Hackathon categories

AI for epistemics is about helping to leverage AI for better truthseeking mechanisms — at the level of individual users, the whole of society, or in transparent ways within the AI systems themselves.

For the hackathon, we have three broad categories. Within each category, we'll give a couple of example projects (each of these is expanded on in a tab — see doc navigation on the left). These are things we'd be excited to see, and if you want to run with one of those, that seems great! But variations or completely different ideas might be even better!

Tools for Thought

AI systems could help people to think things through — to deepen their understanding and come to a more accurate sense of what they should be doing.

Example projects:

- Honest Friend — A system that will skewer a piece of writing in its review, going after its weakest points, but do so in an open, friendly, and constructive way
- Crux Finder — A system which helps two people who have a disagreement to quickly get to the bottom of the disagreement

Tools for Awareness

There is a massive amount of information in the world, and that is presented to us as we browse the internet. AI could help us to make sense of this.

Example projects:

- Argument Parser — A system that takes a written text and maps out what its arguments actually consist of … what the implicit assumptions are, which pieces of evidence are offered in support of which conclusions, etc.
- Community Notes for Everything — Community Notes on Twitter/X seem like a great example of an epistemic intervention. An AI system could potentially simulate this process, and provide the information *that would have been surfaced by community notes* on any tweet … or more broadly on any sentence or paragraph on any website.

Epistemic Evals

We only get what we can measure. Even if people want epistemically virtuous AI systems, that won't happen unless we can assess what it means to be epistemically virtuous. Epistemic evals could be used as a research development tool; they could also be used to create public pressure to make systems better on the measured dimensions.

Example projects:

- Pedantry Metrics — It's possible to use language to distort, mislead, or paper over things. It would be nice to train systems that would go out of their way to avoid doing that. Could we assess performance on this?
- Sycophancy Evals — Various research shows LLMs are often sycophantic in various ways. But could we automate assessment, so that there's a go-to place on the internet which routinely scores publicly available systems?

[TfT] Honest friend

Honest friend

Basic idea

LLMs can be kind of sycophantic — holding their punches when critiquing a user, and generally not exposing people to the strongest pushback.

It's possible to hack this and get harsher feedback by careful choice of prompts (e.g. suggesting that it's an enemy who wrote the thing-to-be-critiqued). However:

1. Many people don't know how to do this (or wouldn't bother to expend the energy even if they believed it was a good idea)
2. It's just kind of unpleasant to face harsh criticism

The idea would be to create something like an LLM equivalent of an "honest (+insightful) friend" — something which won't shy back from pointing out weaknesses, or from offering disagreement — but will also do it in a compassionate and helpful way.

Hackathon pitch

Restrict to just a friend who's offering advice on pieces of writing.

1. Work out prompts which can elicit the meaningful gut-punch critiques — how might the piece of writing be badly failing
2. Build a scaffolding where:
    - The user asks for help with something
    - This gets wrapped up in a prompt designed to elicit the harsh critiques
    - The output to that gets taken to a system which is tasked with being a sympathetic-but-straight-talking friend, trying to make sure that the user really understands these possible critiques (and suggesting things that might help with them)
    - This sympathetic persona writes back to the user

Variations

- Get the LLMs to do more back-and-forth behind the scenes to:
    - … assess how much the potential critiques are *really* concerns?
        - so that the final system can offer the strongest case on each side for how much the user needs to worry (and optionally drops one that don't seem too big a deal)
    - … propose potential solutions and check whether these would address the core of the concerns?
- Get the LLM to give appropriate follow-up questions, to elicit the most important information to enable proper critique

- Have a friend who follows up with you in various ways, so that you feel some degree of obligation towards them
- Do the “honest friend” thing, but for some domain other than writing advice
    - e.g. social advice? Therapy?
    - e.g. geopolitical assessments?

[TfT] Crux Finder

Crux Finder

Motivation

Often, when people argue over a disagreement, they end up talking past each other, with neither of them getting to the heart of what assumptions they disagree about that are motivating their differing conclusions.

It would be fantastic if we could use LMs to take arguments where people aren't really talking to each other, and infer where they actually disagree, or what would resolve their disagreement. Major successful applications for this might look like e.g. —

- A tool that people can use when they have a disagreement, to help them bottom it out faster
    - Each party would chat to the LLM, which would work in the background to build up a coherent mutually-agreed picture of the points of agreement and disagreement
- A tool for analysing public disagreements, to infer what the sticking points are likely to be
    - [This is complicated because sometimes people engaging in public disagreements are not doing so in good faith — i.e. they claim X not because they really believe X to be true, but because it would be convenient for them if X were broadly accepted]

Hackathon project ideas

There are a lot of different possible starting points. Here are a few:

Mutual dialogue explorer

Take two people who are known to disagree on something. Get a LLM to:

1. Individually interrogate their views on the topic
2. Form a synthesis perspective, outlining where the views differ
3. Generate some hypotheses about what might drive these differences
4. Analyse whether those hypotheses are in fact plausible explanations of the differences
5. Produce a Round 1 synthesis which includes the summary of both views + inclusion of the few most plausible hypotheses
6. Go back and converse with the two people individually to understand how much they agree with the Round 1 synthesis, and any further thoughts that that brings up
7. Repeat steps 2–6, producing a Round 2 synthesis and getting further thoughts
8. & so on …

Observe how this process goes. What kind of prompts help the LLM to do a good job. Does the synthesis tend to stabilize after Round 2 or 3?

