator or a high risk non-bitcoin-speculator. There could then be two different elaborations of the Kripke-inspired picture, which predict anchoring onto these different properties:

> (K1) SmartCredit is anchored to property P if SmartCredit is trained using a training set hand-coded for some property Q such that P is the simplest property that produces a reliably low Brier score compared to the hand-coded Q facts.

(K2) SmartCredit is anchored to property P if SmartCredit is trained using a training set hand-coded for some property Q such that P is the simplest property that produces a reliably low log-loss score compared to the hand-coded Q facts.

Then suppose that according to the metasemantic view (K1), SmartCredit represents being high risk, while according to the metasemantic view (K2), SmartCredit represents being a bitcoin speculator or a high risk non-bitcoin-speculator. Both (K1) and (K2) are particular ways of filling out the general externalist Kripke-inspired metasemantics—how could we decide which of (K1) and (K2) is the right way to abstract a non-anthropocentric metasemantics from the Kripkean starting point?

At this point, we turn to the meta-metasemantics. Given our interpreter's knowledge-maximization picture of the meta-metasemantics, we need to know whether a (K1)-style metasemantics or a (K2)-style metasemantics for SmartCredit will maximize the knowledge we obtain through our interactions with SmartCredit. The answer to that question is then sensitive to a number of externalist features of the social and environmental setting in which we use SmartCredit. For example:

- If the environment is heavily populated with bitcoin speculators, (K1) will have SmartCredit inaccurately labelling them as (e.g.) high risk, since the Brier score doesn't weight the extreme probability errors for these cases heavily enough to influence the property tracked, while (K2) will have SmartCredit accurately labelling them as either bitcoin speculators or as high risk non-bitcoin-speculators. (K2) would then, to that extent, be more

conducive to interpreter knowledge. But if the environment is sparsely populated with bitcoin speculators, the stronger property represented by SmartCredit according to (K1) might lead to more knowledge on our part (since we can also infer the disjunctive property).

- What we plan to do with the classification we get from SmartCredit can influence which property is knowledge-maximizing for us. Suppose the bank has a policy of not lending to bitcoin speculators, and the cases of both Simon the bitcoin speculator and Lucie the non-bitcoin-speculator are both given to SmartCredit. We then form both (i) classificatory beliefs about Simon and Lucie, and (ii) a secondary practical belief about how we ought to treat Lucie (that we should or should not give her a loan). When the content of the secondary practical beliefs rely on the SmartCredit content ascribed by metasemantics (K1), our secondary belief about Lucie is knowledge (because *bad risk* is a good reason to deny a loan). But when the content of the secondary practical beliefs rely on the SmartCredit content ascribed by metasemantic (K2), our secondary belief about Lucie is not knowledge (because *bad risk or bitcoin speculator* is not a good reason to deny a loan). So (K1) has some knowledge maximization effect over (K2). But if the bank has no policy against loaning to bitcoin speculators, the secondary practical question arises for Simon as well. (K1) and (K2) make that secondary belief about Simon not knowledge, but (K2), and not (K1), makes the classificatory belief about Simon knowledge. So in this environment, (K2) has some knowledge maximization effect over (K1). In general, since log-loss scoring functions avoid *extreme* errors

> more than Brier scoring functions, (K2) will be a better knowledge-maximizer than (K1) in cases in which our subsequent use of the categorizing is in a context in which we care a lot about avoiding bad errors. So, for example, a 'guilt-innocence' detector might be more likely to be knowledge-maximizing when it's guilt detecting according to the (K2) log-loss metasemantics than when it's guilt detecting according to the (K1) Brier score metasemantics, given the nature of the other beliefs we're going to form based on the guilt-innocence categorization.

Update choice points: Even more pressingly, there are many ways of going from a scoring of the AI output to a specific alteration of the neural net node connection weightings (many complicated papers are written about this in the AI literature). Clearly not all ways of updating are going to produce the same content (intuitively, 'inverting' the update function for SmartCredit's training should 'invert' its representational contents), so again there's room for interaction between the details of the update function and the details of the anchoring.

Taking Stock

Here is what we have done:

> We started with an outline of Kripke's causal chain metasemantics.
> We observed that the details of this metasemantics aren't straightforwardly applicable to AI systems.

We suggest that the Kripkean metasemantics is an anthropocentric instance of a larger class of metasemantic principles.
We took some initial steps toward de-anthropocentrizing, proposing an AI-friendly version of anchoring.
Finally, we outlined some choice points for that theory.

