Universality and security amplification

Paul Christiano Follow
Mar 10, 2018 · 10 min read

If many copies of Paul collaborate to solve a problem, I expect their behavior to be corrigible: each copy will make an honest effort to help the group accomplish its goal, to help the group notice and correct mistakes, and so on.

Sometimes by chance one copy may behave incorrigibly. For example, one in every million days I might be struck by an irresistible urge to watch the world burn. Reliability amplification is the problem of combining several probably-corrigible agents into a more-corrigible ensemble (which only wants to destroy the world on one in every *billion* days). I think reliability amplification is probably possible, because a large ensemble of sufficiently-corrigible agents is very unlikely to have more than one incorrigible member.

Unfortunately, some situations might cause me to consistently behave incorrigibly. Let's call such a situation an "attack." For example, a clever argument might convince a copy of Paul that he should sabotage the collaboration. If we expose ten copies of Paul to the same argument it won't fix the problem, we'll just have ten copies all of whom are behaving incorrigibly.

Security amplification is the problem of combining many corrigible agents, into a group which is harder to attack than any individual. I hope to iterate security amplification and end up with a secure system.

This post explains why I think security amplification is likely to possible.

Setting

I'll be thinking about dialog systems that communicate over a text channel. A "query" is anything that you can ask interactively over a text channel.

In this context, corrigible behavior means behaving cooperatively to the person on the other end of the conversation—being honest, volunteering helpful information, following their instructions, *etc*.

An “attack” is a sequence of messages which cause an agent to stop behaving cooperatively.

I. Breaking it down

The basic hope

A large group can answer a big query by breaking it down into less complex pieces. Let’s suppose we had some formal measure of complexity, and that we can answer a query of complexity k by breaking it into pieces each of which has complexity $<k$, until we reach some set of “foundational” queries of complexity k^0 that are too simple to break down any further.

This suggests a route to security amplification:

- Let k^A be the complexity of the *simplest possible attack* on an individual.
- We hope that $k^A > k^0$. That is, we hope that the foundational questions are simple enough that all of them are safe.
- Then if we give any input of complexity k^A to the group, it will be broken down into pieces of complexity $<k^A$ (which are necessarily safe).
- Therefore the complexity of the simplest attack *on the group* is higher than the complexity of the simplest attack *on the individual.*
- If we iterate this process, building a group out of groups, then the simplest attack will get more and more complex, until eventually there are no attacks at any feasible level of complexity.

In order to realize this hope, we need:

1. to define a procedure for breaking queries down,
2. to define an appropriate notion of complexity,
3. to argue that all queries above some foundational complexity k^0 are broken down into strictly simpler pieces,
4. to argue that all queries below complexity k^0 are safe,

5. to argue that the group behaves corrigibly as long as all of the individuals behave corrigibly.

1. Meta-execution

Meta-execution can potentially answer a query Q without any individual needing to consider anything as complex as Q itself.

I've sketched out part of an example here, to give some sense of what this actually looks like.

2. Defining complexity

I'll start by imagining what would happen if you asked me to pick a query at random.

By "at random" I'm imagining something like the benign universal distribution relative to Paul: I think of all of the possible not-actively-evil strategies I could use for picking queries at random, and then I pick one of those *strategies* at random.

For example, my meta-strategy might be: with 5% probability I give a uniformly random string. With 2% probability I think of events in my life and ask a question relevant to one of them. With 15% probability I try splitting up the space of possible questions into a hierarchy, and flip a coin at each step to decide which part of the hierarchy to explore. With 80% probability I do something else.

One of these strategies is particularly important: I could generate an intermediate random query Q′, and then let Q be a random query that arises in the process of answering Q′. For concreteness, let's say that I use this strategy half of the time.

Let μ(Q) be the probability that this process outputs query Q. We can define the complexity of a query Q as k(Q) = −log μ(Q).

We can also use μ to measure the complexity of a distribution η over queries, by letting k(Q) be the maximum value of log (η(Q)/μ(Q)) for any query Q.