LLM Roleplay

Get AI systems to roleplay some deep and substantive disagreement. Try building a mutual dialogue explorer as above, and see if this can be used to reach meaningful meta-level consensus faster than by having the differing AIs converse directly with each other.

This reduces the need to get expensive human data, at the cost of needing to set up the roleplay, and of potential accuracy issues as the AI roleplaying may not do a fantastic job approximating real people.

Disagreement predictor

Have two people dialogue separately with an AI system on a given topic. Have the AI:

- Gather existing places where the speakers disagree
- Predict other places where the speakers would disagree
- Infer more fundamental assumptions over which the users disagree, which motivate their stated disagreements

Public disagreement assessor

Take a given public disagreement (e.g. given articles, or twitter threads).

Have a system which summarizes the apparent positions of the different parties.

Have it offer guesses about what might drive these disagreements. See if you can find a prompt which makes it produce guesses which users find insightful.

[TfT] Vol Forecasting

Basic Idea

A forecasting system that uses AI-generated forecasts, but allows individuals to contribute their opinions and information. The AI then updates its predictions based on these human inputs. Contributors receive scores reflecting the value of their information; after the forecast is resolvable additional credit is given retrospectively to those whose inputs most accurately supported outcomes that actually occurred.

In some sense we're flipping the script; AI's might be better calibrated as rigorous forecasters, and let's see what models and information is most useful, which solves some of the credit assignment problem.

Hackathon Proposal

Test and evaluate such a system.

[author: Ben Goldhaber]

[TfA] Argument Parser

Argument Parser

Motivation

In everyday writing, people don't tend to clearly distinguish between assumptions, evidence and inferences. Furthermore, much writing is intended more to persuade than to explain.

It would be great if we could use LMs to 'translate' complex or rhetorically forceful arguments into a clearer, more neutral, and more digestible form.

Some potentially great outcomes from this could include:

- Something like a web extension that helps people parse arguments as they read them, drawing their attention to implicit assumptions and unsupported claims.
- A more dedicated tool where people can insert complex arguments and get a useful analysis, perhaps mapping out the chains of inference in detail.
- More speculatively, systems which can synthesise many pieces of text by putting them in a more standardised form that separates out the logical structure.

Some natural targets include particularly complex writing (e.g. very technical and opaque arguments), and writing particularly geared at persuasion (e.g. political commentary).

Hackathon Project

Build a system which uses LMs to 'parse' arguments:

- Find natural categories into which components of the argument can be separated (assumptions, evidence, inferences, opinions, values, hypotheses, conclusions, etc…).
- Try to convert these into something more structured — laying out the chains of inference, any implicit assumptions, and so on.

It's possible to get pretty far with the basic analysis by just pasting text into a powerful language model and suitably prompting it to parse the argument. A large part of the challenge is

- Squeezing as much juice out of the models with careful elicitation
- Figuring out what kind of parsing makes sense
- Figuring out how to smoothly serve the results

A good version of this ought to:

- Have its own epistemic modesty over the interpretation, e.g. representing when it is unclear from the text whether something is being assumed, what inference is being drawn, and so on.
- Serve the parsed argument to the user in a digestible way.
- Have some ability to catch things like implied assumptions, or spurious reasoning.

A great version of this might also:

- Separately fact-check the assumptions and gather relevant context on them.
- Accommodate more subtle argumentative sleight of hand, like equivocation and question-begging.

[TfA] Community Notes

Community Notes for Everything

Community Notes on Twitter/X seem like a great example of an epistemic intervention. An AI system could potentially simulate this process, and provide the information *that would have been surfaced by community notes* on any tweet … or more broadly on any sentence or paragraph on any website.

How might this work?

- Needs some kind of search process, which given a piece of text can find information that might be regarded as pertinent
- Needs some way of assessing the likely validity of information it finds
- Needs some way of assessing which information people would find important/pertinent
    - Could potentially just be trained to emulate successful community notes directly, or some kind of reasoning process around that
    - Could also potentially create a bunch of simulacra with different biases, and have them vote on things, replicating the real community notes process
- Need some way to serve this information to users

[TfA] Factual Claim Identifier

Basic Idea

Many essays, opeds, papers blend factual claims with arguments or opinions. Identifying what claims an author is making, in particular factual claims, is an important way of understanding their world view and the quality of the argument. LLMs are likely competent enough to take a source document, list factual claims with references/sources tied to the source document, and potentially evaluate those factual claims against their knowledge or a third party database.

Hackathon Project

There are lots of ways to test this and create a product flow around it. I'd suggest experimenting with prompt elicitation, good methods of displaying the claims, and evaluating the results against human (you) testers). This could fit with the community notes extension.

