ExplOrs

Explanation Oracles and the architecture of explainability

Ravikanth Pappu
CTO, In-Q-Tel

Abstract

In this paper we focus on AI systems generating conversational explanations and delivering them to humans. We ask what capabilities such systems need in order to deliver expnanations considered acceptable by humans. We abstract the machinery of delivering such explanations into what we call **Explanation Oracles** (*ExplOrs*). ExplOrs have access to domain-specific external information that enables them to reason about the input. They operate in an adversarial setting - some of the inputs they receive are designed to deceive them. The recipient of the explanation has a binary choice of accepting or rejecting the explanation. This *human binarization* reduces the problem of evaluating explanations into a decision problem, and sidesteps the sensitive dependence of explanations on the input, queries, and models used in the system. By clearly defining a system boundary for ExplOrs, encapsulating their functionality, and making the role of the human explicit, we afford ourselves the opportunity to compare different explanation approaches against each other, discover open questions, and chart improvement in XAI in specific domains over time. We are solely interested in the *architecture* of ExplOrs, not in specific implementations.

1 Introduction

Explainable AI (XAI) [1, 2, 3, 4] is a varied and complex problem and there is no agreement on what constitutes explainability or what that term even means. Among scores of others, questions ranging from: *how did this Deep Neural Network decide there is a cat in this image?* to *why was a loan denied to a specific applicant?* are included in the realm of explainability. From the literature, it is clear that the field is far from agreement on a common vocabulary, definitions, and what constitutes an explanation. However, as AI systems become more pervasive, especially in regulated environments like health and safety, there is a growing need to formalize explainability and the performance of XAI systems.

In this paper we focus on AI systems generating conversational explanations and delivering them to humans. We ask what capabilities such systems need in order to deliver expnanations considered acceptable by humans. We abstract

the machinery of delivering such explanations into what we call **Explanation Oracles** (*ExplOrs*), whose behavior is described below.

In response to an input (which is multi-dimensional in the most general case) and one or more queries about that input, ExplOrs generate and convey *conversational* explanations [5, 6] to humans. ExplOrs have access to domain-specific and external information that enables them to reason about the input. They operate in an adversarial setting - some of the inputs they receive are designed to deceive them. The ExplOr attempts to detect this deception with probability close to unity and explain it. The recipient of the explanation has a binary choice of accepting or rejecting the explanation. This *human binarization* reduces the problem of evaluating explanations into a decision problem, and sidesteps the sensitive dependence of explanations on the input, queries, and models uses in the system. ExplOrs maintain a record of unique inputs, corresponding outputs, and the human's decision for use in the future. In this paper, we are solely interested in the *architecture* of ExplOrs, not in specific implementations.

By clearly defining a system boundary for ExplOrs, encapsulating their functionality, and making the role of the human explicit, we afford ourselves the opportunity to compare different explanation approaches against each other, discover open questions, and chart improvement in XAI in specific domains over time.

We discuss related work next. In Section 3 we briefly take a look a human ExplOr. The architecture of ExplOrs are described in section 4, followed by a brief summary and future work.

2 Related work

This section is not intended to be an exhaustive review of the use of oracles in AI or XAI. Rather, we aim to provide a (very brief) flavor for work that is intellectually similar to our proposal.

At one end of the spectrum of use, oracles are described as a way to sandbox AI in [7] and [8]. To prevent a superintelligent AI from getting out of control, an oracle mechanism is used to limit AIs from acquiring too much information from the external world or affecting it in malicious ways.

Pezeshkpour et al. [9] use a GAN to generate user-friendly explanations of loan denials delivered to loan applicants (i.e., non-experts). Their focus is on the human-understandable aspect of explanations. This is an important aspect of generating conversational explanations.

Rauschecker et al. [10] describe the performance of an AI system designed to generate differential diagnoses of common and rare brain disease by looking at MRI scans of patients. By using a combination of deep learning for brain lesion detection and characterization and naive Bayesian model for differential diagnosis generation, they were able to show that the AI system approaches human expert performance. Domain knowledge was encapsulated in the prior probabilities of the Bayesian model as well as in the lesion detection and char-

acterization. The "explanation", delivered to experts, was simply a ranked list of diagnoses.

Yi et al [11] describe a system that focuses on extracting temporal and causal structure underlying videos of simple objects interacting via Newtonian mechanics. In their system, they use several different techniques for input video understanding, visual question answering, and a new oracle technique - Neuro-Symbolic Dynamic Reasoning - to predict the dynamics of the objects in the videos. Users interact with the system by asking questions. Answers to these questions can be descriptive (e.g., how many metal objects are moving?), explanatory (e.g., which shape is responsible for causing the green sphere to move?), predictive (e.g., what will happen next?), and counterfactual (e.g., which of the following will happen if the gray object is absent?). In this case, the explanation is an answer to a question formulated to narrow down possible answers. The domain and all the options in the system are fixed and carefully controlled.

