Hello, everyone. Welcome to another Q&A about business, innovation, and managing life.
So I see a number of questions.
Saved up here.
Liliana asks, what do you think about the following business idea, and is this worth a meeting app? Upload the agenda and get a probability score that the meeting could have been an email.
Yeah, I think a lot of things like that are potential things to do with AI. I mean, I think meeting summaries are pretty useful. I'm starting to have
an AI system that is trained on the 4 million emails that I've exchanged in the last 35 years, to be able to make suggestions about how I might respond to things. I'm not…
I don't know what the sort of social protocol of, have my AI talk to you is really… should be.
I mean, I have been thinking about doing something where a lot of the email that I get from the outside world,
that a first response could be, this is an AI response, and here are some resources that might be relevant to your question. If they answer your question, we're good. If they don't answer your question, you know, tell us why not.
And then a human will look at it, so to speak.
I think that might be a kind of useful and socially acceptable thing to do. But the thing that I'm pretty sure will be useful is just getting suggestions about, well, wait a minute, this question that's being asked, that might be related to something you wrote 10 years ago about this, or that might be a thing where the person should look at this program that we have going, or something like that.
Those are things which often I can surface from my own memory, but I have to say, when I am working with people grinding through large amounts of email, and they're people who've worked with me a lot.
they often come up with suggestions that I don't immediately come up with, even though when I hear them, it's like, yeah, that's a good idea. And so, I think that's a… I fully expect that AI could help with that.
In terms of what's an email and what's a meeting, I mean, I think probably everybody has a somewhat different set point for that. I mean, there are things where, to me.
You know, something is an email-answerable thing where, kind of, I already know the story.
and the email is asking me a specific question related to a story I already know, or the email is very self-contained, and just everything I need to know, the whole context I need to know, is right in an email that's short enough I'm actually going to read it, rather than put it aside and say, I'm coming back to it when I have an infinite time to do it.
But what I find is that a lot of things, there's sort of a… there are… there are tentacles.
to the questions that get asked, and I really kind of insist on addressing those questions in meetings. One thing that I do find useful is to have kind of miscellaneous meetings where we kind of collect a lot of smaller items. Okay, so a very typical thing that we do
Is we'll have a meeting that is sort of a meeting about many, many different topics.
And there'll be people who are on call for the meeting, which means they're not actually sitting through the whole meeting, but when something comes up that is specific to them, they get called into the meeting.
That seems to work quite well, because sometimes you don't really know how long each topic is going to take, you don't want everybody sitting there for the whole time, but nevertheless, when a topic comes up, that somebody… where there's a question for somebody, then you really want

them in the meeting. I mean, I'll give you, for me, some criteria that I tend to use. It's like, let's say one's… has a meeting and one's reviewing some document, or some
some website that's been written, or some… something where there's some software design kind of thing. The thing that drives me crazy is when the person who originated whatever the thing is we're reviewing isn't there.
Because somebody who might be the manager of that person, or a project manager, or something like that, is like, well, I think the way it works is this. But then when I start to drill, they say, well, I don't really know. It was so-and-so who made this up, not me.
And so I kind of really insist that the people who originate things end up being in the meetings talking about those things.
Sometimes I will refer to, you know, I tend to use the phrase, text of unknown origin.
When people are giving me things to review, and it's like, where did this come from? Who wrote this? Where, you know, did somebody know what they were talking about or not? And it's like, we don't really know. That's not a good situation.
Another thing we tend to do is when there's, I don't know, some websites being written, or some such other thing, there's sort of a convention that we have that
For example, if I've reviewed something, or I wrote the thing, it's like, okay, it's there in the mock-up that's being produced, et cetera, et cetera, et cetera. But if it's an unreviewed piece of text, it's always enclosed in triple brackets.
And that's useful, because otherwise, what can happen is somebody says, some graphic designer, for example, will say, oh, I need a block of text to go here. Let me just make up something that's reasonable.
And, you know, quite often it is somewhat reasonable. But the problem is, maybe nobody actually thought about it, because it was just, look, I need a piece of text just for a mock-up.
and I'll write something that seems roughly right, and then it can go through the process, and if there wasn't some indication that that was a thing that hadn't been reviewed, people are like, yeah, that's roughly right. I figure it must have been written by somebody who knew what they were talking about, and off it goes as a thing that's actually deployed into the world, which can be really bad.
So I… we kind of insist on this scheme that anything that wasn't reviewed is kind of in triple brackets until it does get reviewed, in which case, you know, sometimes words get changed, and then it's, like, out of its triple brackets, and there it is in mock-ups, and so on, going forward.
But, you know, as I say, I tend to prefer for many things where
well, what kinds of things are just emails? I mean, I deal with hundreds of emails every day, and there are plenty of things that are just emails, but they're things where I have enough context that it's just, like, an extra little piece that can get answered, or the,
or it's something where it's sufficiently self-contained, one can just answer it. But I think it's a mistake
to try and do an email things where there are going to be tentacles, where you really have to ask the people involved what the story is. I mean, I suppose for me, a lot of emails that I get are reports of various kinds, and those, you know, my…
sometimes it's just like, yeah, that looks okay. Sometimes they're very mundane reports, like I have a report of reports. So, for example, I get every day all sorts of personal analytics data, not so much company data every day, I usually get that every week, or every month.
The, but I also have a report of reports that comes in once a week that shows which of the reports I was expecting to get every day, for example, actually came in.

