Adversarial collaboration

Paul Christiano Follow
Dec 29, 2014 · 6 min read

Suppose that I have hired a group of employees who are much smarter than I am. For some tasks it's easy to get useful work out of them. For example, suppose I am interested in finding a good layout for the components on a chip and can easily evaluate the quality of a proposed layout. Then I can solicit proposals, test the proposals, and award the employees according to the results.

However, for some tasks there may be fundamental problems with such an incentive-based policy. For example, suppose I am interested in hiring these employees to design an AI which is smarter than any of us and will act autonomously in the world. In cases like this, I will have to try something different.

The challenge

In this case there is no clear way to evaluate the performance of my employees even *ex post*. I can build whatever AI they propose and test its performance in a range of environments. But these tests could never incentivize my employees to make an AI that shared *my* values, rather than one that shared *their* values: an AI that shared their values would be motivated to maximize the reward given to my employees—including looking *as if* it shared my values, as long as I was watching.

In this context, rather than blindly testing proposals I should understand them first, and reward my employees for making proposals which I can understand and which are clearly good. Even if I can't think of a way to design a better AI, I may be able to recognize a good idea when I see one.

But this approach focuses my employees' ingenuity on finding proposals which will be *convincing to me* rather than *actually good*. I can try to remove any other incentives for my employees, but even then I should worry that most compelling ideas aren't actually good. This becomes increasingly problematic as the ideas become more complicated, and subtle problems become more and more likely.

Maybe we could just be so careful that it didn't matter; we could hold a proposed idea to such a high standard that if it passes, it really must be good. In practice, this could require so much vigilance that it would be harder than coming up with the idea yourself.

The idea

Instead you could consult not one but two groups of employees: team A, whose goal is to provide an idea that you think is good, and team B, whose goal is to convince you that it isn't. This might be a good idea even if you do have a clear performance metric, to help identify its possible shortcomings, efforts to game the system, or unintended consequences of an idea.

(Parts of this proposal are based on discussion with Katja Grace. If you haven't figured it out by now, the whole thing is a metaphor for the AI control problem.)

The rest of this post will describe some issues with and elaborations on this proposal.

Capabilities: Ideally team B would be smarter, know more, and have longer to deliberate than team A. How much smarter? The tradeoff depends on the relative costs of taking longer vs. incorrectly accepting a bad idea. If an idea is good enough, then we might hope it will look good no matter how smart team B is.

In particular, it would be great if team B could know everything that team A knows, and could eavesdrop on as much of their thought process as possible. This would make it very difficult team A to be deliberately misleading. In the case of machine intelligences, it might be possible for the two teams to literally share the same mental state.

Weakening the critic: It may be *too easy* to convince someone that an idea is flawed. If team B is much smarter than team A, then they may always be able to find some reason that team A's suggestion is suboptimal. I don't think this ia big deal. We can:

- Allow several rounds of rebuttals, in which team A can defend their idea against criticisms, or can change their ideas to address flaws.
- Force team B to find significant flaws. If team B can improve at all on team A's plan, that suggests that team A's proposal isn't

the best; but at some point we should just accept it anyway if team B can't find serious problems.

It's worth remembering that team B might be motivated to try any trick at their disposal to convince us that the proposed idea is no good. They will claim that the problems are serious but too subtle to easily explain, will make desperate emotional appeals, will try to scare or threaten the implementer, and so on. To cope with this, we might call some outcomes a tie—we could refrain from using the idea, but still decline to give team B a reward.

Problematic incentives: Ulterior motives for either team could create problems. Team B might go soft on an idea which they want us to accept (and knowing this team A might propose an idea that team B will like). So we should ensure the incentives are large enough that both teams care about winning much more than they care about whether wee accept their ideas.

In particular, if we are going to do multiple rounds of back-and-forth, we should use independent teams. Team A shouldn't be able to get more reward if their original plan is shot down. Instead we should hire a new team A'.

Preemptively defend against criticisms?: we could set up the game in two extreme ways.

- Team A is rewarded when they propose a plan which sounds good. This may lead to many rounds of rejections, as team A constantly makes good-sounding but flawed proposals, since team A has no motivation to fix hard-to-spot flaws.
- Team A's goal is to make a plan which survives scrutiny.

The first option is better, if it works. If team B can't find any holes in a generic idea that sounds good to us, we should feel pretty comfortable. We should be (slightly) less comfortable if team A engineered their plan so that team B couldn't explain its problems to us.

One could iterate the first option several times, with each round leaving us increasingly savvy to possible flaws in team A's plan. Eventually this starts to run into the same concerns with overfitting.

Team C: We could try to capture the best of both worlds by introducing a third team C, which is better-equipped than either team

A or team B and operates in one of two ways:

- Team A attempts to produce plans that will survive criticism by team B. After a proposal survives criticism by team B, team C enters and attempts to identify further flaws. Team A does not care whether or not its plan is overturned by criticism by team C.
- Team A attempts to produce plans that will be superficially plausible, without concern for whether they will survive criticism by B. Before team A makes a suggestion, team C tries to provide coaching which minimizes our probability of inadvertently accepting a "bad" plan. That is, team C is penalized if we accept a plan from team A which is later overturned by criticism from team B.

Other mechanisms: this approach could be combined with other mechanisms designed to get useful work out of much smarter employees. For example:

- Bootstrapping: ask smart employees how to build more effective systems for getting information out of smart employees.
- Throttling: work with employees who are smarter than you but only by the minimum amount necessary to get the job done.
- Restricting the range of proposals: limit team A's options, and stick with what we can most easily understand.
- Self-verifying proposals: ask for ideas that help verify their own desirability. (Some ideas can fail in subtle other ways; others are demonstrably bad if they are bad at all. Encourage team A to look for the second kind of ideas.)
- Ex post verification: we can give out the reward only after we have seen an idea in action. This ex post verification can be assisted by team B, who can help us design good tests and think of possible shortcomings we wouldn't have looked for.

None of the ideas in this post are silver bullets. And if they were needed, the would-be employer would hopefully spend more time thinking about the problem than I have—and they would have more contextual information.

The point of this post is to help flesh out our understanding of how hard it is to delegate high-stakes problems to very smart, very

cutthroat reward-seekers. The bigger goal is to better understand how hard the AI control problem is (and where the largest difficulties lie).

. . .

Originally published at ordinaryideas.wordpress.com on July 18, 2014.