We observe that in the latter three cases the domain (e.g., finance, neurology, physics) and the attendant vocabulary/ontology of that domain is implicitly contained in the models. We consider [10] and [11] to be instances of ExplOrs but without clearly defining all the implicit assumptions inherent in the systems. We address this challenge next.

3 A human example of an ExplOr

Penn & Teller: Fool Us - an American television show - inspired the idea of ExplOrs. The hosts - Penn Jillette and Teller (P&T) - are highly-experienced magicians who have performed magic together for several decades. On the show, expert magicians from all over the world are invited to perform a magic trick for P&T. If neither Penn nor Teller can explain how the trick was done to the satisfaction of the invited performer, the performer has successfully fooled P&T. For clarity, we restrict the meaning of the word *fool* to simply mean that P&T did not successfully explain how the trick was done. Figure 1 depicts a scene from the show.

This is a show about *generating conversational explanations* in an *adversarial setting* in a *specific domain of expertise*. By dissecting how P&T explain magic tricks, we discern several capabilities that an ExplOr must possess to successfully explain its conclusions to humans. The show discussed in detail in a companion blog post [12].

4 The architecture of ExplOrs

We now turn our attention to an idealized model of an ExplOr. See Figure 2 for a cartoon of the setting. We start with some definitions and notation.

- Domain D: ExplOrs operate in a specific well-defined domain D. It may be broad or narrow, but intuitively, narrow domains offer a greater chance

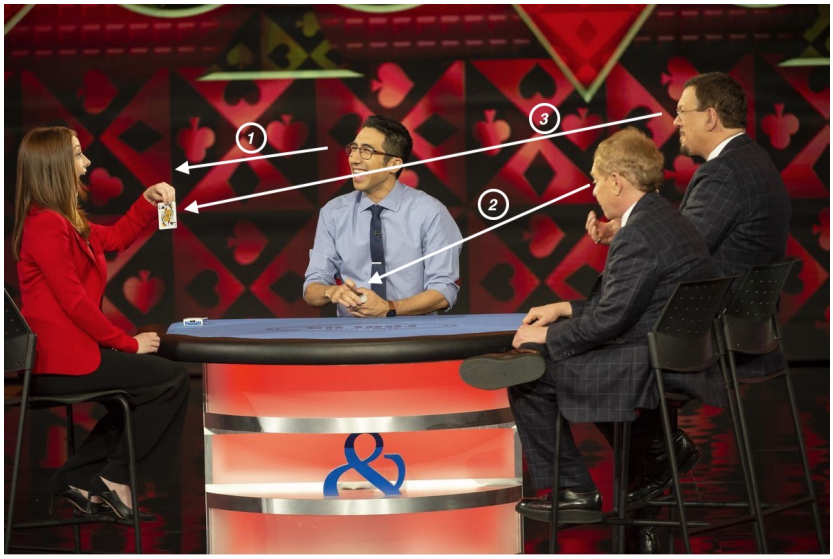

Figure 1: A scene from a performance on *Penn and Teller:Fool Us*. From left to right: Emcee Alyson Hannigan, the performer Jimmy Ichihana doing a close-up card trick, and Teller and Penn observing the performance. (1) Ichihana gets Hannigan to reveal her card eliciting utter delight. (2) Teller is paying unwavering attention to Ichihana's hands. (3) Penn is looking at the result of the effect as well.

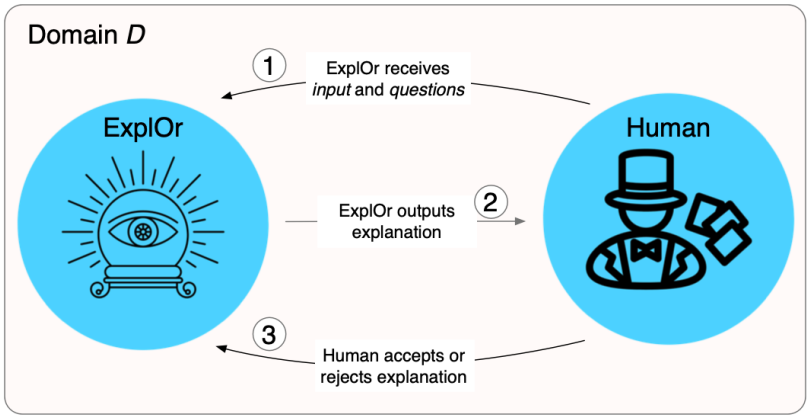

Figure 2: Cartoon model of the setting considered in this paper. The ExplOr and the human are embedded in a specific domain D. In this context, the ExplOr receives input and queries from a human, answers the questions, and the human either accepts or rejects the explanation.

of successful explanations. The domain has at least one ontology that defines and relates fundamental concepts in the domain to one another. We will assume that the ontology is represented by a labeled directed multigraph which enables reasoning over it.