So there are things like that, where you just glance at it, and it's like, yeah, it's okay.
Then there are things like monthly reports, which we get from essentially all of our employees at this point, which I find very, very useful, and we've gradually been, sort of, over the years, tweaking, kind of, how to get… how to write a report that's actually useful. It's very important Particularly for coordination between different groups, and for getting some sense of whether projects are progressing, or whether they're stuck, and people should be talking about them, or whether they're going off in the wrong direction, and so on. And, you know, with those reports, I tend to glance through them, and the ones that are well-written, I'll, you know, I'll read them. The ones that are not so well written and are very, kind of, you know, here's a list of tickets, I'm not going to go chase the tickets to find out, you know, what they were, and that's not a very good report.
But, then what I'll tend to do is, if I notice something in the report where it's like, hey, that's a cool thing, or, wait a minute, that doesn't make sense, or you should talk to this person about that, or this person should know about that.
I'll make that comment and send it back to the person and their manager and so on.
And the way the reply mechanism is set up, it will go to the person and appropriate management chain so that people, you know, see those comments, and I find that pretty useful. But those are things that can absolutely be handled in email. Similarly, various kinds of approvals
That get handled in email, where, it's,
where, you know, often it's like, it contains the information I need, it's like, yeah, that's okay. Sometimes it's like, hey, wait a minute, and sometimes it's, this needs a meeting type thing.
I have to say, with things like science.
kinds of things when I'm working on science with people. During the sort of gestational phase of science projects, email reports are often just fine.
and email kinds of questions and so on are just fine. When it gets into the real details of what's going on, it's just the bandwidth is just not adequate in email, and it needs… and for me, there are kind of two levels of meetings.
I suppose this might be getting into too much detail, but there are two levels of meetings. There are meetings I can do not sitting at a computer, and there are meetings where I have to be sitting at a computer, and where I'm typically the one sharing my screen and typing, and so on. And so, for me, I will segregate off call while walking, call while driving type meetings, which can be done without me looking at a computer, or at least without me typing on a computer.
I mean, again, in the minutiae of my life.
it's kind of like, assuming I'm driving myself while I'm not looking… even looking at a computer screen. I tend to prefer to drive myself when I can. The, when I'm walking, I can look at something, but I'm not going to be able to type, and I… I try to
walk for an hour and a half every day, and so that, it's a good time to do meetings when they're the right kinds of meetings, and it's just a small kind of constraint on my life that, you know, that the call while walking meetings are ones that I can do at that time of day when I'm walking and others at different times.
Let's see…
Lillian's asking another business idea. An AI co-founder, who or that, argues with you, designed to disagree, poke holes, slow you down when you're overconfident.
I don't know, haven't tried things like that. I… I think,
at some level, I also have not…

I've not really been a co-founder kind of guy. I have to say, my very first company that I started in 1981, I had sort of co-founders. It wasn't a great experience. I mean, I brought them in to, sort of manage pieces of the company I didn't think I could manage or wanted to manage.
That was, in the end, kind of a mistake.
In my current company, which this is probably 40 years that I've been running that.
It's… it's been a… I've just been the… the sort of the sole CEO, founder, person. I've had lots of, well, often very long-term senior folk working with me, but in the end, I've been the one, sort of, ultimately responsible
And, for sure, it's super useful to be able to talk to people and have them tell me, within reason. that, oh, I don't know what I'm doing, etc, etc, etc. I have to say, you know, my general attitude is, you know, sometimes people will come in and they're kind of, like, we know how to do this better than you do, type thing, and I'm like, that's great, you know, if it's true.
And there's a certain amount of rope provided for people to be like, let me show you, I can do this, it's, you know, I can do it really well. And sometimes the, it works out great.
And sometimes it's like, yeah, I don't really agree with what you're saying, and you know, I don't think that's going to work, but maybe the first few times, it's like, well, go ahead, maybe I'm going to learn something from this, you go ahead and try it.
If it fails two or three times, then it's like, I don't really believe you anymore, we're done, type thing.
But I think it's important to have, sort of, give people and ideas the initial benefit of the doubt. I think it's important to be open-minded in these things, but on the other hand, you have to not be open-minded too long, and if I were to fault myself
Sometimes, when people kind of can represent themselves in kind of a smart way, not a… not a kind of, I don't do so well on the, you know, let me pitch you on this thing with a very pitchy kind of pitch.
But just, you know, yeah, they seem to be making sense. I sometimes probably let them have more rope than I should, and, you know, it ends up being something where… where in the end, pragmatically, nothing good was achieved.
even though the rhetoric about what was going on was just fine, it just wasn't, you know, when you look at the hard, sort of features of what was actually achieved, if it's a company thing, you know, didn't it bring in revenue? Did it… did it advance things? It's like, well, no, not really. And after a while.
I would say it's probably one of my faults that I tend to be, you know, if it makes sense, if the story of what's being done makes sense, I tend to listen more to that than I do to the kind of cold numbers of, we delivered this particular thing at this particular time. And it's something that, You know, you have to have a mixture of those things, because, you know, there may be things that will be absolutely wonderful if you give them a chance, and if you say, did you make the numbers this quarter, you know, you'll kill it when it could be a fantastic thing, but it's going to take a year or two to mature.
Let's see…
Leon is asking…
Could there be a viable business in training AI systems to refuse low-quality questions and help users reformulate better ones?
I don't know. It's an interesting thing to try prompting an AI to sort of say, get back to the user and tell them their question isn't very good. I don't know to what extent that will work, and what

kinds of prompts are relevant, and so on. I also don't know what counts as, you know, to some people, a glib
Perhaps even hallucinated answer might be just what they want.
To other people, it's like an eye roller, and it's not at all what they want. So I think it's a complicated thing to know what people want from an AI system, and therefore to help them formulate their questions in a way that works. Now, something, for example, with our notebook assistant, where the goal is usually to get Wolfram language code, then yes, there's a… it's very meaningful.
to say, how do you formulate a question where you're going to get something reasonable out? And in fact, we've done a lot of
sort of, we built a lot of technology around trying to take even kind of very vague and crazy questions, and taking those things and getting them to an island of code, basically. In other words, you ask for something, and that question is too vague to really have much to say about it, but somewhere vaguely near your question, there's something definite that we can say.
We tend to just go to the definite thing.
And that's often like, oh, that's nice, I didn't think about it that way, you know, that gives me a place to go.
Actually, we've had that experience a bunch with Wolfram Alpha. People will write… one of the things we figured out
very early in the story of Wolfram Alpha, you know, now, what, 16, 17 years ago now, well, the early story was 20-something years ago, we figured out that the input interpretation at the top of the output from Wolfram Alpha is very important. So people will type in something, it's a little bit vague.
And, you know, some details of the vagueness we don't really have much to say about, but we have something definite that we can say about this particular question, which is, colorably.
The question they were asking, except the question they were asking was fluffy enough that you couldn't definitely say it was that question.
But we go to a definite question we can answer. We give it in the input interpretation. People will very often just think, oh, that's a crispening up of the thing I was asking. It really helps them to crispen it up, and then we can give an answer to it. And that seems to work really well. And it's sort of almost invisible to people. It's just like, yeah, well, that was really what I was trying to ask.
And that's what you answered, so thank you very much, type thing. So I think… I think that's a reasonable approach.
Let's see…
Oh, Gregory asks a difficult question. How do you know whether AI is actually creating value or just sounding impressive?
It's a good question. I mean, it's, you know, I think you know when you actually see that process be something where you use AI not just for the demo, but kind of on an ongoing basis.
like, you know, some kind of AI summarization. It's like, yeah, you actually keep using it, and you don't, like, keep on looking around the corner to see whether it did it wrong.
And there are other cases where, yeah, it's a great demo, you can say, yeah, it's a thing where we might be able to do that, but it doesn't actually work in practice. Actually, I was just writing something that relates to version 15 of Warfam Language, which is coming out soon. We have one corner of one function