Informally, this is the log of "How many samples would you have to draw from μ, in order to get one sample from η?", i.e how many bits of selection pressure you need to get a sample from η.

(It may be more useful to consider the ε-smooth complexity: what is the *smallest* complexity of any distribution η′ that is within ε total variation distance of η? Informally: how many samples would you have to draw from μ, in order to get a sample ε-close to η?)

Note: There's no way that this is the right definition to use and I expect we'll be able to do better. I think an important ingredient is having a decomposition where the (subquestion → question) graph is a small set expander. If anyone wants to take up this analysis, I think it's an important question, I expect to come back to it at some point.

3. Can we break a query into simpler parts?

Suppose that I start with a query Q, and split off a subquery Q′. When can we hope that $k(Q') < k(Q)$?

Suppose there are many possible queries Q for which Q′ appears as a subquery. For example, the subquery "when did [*a*] happen?" might appear in the course of answering "was [*a*] a cause of the Great Depression?", or it might occur in the course of answering "was [*a*] a consequence of the Spanish flu?"

Each of these corresponds to a way that Q′ could be sampled by μ—since one of the strategies in μ is to generate a random query and then generate a random subquery.

So $\mu(Q') > \Sigma\mu(Qi) / 2N$, where {Qi} is the set of superqueries that give rise to subquery Q′, N is the total number of subqueries for each superquery, and 50% is how often μ decides to sample a random subquery.

As long as each possible superquery has at most 1/(2N) of the total probability under μ—i.e., as long as you can't guess the superquery with reasonable probability from the subquery—then we have reduced the complexity. If N is 16, this requires that each subquery must throw out about 5 bits of information about the original query.

In fact the situation is better than this:

- In reality our queries are sampled from some distribution η rather than chosen deterministically. As we increase the entropy of η the set of preimages {Qi} effectively becomes larger. (The analysis also gets a little bit more complicated.)
- μ mixes in strategies other than (pick a random query, then pick a random subquery). If Q′ is qualitatively simpler than Q, then

> we may have μ(Q′) > μ(Q) without even considering the possibility of Q′ being sampled as a subquery of Q.

4. Are simple queries secure?

What is the complexity of the simplest attack on an unaided human? Equivalently: how many queries do we expect to sample, before we find an attack?

This will vary from human to human, so for concreteness I'll consider a thoughtful futurist or AI researcher who has been briefed about all of the issues we can think of.

If we wanted to actually test for the existence of attacks, we could conduct a giant brute force search for "simple" attacks.

If we imagine generating a query every 30 seconds for 40 hours a week for a year, we generate about 200,000 queries a year. Hiring ten people gets us up to 2M queries a year. 30 seconds may be low, but note that the generation process can produce many samples per invocation, and we only need to test the candidate attacks.

If we put in a little bit of selection pressure I suspect we would find attacks at least 1–3 orders of magnitude more often than by sampling randomly.

So it seems easy to test explicitly whether 100 million random queries contain an attack. This could be done by dedicated red teams, and mixed freely with the process used to collect training data. I suspect that thoughtful analysis could get us even further.

Intuitively, I would be surprised if such a search turned up an attack against a careful overseer. If it did turn up attacks, I suspect it would be possible to strengthen the overseer against them.

Overall, I feel pretty safe relying on 10^8 queries not containing an attack, pretty unsafe with numbers like 10^{30}, and am not sure where in between attacks would crop up.

5. Is a collaboration of corrigible agents corrigible?

Even if every individual behaves corrigibly, the group need not behave corrigibly. This is a bit of a tangent from the real thrust of this post, which is about removing the security failures of the human. But it's a tangent worth discussing briefly, since I think it's an important open question.