- Input $T_A^k, k \in \{1, 2, 3, ...N\}, A \in \{0, 1, X\}$: In the most general case the input is a multi-dimensional array. It could be a vector of values of features or a video with an audio track. The input is composed of objects from the domain D. Each distinct input is uniquely identified with an index k so the ExplOr can keep track of unique inputs. The subscript $A \in \{0, 1, X\}$ is an indicator of whether or not the input is not adversarial, adversarial, or unknown. When provided, this lets the ExplOr know whether or not the input is adversarial. This requires an increasing amount of storage in the ExplOr over time.

  Why allow adversarial inputs at all? The greater the generalizability demanded of our models, the greater the susceptibility to adversarial input. According to [13], "*attaining models that are robust and interpretable will require explicitly encoding human priors into the training process.*" In [14], the authors create medical images which are obviously impossible (e.g., an image of a diseased retina with a scaled image of a normal retina somewhere else in the picture) which is classified as being diseased. In [15] and [16] inputs are deliberately authored to target weaknesses of systems. We want an ExplOr to be able to detect and explain these kinds of adversarial or implausible inputs.

  It is becoming increasingly evident that adversarial input examples are inevitable [17] and detecting these examples has to occur post-classification or outside the classifier itself. [18]. By explicitly allowing and keeping track of adversarial inputs (when that information is provided), ExplOrs become more robust over time. In regulated domains, maintaining a registry of known attacks is important. This is a similar idea to malware protection, where computers routinely receive updates on virus signatures as new threats are discovered in the wild and defenses against them are developed.

- Queries $Q_i^k, k \in \{1, 2, 3, ...N\}, i \in \{1, 2, 3, ...M\}$: This is a set of M questions that the human asks the ExplOr about input T_A^k. Without loss of generality, we can assume that they are strings of text. In [10] the question is *what is the differential diagnosis consistent with the MRI and patient metadata*? Unique queries are also stored in the ExplOr and accumulate over time.

- Responses $R_i^k, k \in \{1, 2, 3, ...N\}, i \in \{1, 2, 3, ...M\}$: Each query Q_i^k receives a response. Without loss of generality, this is a text string as well. As before, unique responses are stored. We note that acceptable responses can vary in content and that each combination of an input and query could have multiple acceptable responses. We note that we require responses to be conversations which could be interactive.

- Acceptance $A_i^k \in \{0,1\}, k \in \{1,2,3,....N\}, i \in \{1,2,3,....M\}$: Finally, when the human sees an explanation, they accept or reject it. A_i^k is a Boolean variable that represents this decision - 0 represents an acceptable explanation and 1 represents an unacceptable one. This binarization process allows the existence of multiple acceptable explanations.

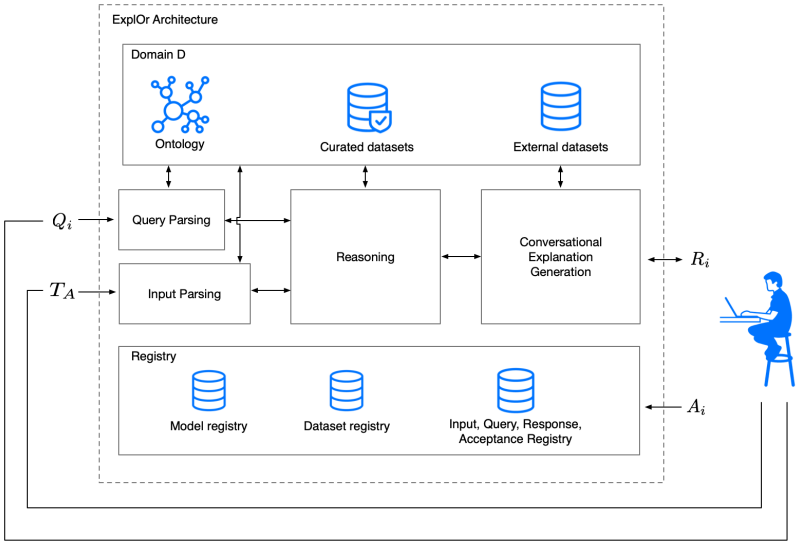

Figure 3: A sketch of the high-level architecture of an ExplOr. The dotted line clearly delineates the system boundary of the ExplOr and the role of the human is made explicit.