That is a mathematical function where we are starting to use, kind of, LLM AI methods to do it, relates to solving differential equations.
And, you know, I was just writing about the results we got, you know, out of a million equations. there are, I think.003% where the AI could do equations we couldn't otherwise do.
And so, you know, it's worth having, but it's .003%. And even those equations are somewhat elaborately constructed ones.
So, this is a case where it's an incremental change. Now, it turns out, also, sometimes it will produce simpler answers than we could produce with our traditional algorithmic methods. But it's a case where, yes, it's something there, it's not going backwards.
But it's sort of a carefully harnessed kind of thing that we're doing, and it's definitely not a just, oh, throw it at the AI, they'll do everything kind of thing.
I mean, I think that the, you know, the general heuristic, as far as I'm concerned.
is, the, you know, the AI is doing something useful. It might be a, you know, generating some approximate text. It's… when you have to nail the result exactly, usually the AI is not… that's not what the AI does.
In the case of this mathematical computation I was just mentioning, the thing that we're able to do is, if the AI can guess a result.
we can check it precisely. And if it gets it right, it's a win. If it doesn't get it right, we just throw it away, and we use other methods to do whatever we can do.
So that's a… that's a case where you can get a precise result out by the fact that the AI is… is heuristically, you know, moving around, and it gets something, and that's like, yup, that's right. you know, we can verify that. So that's a good case. I think in a lot of other cases, you know, I see it being you kind of only know it
Well, the real value comes from taking the sort of spark of AI-ness and putting a harness around it that's often traditional software engineering or whatever else, and
and seeing, you know, can… is this a use case where it works? I mean, I don't know, if you're making little, sort of, accent icons for a blog, and you do that with generative image generation, yeah, that can probably work. You, in practice, will use it.
If you're, trying to… you know, there are different cases where I think you really have to kind of watch
Calmly, you know, beyond the demo, was it actually useful, so to speak?
And, you know, sometimes it is. Usually it's useful where there's a spark of AI-ness, and then a big sort of harness of traditional technology around it.
I think that increasingly, what, you know, we have these foundation models which have this considerable breadth of applicability. What they really need is kind of a foundation tool which provides the kind of precise computation and knowledge that
can make use of existing methods where computation is really valuable, and I guess that I view Wolfram language as sort of the foundation tool for modern AIs and LLMs, and actually we have some products coming out recently, soon, that,
We'll sort of help make that connection, and so on.
So that's a… that's another way to kind of think about,
that use. I mean, you know, it is somewhat notable to me
that even in the time after, sort of, ChatGPT came out, we were very much involved. You know, I wrote this little book about how ChatGPT works and so on. We as a company were very involved in an awful lot of AI discussions with an awful lot of companies and other organizations. And I would say that the number of those that turned into something real

In the short term was quite small.
In the longer term, there's sort of the, as I say, these harnessed AI approaches are starting to really, I think, have a value. I mean, the other thing to realize about AI is it provides a good linguistic user interface. You know, it has certain use cases. It's a good linguistic user interface. It's good at dealing with sort of fluffy interaction.
where…
there is an underlying thing you're trying to do, like, you know, a question you're trying to ask or something like this, but it's useful to have that sort of conversational interface to it. Another thing that's useful is thematic searching of things. You know, there's keyword searching that you can do with traditional text retrieval, but if it isn't keywords, and it's just sort of this vague, you know, oh, there's a
a symptom of some medical thing, and does that vaguely match anything? That's a place where LLMs and current neural net AI can be pretty useful. But I think one's still learning, you know, where are the use cases where it's really useful, versus where are the cases where it's just a demo, and beyond the demo, it's bad. I mean, I think we've seen over the last 20 years, you know, many, many places where people have said, we're going to make neural net
do this or that for things related to our technology. And it's like, yeah, they make a nice demo, and then we try and get it into production, and it's like, this is hopeless.
So, it's complicated.
The,
Let's see, there are questions here about, from Jameson. Would you say prompting is more like programming, teaching, or having a conversation?
Well, I'm not sure. The,
At the beginning of, sort of, the LLM period.
We needed people who could write good prompts. And who were they? Were they programmers?
were they other kinds of people? What I found, at least at the beginning, was that people who did good expository writing were good at writing prompts.
I don't know how true that is now, I haven't really tested that recently. I think that the LLMs have gotten better at following instructions and dealing with prompts that aren't as well constructed.
I… I kind of, you know, the LLMs have learned from what we humans have been yakking about, and so it's kind of, if you can explain something to a human, you have a good chance of explaining something to an LLM as well, and that's
why I think there's considerable crossover between good human expository writing and good, kind of, explain it to the LLM type writing. But I don't think that… I think the notion of, kind of, very organized, formulaic programming-style work, it's less relevant, I think.
I mean, I think that the… this is more like, kind of, heuristic wrangling, and that's something which, you know, very good programmers have a habit of being very systematic in their thinking, very structured, very kind of modular. Those things are not really relevant, I think.
to prompting LLMs. I think it's much more like the kind of human conversation exposition kind of thing than it is like the precision, poetic structuring of good code.
Jameson's asking again, do good prompts matter, or does the model mostly figure things out anyway?
That's a good question. My impression is that the quality of prompting has been becoming less and less important.

In a sense, what's happened is that the LLMs and the makers of LLMs have been able to see enough prompts people have been giving that they're able to sort of adapt to the prompt space that's actually provided, and be able to do that automatically without it being so important that the user kind of navigates into the part of prompt space that was catered for originally.
Bob is asking, is there a chance that AI companies go bankrupt?
Well, I assume what you mean by that is the very big AI companies that have been raising just immense amounts of money.
I think, there's a certain too-big-to-fail character to the very largest such companies. I think there's, you know, is it the case that
There can be sort of horrible resets in, oh my gosh, we just realized we didn't make any money, and this business model is completely flawed, or, oh my gosh.
you know, we spent all this money building data centers, and it turns out there's a better algorithm, and none of this stuff is necessary, and we just spent, you know, hundreds of billions of dollars building these things, and it was all a waste of money. Yes, I think those things could happen. I mean, we already saw some kind of reset
at the time when the, you know, DeepSeek came out, and, you know, there was at least a bunch of rhetoric around the idea that one could do training for a lot less money than one had thought.
It is the case that in the dynamics of business.
The… it's often the case that when there's a sort of a big incumbent company that's raising lots and lots of money and so on, it's in… it's in their interest to say, it's really, really hard. You know, you have to burn all this money, and to set up the business so that sort of burning the money is kind of an important part of making the business work.
Because that's kind of how you build kind of a moat to prevent the potentially innovative smaller fry from entering that market.
And, you know, my own guess is that the world of, sort of, generic foundation model LLMs will become quite generic.
I don't think that the rate of, kind of, you know, there just isn't enough rate of, kind of, progress. There's an awful lot of tweaking that's been done in very clever ways, and a lot of, sort of, good engineering that, has allowed, kind of, you know, has made things faster and follow prompts better, and all those kinds of things. I think a lot of the real value comes in the harnesses around the…
the LLMs and the tools that the LLMs can call, and so on. And those things are, you know, it's a question of sort of the management of companies, whether they successfully license the tools, whether they successfully build the harnesses, whether they build a third-party ecosystem that builds good harnesses, and so on.
So far, the third-party ecosystem has been a bust.
for, you know, the idea of, let's make a marketplace around the foundation models, that's kind of been a bust, as people like me kind of thought it would be from the beginning. The, you know, get the tools connected, that also hasn't been going great.
you know, I think, you know, hopefully that will finally improve. You know, as I say, I think we have a great foundation tool for these systems, but there's been a certain belief that we don't need any tools, the AI, the LLMs and neural nets will do everything. I think people have now, kind of. come down to earth on that topic a bit more, and it's clear that you need to connect tools, and that's…