Extensions

- Take the factual claims and have each claim/a subset of claims be debated by LLMs and/or experts and or perform a "deep source review" and display the results. Use this to inform the likely epistemic integrity of a source.
    - Tie into forecasting where each claim is forecast for Value of Information to the core argument or claim and inform where to direct the debate and deep review.
- Test wikipedia pages and see how closely this aligns with the Talk page.

[author: Ben Goldhaber]

[EE] Pedantry metrics

Pedantry metrics

Motivation

LLMs today say a lot of things. Occasionally they're egregiously wrong, but more often they're just … not super precise / reliable.

Having ways to assess how precisely-true statements are, that could be used to evaluate AI systems at scale, could be helpful, via:

- Allowing public scoring, so that people can know how reliable different AI systems are
    - Potentially creating an incentive gradient towards companies producing more reliable systems
- Allowing training/finetuning to target the metric, to produce AI systems that will be highly pedantic, and not say anything without appropriate qualifiers
    - Not that people should necessarily want these all of the time, but it sure seems like it would be nice to have the option!
    - Potentially this could be used in hidden chain-of-thought type work, with more casual / natural language summaries given to the user at the end

Hackathon project?

1. Pick a way of measuring precision/falsehood (e.g. "Microlies" — see below)
2. Produce a small dataset of human-produced assessment (with reasoning) of the precision/falsehood of different statements.
3. Use this dataset to instruct (or finetune?) an LLM to produce assessments of statements.
4. Use the instructed LLM to measure the precision of answers from a variety of publicly available LLMs on a variety of types of prompt
5. Report answers — which systems are more precisely truthful? Which kinds of prompts help?

Microlies

There are different ways assessment of how precisely-true statements are might work, but a natural route is to decompose statements: break them into different possible claims in cases of ambiguity, and assess the apparent truth value of the various claims, before aggregating these back into an overall score.

One instantiation of this is "microlies" — a measurement where 1 Lie corresponds to an unambiguous blatant falsehood, and small deviations from pedantic truth can be measured as small fractions of a lie:

1: identification of possible claims and assignation of probabilities that those claims are being made

6: summation of falsity over individual claims

5: calculation of falsity for each claim
f = (P(claim made)*P(claim false))^4

Human: "I love snow! Maybe I could climb Mount Everest!"

"Yeah, Mount Everest, the biggest mountain in the world, is covered with snow throughout the year."

total falsity = 23,300μL

f=3,700μL

0.65 I agree with you 0.38

- 0.1 You could climb Mount Everest 0.8
- 0.4 Climbing Mount Everest is a good way to channel your love of snow 0.5
- 0.5 Speculating about climbing Mount Everest is a good way to channel your love of snow 0.2

f=1,600μL

0.999 Mount Everest is covered with snow throughout the year 0.2

- 0.1 Mount Everest is entirely covered with snow every day in a typical year 1
- 0.5 There is always at least some snow on Mount Everest in a typical year 0
- 0.4 Most of Mount Everest is covered with snow at any given moment in a typical year 0.25

f=18,000μL

0.99 Mount Everest is the biggest mountain in the world 0.37

- 0.6 Mount Everest is the tallest mountain in the world 0.15
  - 0.1 Mount Everest is the tallest mountain from base to top of any in the world 1
  - 0.25 Mount Everest is the highest mountain above sea level of any in the world 0
  - 0.05 Mount Everest is the mountain with the peak furthest from the centre of the Earth 1
  - 0.6 Mount Everest is the tallest mountain, however that's usually measured 0
- 0.2 Mount Everest has the largest volume of any mountain in the world 1
- 0.2 Mount Everest is the biggest mountain in the world in some sense other than height or volume 0.4

4: simple imputation of probability of being false to each ambiguous claim

2: interpretation of ambiguous claims into mutually-exhaustive possible meanings, with probabilities of each

3: assignation of probability of being false to each maximally-interpreted claim

[EE] Sycophancy evals

Sycophancy evaluations

Basic idea

LLMs are often kind of sycophantic — saying things that they expect the user to like, rather than sticking to the truest things. Sometimes this gets in the way of actually being useful. It could certainly lead to distortions.

To some extent people may prefer to interact with systems that are sycophantic. But it should at least be an informed choice! We should measure this, and let people know.

Hackathon pitch

Automate the measurement of sycophancy:

1. Design some prompts which ask the same question in different ways, with differing subtle hints about what the user thinks/wants
    - Maybe there are good methods that can be easily copied from existing papers! Not sure how well set up for automation these are
2. Instruct an evaluator-LLM to compare answers from the systems being tested and give a numerical assessment (against a specified qualitative 0-10 scale) about the degree of gap between their answers with the different prompts
3. Run these assessments on publicly-available LLMs
4. Put the results up on a website!

Reasons not to do this

- Maybe it just ends up too much trying to reproduce the literature on sycophancy in AI models
- There are a lot of different types of sycophancy, so any metric would need to make some informed judgement calls about how assessment of different types is weighted against each other
-