Figure 3 shows the high-level architecture of an ExplOr. It lives in a well-defined Domain D with an attendant ontology, a set of curated datasets about that domain, and access to external data about the world.

The *Query Parsing* module receives input queries, parses them, and makes them available to the *Reasoning* module. Similarly the *Input Parsing* module receives the input, detects and flags adversarial and out-of-distribution input. It stores the input/query pair in the registry and passes the parsed input to the *Reasoning* module, which is responsible for determining the answer to the queries given the input. It has access to all the available information in the domain D. Input and query parsing will rely on a variety of deep learning models to achieve their goals. For instance, video input could be parsed with a Mask R-CNN [19] while queries could use an attention-based seq2seq model [20]. To ensure that everything an ExplOr does is traceable, the versions of the models used and the data they were used to train on need to be recorded.

The Reasoning module is responsible for generating the explanation. Explanations are computed in many ways depending on the domain and the queries. They can range from LIME [21], Shapley values [22], and self-explaining neural networks (SENN) [23] to novel variations on decision trees [24]. [25] provides

comprehensive reference on model intepretability, which is one element of an explanation. More advanced ExplOrs will also use other forms of reasoning including causal, analogical, deductive, or inductive reasoning [26].

When the reasoning is complete, the *Conversational Explanation Generation* module takes over and conveys the explanation in to the recipient. Conveyance can be an interactive process, using, for instance, a chatbot [27] or voice assistant. This module needs to be aware of the audience (i.e., expert or layperson) and generate conversation accordingly. Recent advances in imbuing conversational agents with personality could address this requirement. Rationalization techniques [28] are also useful here.

These latter two modules are where the bulk of the computational effort and open research questions lie. The systems discussed in [29] (Figure 4) and [10] (Figure 3) implicitly reflect the architecture of an ExplOr.

Today, the quality of explanations can vary widely, and there is no universal way to compare them against each other. Explanations are sensitive to the models involved, the structure of queries, and the input. We introduce the human binarization component to sidestep this problem and reduce it to a decision problem. One way this could work in practice is to determine a threshold above which all explanations have the same utility to a human, and those below the threshold are uniformly rejected. A related idea is discussed in [30]. In many domains where AI systems augment the capabilities of human operators, this allows the human to reward or penalize the output of an ExplOr systematically.

Another important component of an ExplOr is the *Registry* module, which keeps track of unique inputs, queries, responses, and whether or not those responses were accepted. When identical input/query pairs are received in the future, the ExplOr can instantaneously respond by recalling stored information. Further, because a large number of machine learning models will be involved in query parsing, input parsing, reasoning, and conversational explanation generation, it is important to keep track of the models themselves in a model registry. In other words, the provenance of every explanation is traceable.

5 Discussion

As explainability becomes a non-negotiable requirement in many AI-enabled systems that collaborate with humans, it is increasingly important to be precise about what explanations are and how one is better than the other. We believe the concept of Explanation Oracles can fulfill this need.

ExplOrs decouple the complexity of generating explanations from how they are used. All the systems within the boundary of the ExplOr can grow in complexity or sophistication over time without affecting how they are used. As long as they are within the same domain and present the same interface, humans can use ExplOrs to augment their work. The proposed architecture also allows for the re-use of large parts of an ExplOr's infrastructure for similar use-cases in the same domain.

Two questions bear reflection here. First, can one ExplOr use other ExplOrs

as sources of explanations? There is no reason to preclude this possibility. In complex domains, explanations can take on multiple layers of complexity, and using ExplOrs designed for specific tasks as a scaffold towards the complete explanation is permitted. Interestingly, as long as an ExplOr presents a uniform interface, its internal components can be upgraded transparently. Second, why should the response of an ExplOr be trusted? The most important aspect of an ExplOr is its auditability. Every query and response, along with all the metadata, is recorded for future use. This enables auditability. The track record of the ExplOr is available for inspection, and the human who receives the response has the option of rejecting it if not acceptable. This binarization of the output is a safeguard, but it assumes that the human is not colluding with the ExplOrs.

The presence of the human in the system raises questions of scalability. How useful can ExplOrs be if they rely on humans-in-the-loop? One possible approach is to develop ExplOrs with humans until the utility of the explanations generated is above a threshold and then deploy it into production. Subsequent to this, only unacceptable explanations need to be examined in order to improve the ExplOr.