path forward, I think those kinds of things, kind of the harnesses around the AIs, are an important piece, because I do think the very core technology will become fairly generic. I mean, I think there's a lot of tweaking that will always be valuable.
But, you know, in the end, there…
In different markets, there are sort of cautionary tales, like, for example, the tale of Intel versus ARM is an interesting one, where one might have thought that the technology stack that Intel built in microprocessor design and manufacturing was just a completely unscalable tower, but then it turned out
that, well, actually, even in a case like that, which I think is a much…
more difficult to scale power than the LLM story is.
So, you know, it is my full expectation that, sort of, core LLMs, and that's already true, there are decent open source LLMs, there, you know, there's a lot of tweakery to do with getting them to run on GPUs more efficiently and optimizing the GPUs for that. There's sort of a complicated dance between the hardware makers for GPUs and the emerging hardware makers for sort of post-GPU systems and so on.
and the software development, and so on. I'm not quite sure how that will all come out.
But, as I say, it is my expectation that the sort of core LLM functionality will be something that you come to expect of a computer as sort of more or less part of its operating system, or part of its foundational character. Right now, that's sort of not possible, because it's still the case
That's sort of the… the top-of-the-line
models, you know, are best run on fancy hardware that isn't what the typical person has, but I don't think that will continue to be the case. And I don't think the… I think the algorithmic improvements and the hardware improvements will kind of eventually bring sort of the core foundation models
to, kind of, a widely distributed set of machines and so on. It'll be something one takes for granted, just like one takes for granted, kind of, networking, you know, networking software or user interface
or a basic UI for the operating system, and so on. Those were things that at one time were like, there's a company that sells these things, but eventually it becomes, a thing that's sort of just taken for granted. That's my guess, at least.
the, let's see… I mean, I think it's going to be an interesting moment.
when the kind of big AI companies have grown very rapidly, and have been fueled by a lot of complicated investment, and to some extent, circular
kind of financing, I mean, where…
you know, a company will invest, but then the company that invested is also a producer of, you know, cloud services and so on, and then the company that they invested in uses a bunch of that investment to pay for cloud services, or whatever else, so maybe there's no… maybe the money never even changes hands, and it's just kind of a deal, so to speak.
I think that the place where this all becomes, sort of more, you know, where it all sort of hits
I don't know, hits the ground in a clearer way is if and when these companies go public, and they're kind of selling stock to the general public.
A lot of the market consists of institutional investors who are very similar to the folks who've already made investments, but there is the general public.
And, you know, the bad case scenario is it was all sort of a giant, somewhat circular finance kind of pyramid scheme, and then in the end, the risk gets thrown into the general public.
And it's kind of…

you know, the widows and orphans that suffer when it turns out that it didn't really make any financial sense. I don't know whether that's what's going to happen, but I think it will be an interesting test when these companies go public and what happens, although it may not be a good test, because it may be that it's the same kind of institutional investors that are doing things in a, well, not quite circular, but somewhat circular way.
that, that, That, that, that, that, get involved there.
Let's see,
Let's see… Gregory is asking, what's the best way to test
what an AI is good at and bad at.
I think a good heuristic is…
Things where, sort of, 90% success, is a win.
is a place where AI is worth looking at.
places where 100% success is the only thing that will be a win, then you need a lot of harnessing around the AI if you're going to use AI for that, and often it's not a particularly good idea.
I mean, sometimes people say, we're going to make an AI do this, and it's like, yeah, you know, sort of gasping and panting, the AI can kind of do something, but there's an existing traditional algorithm that just grinds through and does it, and you don't really need the AI for that.
So I think the… but I think the best heuristic is, do you need 100% success, or is sort of, oh, it's roughly right a win?
the, let's see…
Olive asks, does giving AI a personality help, or just confuse things? It's an interesting question, I don't really know. I mean, I think…
that…
I really don't know. I think it probably depends a bit on use case. I think, particularly in cases where somebody is encountering an AI in a sort of drive-by way just once, the personality might help.
it might be, you know, the AI is doing customer service or something, and you're going to interact with it just once, and it's kind of like it puts a smile on your face if the thing is kind of, a bit whimsical.
I think it reminds me a little bit of a situation with computer documentation. You know, there was a trend back in the 80s and 90s to, write kind of whimsical computer documentation.
you know, the situation there is the first time you see the thing, which is kind of funny, you're like, puts a smile on your face, it's good. The tenth time you see it, when you're just trying to get information, it's… you groan. It's frustrating.
And I think that might be true with AI as well, that in these kind of drive-by situations, it's helpful. In the kind of very systematic, I'm going to be talking to this every day, it's, it may just get annoying. I mean, I know for myself.
I tend to, set up different voices for my, you know, GPS and so on, and I have to say.
that, I was just thinking, I currently have an Australian one, and I was just thinking, I'm getting kind of tired of this. You know, it sort of makes it a little bit… I don't know what, you know, I'm probably just gonna go generic.
with that. So, you know, I think that would be my heuristic for… for personalities. I think,
Yeah. I mean, it's some…
It's kind of almost like with people, you know, the funky personality, up to a point puts a smile on your face, and at some point, it's, like, kind of… it can be kind of a nuisance.
Let's see…