Appendix to Chapter 5: More on Reference Preservation in ML Systems

We just expressed optimism about anthropocentric abstraction of 'reference-preserving intentions'. We should add that a full theory will have to engage with a range of interesting differences between humans and ML systems. There are fundamental differences between programs and people in the way that information is transmitted and this will matter to whether reference chains are being preserved in 'the right way'. Here are some additional cases to consider:

(1) Suppose we have programmed a neural network on a particular computer in Oslo. That network then gets trained on lots of duck photographs. Let's assume that's enough for anchoring, and as a result, that neural network's outputs are now about ducks. We then email that program to another computer in Austin. On that computer in Austin there's now a new token-distinct but type-identical program. Is that program part of the same referential chain? Are its states also about ducks?

(2) We can easily imagine more complicated cases. Suppose we have a neural network that's been trained to recognize photographs of Pacific black ducks. We want to make a new program that recognizes photographs of eider ducks. Rather than retrain a new neural network on a new collection of eider duck photographs, we take the neural network weightings of the Pacific black duck recognizer together with a description of the typical colouring of an eider duck and apply a metaprogram that reweights a neural network to transform its recognitional sensitivities. We end up with a new program that functions well: it reliably (but not always) labels eider duck photographs as hits and non-eider duck photographs as misses. But are its reports about eider ducks? That depends on whether this more complicated method of causal transmission counts as reference preserving. More generally, programs offer opportunities for causal transmission and manipulation (by human programmers, by other programs, and so on) that aren't available with people, and a good non-anthropocentric version of externalism needs to include tools for deciding which of these opportunities are reference preserving and which are not.

(3) Suppose we are trying to do early cancer detection, so we create a machine learning cancer detector. We train it in the usual ways, giving it a sample set of cases and a scoring system on those cases. But once the program has been trained up, we also allow it over time to use data mining methods to look for additional patterns in the new cases and dynamically adjust its own categories. That means that over time, the program might end up categorizing in ways that largely disagree with the

scoring on the original training set. But we can imagine multiple ways in which this change in categorization behaviour might go. We might discover that the program has become a better cancer detector than we were—that we had made mistakes on some of the original training set, but that the program is now able to detect cancer better than we could, and is correcting those mistakes. Or we might discover that the program has become a deeper characterizer than we were. Perhaps we learn that 'cancer' is actually a confused category, one which lumps together medically distinct conditions and artificially separates other conditions that are medically similar. The program as it develops has got onto a different, more medically robust category, and is tracking that rather than cancer. Or we might discover that the program has gone off the rails entirely—that its dynamic adjustment of its own categories has drifted hopelessly away from anything that we ever wanted to track, and that it's now just tracking some random and medically uninteresting collection of blood chemistry features. In each of these cases, we're faced with the question of whether the program's outputs are still about cancer or have come to be about new categories. Answering that question, from an externalist perspective, requires determining whether the dynamic development of the program is properly reference preserving. No simple application of the Kripkean model is going to answer that question.[2]

[2] It is unclear how different this is from human cases: we can imagine a human researcher, who starts off as a straightforward cancer researcher, whose research develops in each of the three ways sketched above, but who keeps using the word 'cancer'. We're then confronted with similar questions about whether his use of the word 'cancer' is still part of the causal chain to which he was originally introduced.

(4) Lockdown is a public safety AI system, designed to assess the risks of venturing outdoors. Lockdown takes a wide variety of input data on weather, crime, epidemiology, economic markers, social media activity, and so on, and delivers a verdict of 'safe' or 'unsafe'. But Lockdown delivers location-specific recommendations. Albert, running Lockdown in Oslo, gets an output of 'safe', meaning that it is safe to go outdoors in Oslo. Beth, running Lockdown in Stockholm, gets an output of 'unsafe', meaning that it is dangerous to go outdoors in Stockholm. Lockdown's outputs are thus context-sensitive—a Lockdown output of 'safe' means, roughly, that it is safe *here*, where 'here' picks out the place being evaluated on that run of Lockdown.

There are two interrelated problems about representational content that are raised by an AI system like Lockdown. First, if Lockdown's contents are best understood as context-sensitive, stating that things are safe or unsafe in the context of utterance, what counts as the context of utterance? Second, what determines that these kinds of AI outputs are context-sensitive, rather than context-insensitive? How does an appropriately de-anthropocentrized metasemantic story predict which AI outputs are context-sensitive and which are not?

These four cases illustrate the kinds of complexity that will arise in developing a complete externalism for ML systems. One possibility here that we find quite plausible is this: many of these questions do not have predetermined answers. What will count as

a correct answer in many of these cases will depend on **decisions** we as speakers (and as speech communities) make as the engagement with ML systems become more entrenched. Maybe in 100 years, we will have developed stable patterns of how to interpret these kinds of cases.