Freewrite for Honors Thesis

Ryan Othniel Kearns, for Prof. Thomas Icard
Stanford University Dept. of Philosophy
PHIL 196: Tutorial, Senior Year
January 10, 2022

Abstract

In this thesis I propose and defend a new conception of trust, called a *trusting instance*. A trusting instance is a discrete occurrence of trust between a truster and trustee that is both action- and context-sensitive. I argue that what occurs in trusting instances can justifiably be called "trust," and explore how trusting instances can help us better make sense of trust in artificial intelligence (AI) systems and other non-interpersonal settings.

This approach differs from the mainstream philosophy of trust, which takes as a principal consideration the notion of a trusting *relationship* between a truster and a trustee. As I will argue, these notions are not fully encapsulating, given (i) many cases of trust are spurious, non-repeating instances and (ii) many artefacts we would deem trustable are incapable of having relationships (notably, AI systems).

I argue that the trusting instance is a useful conceptual tool for understanding trust in AI systems, for two reasons. First, the trusting instance admits a nice logical formulation. Second, said logical formulation makes it natural to talk about how factors like transparency and opaqueness influence trust, which are key concepts in the theory of trustworthy AI systems. While this concept cannot apply to some important cases of trust, such as longstanding trusting relationships between people, we can make sense of AI system "trustworthiness" in terms of logical quantifications over trusting instances. I would like to show how such quantifications can be epistemically meaningful.

Finally, to defend the trusting instance concept, I show how it aligns with several eminent accounts of trust in philosophy, including T. Chi Nguyen's "Unquestioning Attitude" account, Mariarosaria Taddeo's "e-Trust" account, and Jeff Buechner and Herman T. Tavani's "Diffuse, Default Model" account. Trusting instances can give a logical foundation to any of these accounts, since the basic theory is agnostic to the exact conditions where the trusting instance predicate (T) holds.

Brief sketch of the logic

To understand the trusting instance logically, we define the predicate T, which holds when there is a trusting instance. T takes three parameters: a, the truster or the one trusting, b, the trustee or the one being trusted, and x, the action that a trusts b to take. x captures our idea that a trusting instance is *action-sensitive* -- I trust my cab driver to deliver me safely, but I need not trust him to deliver a non-partisan account of today's news along the way. So, we have predicates of the form

$$T(a, b, x).$$

To capture the idea that T is *context-sensitive* we need only evaluate T *model-theoretically*. In a given logical language, we cannot have instances of $T(a, b, x)$ or $\neg T(a, b, x)$ true simpliciter, even if a, b, and x are well-defined variables. Instead, we have $T(a, b, x)$ evaluated according to a model $\mathcal{M}$, which brings along a set of *contextual variables* C. C is a set of logical sentences, and each $c \in C$ takes one of the following forms:

1. $c \equiv P(a)$ where P is a predicate. So, c is some fact about a, the truster.
2. $c \equiv P(b)$ where P is a predicate. So, c is some fact about b, the trustee.
3. $c \equiv f(b, x, C')$ where f is a function mapping past instances of b doing x in contexts C' to their outcomes. So, the domain on which f is defined records the relevant *history* of b. For example, if I am debating whether to let you fix my car, $f(b, x, C') = y$ might record that you fixed my cousin's car last week and it appears to be working fine. Here b is "you, the enterprising car mechanic," x is "fix the car," and C' is another context set -- maybe $c' \in C'$ says "my cousin's car also has a Toyota engine."

We use the notation $\mathcal{M}_C$ conventionally to denote a model where every $c \in C$ holds, that is,

$$\mathcal{M}_C \iff \mathcal{M} \models c \text{ for } c \in C.$$

Given the definitions for models $\mathcal{M}$ and context sets C, we now have epistemically meaningful statements of the form

$$\mathcal{M}_C \models T(a, b, x).$$

This formulation is made deliberately flexible, at the omission of any hard opinions on the *nature* of trust in this paper. I leave that to future work, or better, to existing accounts of trust from Nguyen, Taddeo, and Buechner and Tavani that I think are fantastic. The flexibility affords a couple of advantages I think are worth mentioning:

Trustworthiness as quantified trusting instances

We can use trusting instances to make sense of statements like "b is trustworthy," where b could be a person or even a machine learning algorithm. Statements like the form

$$\mathcal{M}_C \models (\forall a \in A)(T(a, b, x))$$

express that b is trusted to do x by a set of trusters A. A stronger statement of the form

$$\mathcal{M}_C \models (\forall a)(T(a,b,x))$$

says that b is universally trusted to do x by any possible truster, given knowledge of each $c \in C$. Even stronger, the statement

$$\mathcal{M}_\varnothing \models (\forall a)(T(a,b,x))$$

renders that contextual knowledge irrelevant, and says that any truster, without predication, can trust b to x independent of knowledge about b or b's past actions.

The strongest statement we have is

$$\mathcal{M}_\varnothing \models (\forall a)(\forall x)(T(a,b,x))$$

Or, "b is trusted to do anything, by anyone, in any context." While I can't think of a case where this would be useful (or true), it shows the flexibility that the formulation affords.

Modeling transparency as contextual predication

Another utility of the trusting instance concept is how fluidly it models statements about computational transparency. AI researchers are interested in the conditions where users might trust algorithms *in virtue of* algorithmic transparency. Borrowing from Katie Creel, there are at least two ways to have transparency about some b's ability to x:

1. You can have knowledge about b's higher level algorithm or its implementation of the algorithm, or
2. You can have knowledge about past instances in which b did x, including contextual factors.

Point 1 is captured with predicates $c \equiv P(b)$ from above, and point 2 is captured with functions $c \equiv f(b,x,C')$. Now, we have the capacity to express how instances of trust and transparency are related. Consider the following:

$$\mathcal{M}_\varnothing \models \neg T(a,b,x), \text{ but } \mathcal{M}_{\{f(b,x,C_1)=y_1\}} \models T(a,b,x).$$

This statement expresses that a does not trust b to x absent any context, yet, after seeing an example where b did x under conditions C_1, resulting in outcome y_1, a is now willing to trust. Thus, it is an example where some guarantee of Creel's *run transparency* inspired trust in a computational system for one particular truster. We can express arbitrarily more complex statements, including counterfactual statements, using this form.