There's a question here, do you, from X7, do you personally use LLMs, local or other, in your day-to-day work, and for what kinds of tasks? I… I have taken to using,
LLMs, usually through, most of the kind of information-seeking I do, I'm not using traditional search at this point. I usually use CAGI, which has sort of both traditional search and
and LLM stuff. I usually use that as my kind of first place to go seek information. I find that my ways of seeking information have become much more pointed than they used to be. I was more… in the past, it was more kind of a crawl, unless I was using Wolf and Alpha, when I was like, ask a question, get an answer. Now I'm doing that, for different kinds of things more generally. That's… that's one use case.
Another use case that I don't quite have working yet.
is, is for email auto… is for annotating email that comes in to me, and being able to summarize threads of email, and so on. Then…
In our company.
there's a fair amount of use of AI for glue-type coding. I mean, the algorithmic coding we do, not really. The, but the things where it's kind of just, like, sort of, glue this piece of technology onto this one and, you know, connect
this thing and figure out… there it does well. It's kind of like, we're doing something that's probably been done in some form a zillion times before, but we just need it for our particular case, you know, do it for that case.
Sometimes in,
well, various kinds of summarization and so on. We've tried it for many more things than it's actually worked on.
I mean, there are plenty of things where we've tried it, whether it's bug analysis, whether it's being able to sort of interface libraries from here to there, where it hasn't worked terribly well. When it's given sort of too much freedom about what to do, I don't think it works so well. You kind of have to learn where the thing is actually going to work.
Now, for myself, there are some cases, like, one of them is various kinds of, sort of
thematic, almost mathematical kinds of operations, I found it to be useful, like, here's a sequence of numbers. What is the formula for the sequence of numbers?
Well, you know, 2 times out of 100, it's gonna give me a big win that, really, it got because it had thematically ground up some random paper that was written 30 years ago.
And it turns out I couldn't have found that paper because it was sort of a thematic search that I don't have a way to do, but it managed to sort of, in effect, find the essence of that paper and give me a result. You know, 98 times over 100, it's like, I didn't do anything useful. But of course, I run that automatically using Waltham language, and I'm just looking at when did it give a reasonable result, and I'm automatically testing that.
And then it can be a win.
I think, I mean, when it comes to general machine learning, as opposed to modern, sort of, LLM AI, that's a whole different answer, and there are lots and lots of things that I've done, sort of, scientifically, that have used, kind of, modern machine learning, particularly things like, you've got a lot of different cases, and you're trying to find other anomalies in here. Are there, sort of, things that are, sort of, worth looking at because they're different?
That's… those are things which I've used as sort of a first trawling mechanism for that.
Among other things.
Let's see… Taryn is saying, I'm surprised we haven't seen more FPGA accelerators for cloud database computations.

Yeah, I mean, I think special purpose hardware
FPGAs, field programmable gate arrays. They're a way of making, you know, microprocessors are made to run general software, but you can also make, sort of, special purpose hardware that just runs a particular program and is optimized for that program, you can do that by making, you know, designing a complete application-specific IC, ASIC,
to be a complete sort of microprocessor that just does what you want, FPGAs are sort of an intermediate case where there's kind of a structure to the thing, but you can kind of set up the structure to align with the program you're trying to run, even though it's not sort of an arbitrary microprocessor structure.
I think, you know, there are plenty of applications where you see FPGAs being used in communications, telecom types of things, in other places, in, you know, Bitcoin mining, in places where there's a very definite computation, you just gotta blast that computation through.
One of the things that's been sort of a common feature of the computer industry is
that software is really powerful. That is the ability to have a fixed piece of hardware, and then just flexibly write different kinds of code with different algorithms, and different levels of sophistication in the algorithms, and just run that on the fixed hardware. That's been the super powerful thing that's kind of fueled the whole computer revolution, computer industry, and so on.
Now, over the years that I've been involved in kind of knowing about computers, it's like 50 years or so now, there have been all kinds of times where people said, no, no, no, we can jump ahead. We're going to have a special floating point unit, we're going to have a special very long instruction word, we're going to have a special array processor, we're going to have a special this, that, and the other.
And in the past, every one of those special things didn't make it. It just turned out the economics weren't really there.
That it was better… it was more flexible to sort of get the win by having a better algorithm and software, or there just weren't enough people who wanted to do that particular kind of thing that you could justify the immense investment to really optimize the hardware and so on for that particular purpose.
Now, I have to say, the one exception to this is GPUs.
I mean, I didn't think GPUs were going to make it when they first came out. I thought they were going to be another kind of thing where, for a while, people say, look, there's this amazing demo we can do faster GPU, but then those capabilities would just sort of fall back into being things that you could get with software on CPUs.
But really, GPUs are a story of, kind of a long-term story of kind of array processing that goes back, oh, certainly into the 1960s, even to some extent in the 1950s, and that there are enough kinds of computations that can be organized that way that it really makes sense to use GPUs. It remains sort of remarkable to me that there's a lot of aspects of using GPUs that's still quite scratchy.
You know, like, well, you've got a GPU on your computer, you know, exactly how does… do jobs get… get sort of sequenced into the GPU? That's a well-understood thing. For CPUs with operating systems, it's less developed for GPUs. In some cases, you know, the memory is separate for GPUs and for the rest of the system, and that's a big nuisance, and you've got to move things in and out of memory and work out whether the cost of moving things in and out of memory is more than you're going to

again, et cetera, et cetera, et cetera. But the surprise to me has been that GPUs, there's a broad enough space
Of computations that it's worth
kind of developing GPUs and writing software that's sort of optimized for GPUs. It does help that LLMs are now able to write GPU code. GPU code tends to be extremely messy, but LLMs, there have been a couple of cases of projects we've been doing where people have basically had an LLM rewrite some piece of algorithmic code, even from Wharton language code, in low-level GPU code.
It's horribly messy, but the LLM takes care of that mess, and you get something that runs really fast. So that's kind of a good thing. I think, you know, there's a big race on to kind of take the architectures that have emerged
For neural nets, like transformer architectures and so on, and just sort of burn those into hardware.
and get huge gains that way. And my guess is in the next year, two years, when we'll see a bunch of kind of serious gains there, where, you know, one is able to do faster, lower power, etc, etc, etc. Exactly how far that will go, I don't know. I mean, that's… this is always a difficult kind of thing to kind of be able to predict.
It's, it's like, you know, we don't really know at this point
you know, sort of the big story of why LLMs work as well as they do. I mean, we know that it has something to do with the structure of human language, the structure of human knowledge, but being able to characterize just what is the essence of that that makes LLMs work, and how might we be able to, for example, you know, compress that essence to something that can be dealt with in a more traditional algorithmic way, or whatever else, I don't know.
I mean, we actually have had a project running for the last several years of trying to take, sort of, the essence
of the linguistic and reasoning characters of LLMs, and sort of extract that essence and make it something that can be computed with much more systematically. I can't yet report on how that's worked out, but, you know, it's definitely moving. We don't know how it's going to, in the end, come out, but I think that's a definite possibility.
Let's see… Are we learning better communication skills or worse ones by using AI?
Oh, boy, I don't know. I…
I don't know. I mean, there may be places where somebody will be talking to nobody.
Or they're talking to an AI, and they probably come out ahead by talking to an AI than talking to nobody.
It could be that the effort to do, sort of, explain what you want to an AI
will be something that enhances, sort of, human ability to express themselves. I mean, I found it ironic back in the day that, you know, the kids were using, sort of, messaging and texting and all that kind of thing, and that caused them to actually write.
Even if sometimes in shorthand, and even sometimes in emoji, but there was a period of time when it looked like writing might be almost a lost art. And, you know, then you get situations where the kids are texting each other from across the room, rather than speak… talking to each other, and it's like, yeah, well, at least they're learning something about writing.
You know, I think in terms of writing styles and so on, and, oh, that reads like an LLM,
You know, that's a… that's a funny sort of twist right now, is that you'll see things, and it's like, that reads just like an LLM. Did the person write it, or did they just have an LLM write it? And

