Working for Trust

The philosophical literature views trust primarily as a concept that only affects human relations, because trust supposedly involves a possibility of betrayal which mere reliance doesn't (e.g. Baier 1986). Sporadically it has been argued, mostly in ethics of technology, that one actually can be betrayed by technological artefacts and a genuine notion of trust is at play (cf. Nguyen (unpublished)). These discussions overlook that computer science has been trying to give operationalizations for trust for decades. File-system permissions, user and kernel space segregation, transport layer encryption and checksums are only a few of the many examples that come to mind. But computer science also overlooks an important aspect of trust. It is modelling trust on worst case scenarios, where the environment only consists of nefarious agents with malicious intents. This has its price, as most of us aren't the vicious users imagined. Things tend to become sluggish.

As we surely are in a mixed human-machine situation (cf. Humphreys' hybrid scenario (Humphreys 2009)) and many of our daily actions rely on computer results and the proper functioning of algorithms we should be able to say when talk of trust is warrented and when it isn't. Most of our hybrid actions are so basic and simple that any talk about trust seems completely overblown. Who would think twice about not trusting a smart phone's todo list as opposed to a paper based one? But some machine results are more involved and in-transparent algorithms enter in a more essential way. Say the algorithmically curated feed in your favourite news app, or the health advice you receive from google or apple health. In research contexts machine learning methods now amend and replace older statistical techniques, for they are faster, operate on bigger data sets and make new inferences possible. To exploit this newfound inferential power on sensitive datasets like health data, comply with data protection and privacy regulations, federated learning frameworks have been introduced which promise confidentiality, integrity and availability (c.f. Warnat-Herresthal 2021). It is those situations where algorithms and human actions entangle in an essential way that call for a marriage of trust, reliability and computations.

I start my analysis by the observation that trust in human-centric as well as in human-machine situations has the function of making things smooths. It allows humans to run inferences without sceptically regressing, it allows acting without second-guessing. But for trust to have this function it necessarily needs to be tacit (cf. Lagerspetz 1998, Nguyen (unpublished)). This fact has, I think, often been overlooked in the philosophical debate on trust and it has important consequences for trust in human-machine contexts. It is instructive to look at human-only situations first before going to mixed environments. Trust certainly needs to be established in human-only situations before it can take up its function tacitly. This is the task of trust establishing practices - the first component in the dynamics of trust. But after being established, these practices can fade to the background and trust becomes tacit. Trust now is static. To give a clear cut example: Mathematicians trust and use previously established results without referring back to the original justifications, the proofs, all the time. They don't even think about themselves being in any kind of "trusting" state. The just apply the theorems.

While trust needs to be tacit to fulfil its facilitating function, machines require an explicit operationaliziation of trust. Constant work is needed to keep trust in the foreground, it needs to be computed. Using the aforementioned federated learning framework I will show how the employed

operationalization of trust directly translates to an increased need for work in the form of computing power. I argue that there is a general trade-off between trust and the need for computing power in human-machine contexts. How should we balance this trade-off? Are there situations where maybe less trust is required and therefore operationalizations of trust can be less explicit and less computationally costly? To this end I introduce trust environments. A trust environment is a human or human-machine situation where a certain level of tacitness of trust is required to achieve an end (in this function trust is similar to epistemic routines (Gelfert 2021)). It follows that the trade-off is controlled by properties of the trust environment. These properties can be manifold. Examples include expectations of the future, estimations of the environment and personal judgments. In the federated learning example mentioned above these are expectations of future attacks on parts of the computing system, security of the communication network and the importance of privacy of medical data. I conclude by emphasizing that the defining characteristic of human only trust environments, the quasi static and tacit propagation of trust after its inception, is violated in mixed trust environments. There trust, for better or worse, has to be actively kept in the foreground.

References

Baier, Annette. “Trust and Antitrust.” Ethics 96, no. 2 (1986): 231–60.

Gelfert, Axel. "Fake news, false beliefs, and the fallible art of knowledge maintenance." The Epistemology of Fake News (2021): 310.

Humphreys, Paul. “The Philosophical Novelty of Computer Simulation Methods.” Synthese 169, no. 3 (August 2009): 615–26. https://doi.org/10.1007/s11229-008-9435-2.

Lagerspetz, Olli. Trust: The Tacit Demand. Vol. 1. Library of Ethics and Applied Philosophy. Dordrecht: Springer Netherlands, 1998. https://doi.org/10.1007/978-94-015-8986-4.

Nguyen, C Thi. “Trust as an Unquestioning Attitude,” (unpublished)

Warnat-Herresthal, Stefanie, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, et al. “Swarm Learning for Decentralized and Confidential Clinical Machine Learning.” Nature 594, no. 7862 (June 2021): 265–70. https://doi.org/10.1038/s41586-021-03583-3.