There are two very salient sources of incorrigibility (though without a crisper picture of optimization and corrigibility this list isn't exhaustive):

- The group is applying selection pressure to patterns and behaviors propagating through the group. Those patterns may themselves become powerful incorrigible consequentialists.
- The group is implementing incorrigible optimization.
  For example, suppose meta-execution asks the subquestion "What does the user want?", gets a representation of their values, and then asks the subquestion "What behavior is best according to those values?" I've then generated incorrigible behavior by accident, after taking innocuous steps.

I think the first problem is likely resolvable by applying security and reliability amplification, which operate on groups as well as individuals. I think the biggest thing that's missing here are tools for analysis (though maybe once we have those tools we'll notice more problems). Currently this feels pretty complicated and confusing to me.

The second problem clearly requires the human operator to do something other than just "being sensible." That is, in order for this scheme to possibly work, the human needs to understand something about what kind of optimization causes incorrigibility and the human would have to avoid actions that introduce problematic optimization. It's conceivable that this can be done by an existing human if they are paranoid, but realistically the only reason I expect this to go well is because I expect us to make progress on the theoretical issue of understanding incorrigibility. I think this is an important theoretical problem.

II. Universality

So far I've avoided a key question: why think we can break tasks down at all?

Indeed, some human capabilities *can't* be broken down into pieces. Suppose that a human has seen a bunch of examples $(x, f(x))$—such as English phrases and their French translations—and has learned a predictive model of f. The human may not be able to access their model except by running it, in which case they have no hope of breaking down the task of computing *f*—in this case translating a sentence. (Wei Dai proposed the translation example.)

The proposal

I propose to go on breaking tasks down anyway. This means that we will lose certain abilities as we apply amplification.

For example, given the task "Translate the sentence [*x*] from French to English" we will answer it without having any translator look at the entire sentence *x*. This means that the quality of translation will fall.

After enough steps of amplification, we may eventually arrive at an agent that doesn't know French at all, and is stuck with recommendations like "Consult an English-to-French dictionary."

Effectively, this proposal replaces our original human overseer with an impoverished overseer, who is only able to respond to the billion most common queries.

Is this OK?

The first key question is whether this impoverished overseer remains universal.

That is, if we put together enough copies of this impoverished overseer (by iteratively apply meta-execution) would we be able to obtain arbitrarily smart groups? Or would we get stuck?

Here we need to be careful about "arbitrarily smart." There are clearly problems the group will never be able to solve—due to lacking knowledge/expertise— including problems that individual humans *can* solve.

This is potentially OK, as long as we learn a good policy for leveraging the information in the environment (including human expertise). This can then be distilled into a state maintained by the agent, which can be as expressive as whatever state the agent might have learned. Leveraging external facts requires making a tradeoff between the benefits and risks, so we haven't eliminated the problem, but we've potentially isolated it from the problem of training our agent.

Comparison to agent foundations

If the impoverished overseer is universal, then the set of questions of complexity $k < k^0$ form a simple “core” for reasoning: by creating a giant lookup table of human responses to these questions and simply using that lookup table enough times, we can produce arbitrarily sophisticated behavior.

If humans are universal at all then of course such a core exists (just take all questions that a human can articulate in their lifetime). But finding a small core seems like it requires a better understanding of intelligence.

I think that coming up with such a core is a very natural and important problem for researchers interested in philosophical issues in AI. My view is that if MIRI-style research adds value, it is most likely to be by finding an explicit core for reasoning rather than finding an explicit recipe for AGI. This core would then be combined with iterated amplification to yield a competitive AI. However, I don’t think that such a core is likely to encode an answer to questions like “what is the right decision theory to use?”—instead I expect it to look more like a solution to metaphilosphy, automating the process whereby humans answer questions of that form.

Conditioned on amplification working well, I think there is about a 50% chance that it uses an explicit core that we understand and a 50% chance that it uses a messy core learned from humans.

In addition to making it much easier to analyze amplification, having an explicit core of reasoning would also potentially make verification much easier, as discussed here. Overall, I think that this kind of perspective might capture most of what is useful about the MIRI view while still being able to leverage the benefits of modern ML.