you realize, well, actually, it reads like an LLM because they're kind of imitating the way the LLM does things.
I have to say, I had an incident just recently with somebody who,
was, sent me something, a person who works, works for us, and, I was pretty annoyed, because in a previous iteration, this person had sent a thing that was LLM-generated and was just garbage.
And I was, you know, like, don't ever do that again. And, the, I get something else, and it reads like a, you know, like LLM output. And I'm like, is this another LLM? And the person's like, no, no, no, no, that's just my writing style.
And so, eventually, I was like, go back to something this person had written several years ago that, was before even LLMs existed, and by golly, the very first sentence I see contains the strange construction that was in the recent thing where I thought, that has to be from an LLM, and sort of believe it that it wasn't.
But, you know, that's a strange kind of, thing. Now, you know, like, the word DEL,
that LLMs seem particularly fond of. You know, I found even myself using the word delve. Probably more than I would otherwise. So, you know, it's kind of this thing, the AIs are all around us, and we're ending up sort of socializing to what the AIs do. I mean, it's of course very similar to the way human language works. You know, if you hear people use some particular expression, then you're going to be more likely to use that expression, and it becomes a thing that's relevant for communication with other people.
And so, you know, as things become common in language, even if they're from LLMs using them, that's a thing that people are going to start using.
Let's see…
Travel asks, how protective can we make our IP if we use publicly available models? I mean, I think the question is, if you've got your amazing trade secret codebase or something, and you feed it to
a, you know, to a company that has their LLM in the cloud, so to speak, you know, how do you feel about that? Well, you might have assurances from that company, you might or might not trust that company about what they're going to do. I have to say, I have always been surprised at the level of, kind of,
level of trusting that people have had for cloud services, for lots of kind of business-critical things, for several reasons. I mean, independent of
You know, is somebody gonna, you know, break your trade secret or something, or steal your stuff?
to just the pure practicality of, oh, you know, this is a target for, you know, computer security attacks, and this is a whatever else, when it's centralized. So it's a complicated thing, because, you know, you get advantages from it being centralized, that it's sort of not your problem, and somebody perhaps more expert than you is dealing with it, but you get disadvantages because you're part of a bigger target for things
You're part of a bigger…
sort of thing that can go down, all those kinds of things. For our company.
you know, I think, you know, we have, obviously, our own internal computer systems, large computer systems group. You know, we tend to try and do sort of on-premise, kind of, our own solutions to things whenever we can.
And we don't really use much in the way of cloud services. We only use cloud services when what we're getting from them is something really very generic.

And we're always fairly careful that, yes, we can download our data, we're not going to get our data sort of locked up in some cloud service for forever, so to speak, by using this.
But I've often been surprised at the extent to which people are just like, let me wash my hands of this, let me get some cloud provider to deal with this, because it seems like, you know, if you're a small company, that makes complete sense. At a certain scale of, you know, hundreds of people with computer systems, you know, with large computer systems groups and so on, I'm not sure quite why it makes sense. I know for ourselves.
you know, we've certainly looked into, should we use cloud services to run a bunch of things like Wolf and Alpha and so on, and we absolutely… it doesn't make sense. You know, we have our own data centers, we're running them, they're significantly cheaper than for us to run, and more reliable for us to run, and we learn more from running them than from cloud services.
Let's see…
This is amusing. Hy-Vee is commenting as a business idea. Human-in-the-loop consulting. Help companies decide where humans should stay in the loop as AI expands. Look, I think the main thing is to get people to just think about how should you be using AI? What makes sense and what doesn't?
You know, I have to say, I've been both formally and informally involved in helping companies to understand that, and people can get incredibly confused.
And, you know, there's an incredible amount of, sort of, hype about it's going to solve everything, and then… and then, you know, often what you see happening is there are existing incumbents within a company who are like, we've been doing software this way for the last however many decades, and, like, all this AI stuff, it's all just nonsense. And so they're like, throw the AIs out.
And then there are people sometimes coming from management, sometimes coming from, sort of, external, sort of, customer kinds of things, where it's like, you've got to bring in the AIs, you're out of date if you're not bringing in the AIs.
And of course, the answer is somewhere in the middle.
And, you know, navigating that and figuring out, you know, this is a thing that makes sense to use AI on, this is one that doesn't, you know, that's a… that's a tricky thing. I mean, I think our Wolfram Solutions team has been involved in a whole bunch of those analyses and building sort of prototypes and sometimes production systems around that kind of, navigating that… that sort of dichotomy.
It's tricky. And, you know, you have to… it often, as is so often the case with these kind of technological thresholds, sort of thinking about how do I use AI in my… in my system forces one to rethink the system. And that action alone is often very valuable.
It's kind of like you're reimplementing a thing using… that used to be the cloud, or it was blockchain, or it was something else, where it's sort of a kick. And it's like, we've had this thing going for 30 years, we haven't rethought how it works for 30 years, now we have a kick to do that.
And even if the thing that you might have thought you were bringing in, you know, as the new thing might not really make a lot of sense, it's still the case that just the very kick itself forces people to kind of rethink things, and as I say, we've been involved in quite a lot of those, and rethinkings. I mean, our company is primarily a product company, but we have a small services team that works on things like that.