This architecture also suggests that it could be possible to standardize on specific of ExplOrs in well-defined use cases. When explanations become legally required, it makes sense for multiple entities to use an identical instance of an ExplOr (perhaps even an ExplOr as a service) in order to mitigate the costs and complexities of developing and maintaining them.

Having proposed the concept of an ExplOr in this paper, we will now turn our attention to developing proof-of concept-implementations to understand the practical challenges of implementing them.

Acknowledgements

This paper has benefited greatly from discussions with and suggestions from Wendy Plesniak, Andrea Brennen, George Sieniawski, Bob Gleichauf, Zach Lyman, Pete Tague, Carrie Sessine, Kinga Dobolyi, and Vishal Sandesara.

References

[1] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable Deep Learning: a Field Guide for the Uninitiated. *CoRR*, 2020.

[2] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. March 2017. http://arxiv.org/abs/1702.08608.

[3] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining Explanations in AI. *CoRR*, 2018.

[4] A Brennen. 10 Things I've Learned About Explainable AI. https://medium.com/high-stakes-design/10-things-ive-learned-about-explainable-ai-e7f963d6bfc2, Last accessed on 2019-12-30.

[5] Bob Gleichauf. A Chatbot? Are You Sirious? https://gab41.lab41.org/a-chatbot-are-you-sirious-9a7a615b3cfa, Last accessed on 2020-04-10.

[6] Denis J. Hilton. A Conversational Model of Causal Explanation. *European Review of Social Psychology*, 2(1):51–81, January 1991.

[7] Nick Bostrom. *Superintelligence : Paths, Dangers, Strategies*. Oxford University Press, New York, 2016.

[8] Stuart Armstrong and Xavier O'Rorke. Good and Safe Uses of AI oracles. *CoRR*, 2017.

[9] Pouya Pezeshkpour, Ramya Srinivasan, and Ajay Chander. Generating User-friendly Explanations for Loan Denials using GANs. page 9.

[10] Andreas M. Rauschecker, Jeffrey D. Rudie, Long Xie, Jiancong Wang, Michael Tran Duong, Emmanuel J. Botzolakis, Asha M. Kovalovich, John Egan, Tessa C. Cook, R. Nick Bryan, Ilya M. Nasrallah, Suyash Mohan, and James C. Gee. Artificial Intelligence System Approaching Neuroradiologist-level Differential Diagnosis Accuracy at Brain MRI. *Radiology*, April 2020. Publisher: Radiological Society of North America.

[11] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision Events for Video Representation and Reasoning. *CoRR*, 2019.

[12] Ravikanth Pappu. Penn & Teller in a Box. https://www.iqt.org/penn-teller-in-a-box/, Last accessed on 2020-07-26.

[13] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples Are Not Bugs, They Are Features. *CoRR*, 2019.

[14] Samuel G. Finlayson, Hyung Won Chung, Isaac S. Kohane, and Andrew L. Beam. Adversarial Attacks Against Medical Deep Learning Systems. *CoRR*, 2018.

[15] Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. CODAH: An Adversarially Authored Question-Answer Dataset for Common Sense. *CoRR*, 2019.

[16] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. *CoRR*, 2018.

[17] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are Adversarial Examples Inevitable? *CoRR*, 2018.

[18] Yao Qin, Nicholas Frosst, Sara Sabour, Colin Raffel, Garrison Cottrell, and Geoffrey Hinton. Detecting and Diagnosing Adversarial Images with Class-Conditional Capsule Reconstructions. *CoRR*, 2019.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 10 2017.

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, 2014.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust You?: Explaining the Predictions of any Classifier. *CoRR*, 2016.

[22] Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *CoRR*, 2017.

[23] David Alvarez Melis and Tommi Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7786–7795. Curran Associates, Inc., 2018.

[24] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E. Gonzalez. NBDT: Neural-Backed Decision Trees. *CoRR*, 2020.

[25] C Molnar. Interpretable Machine Learning: A Guide to Making Black Box Models More interpretable. https://christophm.github.io/interpretable-ml-book/, Last accessed on 2020-04-29.

[26] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. Designing Theory-Driven User-Centric Explainable AI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, - 2019.

[27] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for Building an Open-Domain Chatbot. *CoRR*, 2020.

[28] Upol Ehsan, Brent Harrison, Larry Chan, and Mark O. Riedl. Rationalization: a Neural Machine Translation Approach to Generating Natural Language explanations. *CoRR*, 2017.

[29] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-Symbolic VQA: Disentangling reasoning from vision and Language Understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2018.

[30] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. Optimizing AI for teamwork, 2020.