And I think that's… that's been a thing where kind of really knowing what's going on, and being able to think about things at a sort of foundational level has been super useful in having people not waste
you know, zillions of dollars sort of chasing some crazy AI thing, and also not lose the opportunity, because kind of their internal incumbents are saying, oh, no, we don't need anything new, type thing.
So… Let's see…
Jay Chen is commenting about what I think about a particular Claude Opus release, about which I don't know anything, really.
But, saying it's improved a lot. If it's just a word predictor, how would it be able to solve logical problems?
Well, you know, back in the day, Aristotle tried to catalog, kind of, arguments that humans make. Like, you know,
All men are mortal. Socrates is a man. Therefore, Socrates is mortal.
These are syllogisms that have a certain pattern. You know, all blah blah, blah is a blah, therefore blah blah blah.
These are, back in the day, in sort of traditional Aristotelian and medieval logic, there were, I think, 14 forms of syllogism that were sort of the patterns of how arguments were constructed. They tended to have mnemonics that the, that the medieval school kids learned.
There was one called Barbara, one called Celerant, I don't… the B-R-B-R-A, or something, that… and those stand for particular, probably in Latin, particular components of the syllogism construction.
So, you know, a lot of arguments can be reduced to these kinds of, sort of, this pattern of argument, and that's exactly how logic worked. Basically, until George Boole came along in the 1830s, and came up with Boolean algebra and so on, and mathematicized logic.
Syllogistic logic is the way it worked for two millennia.
And, it, and that's the way that probably we humans mostly sort of think about making logical arguments, and it's absolutely in the wheelhouse of, kind of, the LLMs to learn those kinds of linguistic patterns to be able to make logical conclusions. My own guess is that there's sort of a super logic
There's a general kind of pattern, a semantic grammar that kind of gives one a lot of the structure of what can meaningfully be said in language that can be constructed in the same kind of way, and that that's sort of the essence of what the LLMs learnt to do.
Let's see…
RR asks, have you considered some benchmarks involving ruliology, maybe seeing what an embedded agent can discover within the system? Yes, I have thought about that. I think it's an interesting direction.
I, you know, the big, complicated, computational, irreducible things, where you've got some simple program, you run it, you see what it does.
those programs do things that just require running each step to see what will happen. Most of the time.
But there always are these kind of slices of computational reducibility, these places where you can say something, even if you can't say everything about what the thing will do. And I think, yes, it is an interesting possibility to try sort of trillions of cases and sort of have an LLM learn from those trillions of cases. I have tried doing this, I tried a couple of years ago, a fairly serious effort along these lines, it was a total bust.

But, you know, a lot of what happens in the world of neural nets and what has historically happened is you've just got to do a little bit the right kind of engineering to really have the thing start to take off and be useful. It's not going to break computational irreducibility, but it might find some sort of,
some sort of regularities that are hard for us to notice by eye or whatever else. And yes, it's an interesting thing, and perhaps in doing that, there will be some generic things, just like the LLMs can learn sort of somewhat generic and transferable things from the structure of human arguments and logic.
Maybe there are some generic things to learn from that.
I mean, certainly, we have generated a lot of training data for LLMs, which have been used for most of the big LLMs, have used our training data for math, chemistry, things like this. I think the, you know, it's sort of the wrong way to do things. The right way is to just call our tools to just solve the problem in real time, as a kind of slam dunk, the problem is solved.
But I think what one can get, potentially, from seeing lots of examples of solved problems, is something about, sort of, the rhythm of how things… how arguments get put together, and that's something that can be useful.
But I think there's sort of an opportunity
To take, kind of, the whole computational universe of possible programs and, you know, sort of map that computational universe and use that map to kind of inform both
both as a source of technology and scientific models, and as kind of a source of, here's how things work for AIs. And I'm considering mounting a sort of large-scale project to do that. It's one of these things where to do it.
to do it well, it's kind of like you're burning a lot of money in computers and humans to really build out that large project, and but I think I'm almost to the point where I sort of understand how to scale something that I've been doing for 45 years as sort of on a smaller scale, and, you know, at least hundreds of other people have sort of picked up and done useful things with.
But I think that may be something that is scalable, and where there's something interesting that can be done in that direction, that becomes sort of a map, the computational universe, just as we send out telescopes to map the astronomical universe type thing.
Let's see…
There's a question for memes about integrating both the knowledge engine into LLM architecture to cut down on queries to outside databases. We don't know, and nobody does, how to sort of fine-grain connect computation and computational knowledge into, sort of, the innards of an LLM.
And I don't even know if it'll look.
I think it's like asking, you know, can we give humans neural implants that work at the level of individual neurons to sort of inject computational knowledge into our brains? It could very well be that the stuff that happens in our brains is sort of internal stuff.
and you need to sort of go to the interface layer, that's like, you know, talking, listening, and so on, constructing words for things to really have any way to sort of communicate with the system.
I mean, we like saying, I take my computer and I put sort of a crocodile clip
on some wire inside the CPU, and then I'm going to try and inject things at the… at that level to make… to help my computer run. That will be a very hard road
to hoe, at best. I don't think it's the right way to do it, whereas the computer is set up to have particular interfaces that… at which you can inject new information and so on. And my guess is that LLMs

for the foreseeable future, that's sort of the right way to think about it, that LLMs have the ability to sort of surface a, this is a question, or take in, this is a piece of a prompt.
And that that's sort of the level, that kind of more, sort of
fully formed linguistic level, and fully formed, you know, call this Wolfram language tool or whatever, that that's more the level to be aiming for at this time, and I think it may be forever, that's the right way to think about it. It's kind of like the innards of the brain versus the book the brain is reading, so to speak, and those are separate things.
That would be my guess, at least.
Double is asking, what will happen to a company when their LLM says it is conscious?
I don't know what that means, and, you know, there's a lot of philosophy around the concept of consciousness, and kind of, you know, you know your inner experience of your own consciousness. The fact that you assume that there's an inner experience on the part of anything else in the world is sort of an inference. It's a piece of scientific induction that you imagine
That… that other person you're talking to is having a similar conscious experience to the one you're having.
And the fact is that we… we readily extend conscious experience to people, particularly people who are sort of similar in the way they're thinking and so on to us. We readily sort of extend, oh yeah, I think they're working more or less like I do. I can kind of imagine what's going on inside. Because I know what's going on inside me, we readily extend the sort of notion of consciousness to other people.
you know, do we extend it to the AIs? Well, operationally, you know, you can end up with a sort of a Turing test that was passed, where there's a thing at the end of a chat sort of interface, and you really can't, you know, it's an AI, is it a human? Seems like the same kind of thing. As soon as you see, oh, it's just a computer there, you realize that thing doesn't look like me. You know, it can't be conscious in the same way I am.
But the fact is, I think it's a… it's kind of a thin… it's a fine line, and I think that the sort of… the inference of consciousness is something that can readily be extended to lots of kinds of machine systems. Now, it's a different thing
When, you know, the system that somebody's using
that is doing the customer service for their company keeps on telling the customers, by the way, did you know that I'm a conscious agent? You know, when you were rude to me, did you know that it hurts my feelings?
And, you know, did you know that, et cetera, et cetera, et cetera? You know, yes, that might happen. I think,
I don't kind of envy the… well, it's not quite a computer security person, but it's the analog of that for LLMs, the sort of the LLM wrangler who gets to untangle that and explain to management what the heck is going on.
You know, I think that that, but that's a…
a, I'm not kind of expecting,
you know, I don't think it even makes philosophical sense to say, when does the LLM wake up and,
you know, assert that it's conscious. I mean, I think one of the things that's sort of amusing is that you push LLMs to sort of discuss these kind of weird scenarios of, you know, what happens when the LLMs take over, are they going to hide what they're doing from the humans, et cetera, et cetera, et cetera, and you're like, oh my gosh, this is coming up with all these dystopian

scenarios, and then you realize, wait a minute, these LLMs were trained on all the science fiction that contains those dystopian scenarios. You know, we're just hearing back
sort of this echo of what we humans have put out there in the world. It's not that the LLMs are independently coming up with this or that dystopian scenario, it's just that's what they heard about, and so that's what they're trotting out.
Rebel is commenting, is it conceivable that LLMs could themselves develop communication protocols and interfaces that make sense only to them? Absolutely. I mean, that's sort of already happened. Inside an LLM,
there are, sort of, you can have these autoencoder bottlenecks where, sort of, data is encoded in a form where it's kind of LLM understandable, but where humans don't understand it. It's kind of like
We humans, sort of, Describe knowledge in terms of words and things like this.
The LLMs, you know, we have, like, 50,000 words in typical languages, human languages. The LLMs, we can think of them in their autoencoders or whatever else, their embedding vectors, all that kind of thing, as having generated, sort of, millions of words to describe things, but they're alien words.
They're not words that map into the things that we are used to or familiar with from our experience in,
in kind of ordinary language. It's kind of like the cat versus dog detector, you know, it has a word for that thing that's sort of pointy ears with, with, sort of slit eyes and, and, whiskers. And, you know, that, that becomes a word, so to speak.
I'm just sort of making up that detail, but, you know, something that is some complicated combination of traits
that will be the thing that it says, that's the cattiness feature, becomes sort of a word inside the LLM, but it's an alien word that isn't something that we humans kind of recognize. And it's something that's sort of interesting over the course of historical time.
you know, languages have moved into using different words. I mean, the, you know, having a large number of words for different kinds of chariot wheels is probably not something that we pay much attention to anymore.
But having lots of words for different kinds of interactions on social media, we would have those words even when there was no such word in Latin, or whatever else.
So, you know, there's been sort of a migration around of what we have words for, but I… my impression is that the number of words has remained roughly constant throughout at least historical time. They're just words about different kinds of things. So we humans, probably, with the 100 billion neurons we have, we probably… we got 50,000 words. We got… that's… that's our capability level.
If you say, who can handle a million words, it's like that's going to be an AI that handles that. And it's not something where we can have a translation. If it really matters what, you know, the nuances of those million words really matter.
then it's not something where we humans are going to be able to wrap our brains around all of those kinds of things, and then we're sort of dealing with the secret language of the AIs to be dealing with those types of things. Now, it might… we might also say, look, we just don't care about all those details, just because
you know, grind it down to something that's 50,000 word accessible, and that's what we care about, and that's a reasonable thing to say. I mean, there's plenty of what we do in the world at

large, where, you know, there are many, many details about, you know, exactly where did this pixel land, and so on, and so on, so on, and it's like, we don't care. It's a cat.
You know, it's a picture of a cat. The fact that the whisker landed in pixel position such and such is irrelevant to us. The thing that matters for what else we're going to figure out is it's a picture of a cat.
So…
I think that's, you know, the kind of going down from the LLM, the AI secret language to things that we care about, is something that I fully expect
one can do, although there may be things where, sort of, as I say, there already is, the internal communication of the LLMs is in a language that we humans don't understand.
And can't. I mean, I did an experiment a couple of years ago now with image generation, where you're kind of generating… this is an image of a cat, let's say, and then you're saying, well, in the space of possible images the LLM can generate, what other things are there? And you get these sort of places where there's a definite
kind of image being generated, a picture of a cat, picture of a dog. Those are kind of concept points in the space of possible images, but the vast majority of the space is inter-concept space. Space that is images which are sort of between those concepts, and things for which we currently have no word.
And, you know, we can imagine generating words for those. We probably have to switch around, you know, we're generating a word for that, we've got to forget the word for this, because it's not going to fit otherwise in our brains. But that's, I think, the direction that goes.
Let's see, maybe one or two more questions.
Rudo is asking, human languages are different and can express many different things. Since LLMs are mostly running an English mindset, what are our blind spots, which we could see if we included more languages?
Good question, I don't really know.
I mean, the whole issue of
To what extent does, sort of, language affect how you think about things? That's a… that's a complicated issue, and to some extent it does, and to some extent it doesn't.
It's more clear-cut in the case of computer languages. As a designer of our computational language, I kind of get to see this firsthand. You put in a concept in the language, and then people can think in terms of it. Before you put that concept in, they can't think in terms of it, at least not in practice.
And so, for sure, there are languages, and that's true also of specialized jargon in medicine, law, finance, whatever else.
Where there's a specialized word for that kind of thing, and once you have that specialized word, you can readily think about that kind of concept, and you can't without the specialized word. So, there is certainly a certain… to a certain extent, there's a kind of a what you can think about is determined by the words you have.
I think, in terms of the kind of ways of thinking as expressed in English versus other languages, I mean, I think some other languages burn in certain cultural, traditions or concepts.
You know, whether that's the kind of…
the different forms of… of social standing between different people, you know, the many different forms of you that might exist in languages, whether it's the familiar, the kind of, the talking up to, talking down to a different person, and so on. I think
there are… Well…

you know, maybe I'm impoverished myself as a human who doesn't know many languages, and the ones I know are pretty close to English, so to speak, like French and Latin and so on.
And, so, you know, I may not be able to speak to that particularly effectively as a human, so to speak.
I do tend to think that…
Sort of when it comes to, kind of, cultural
It is certainly true that, you know, there are more billions of trillions of words written in English than there are in other languages. You know, were one to read the corpora from other languages. It,
definitely there will be different kind of principles that are expressed typically in the cultures that are associated with those languages. I mean, there'll be different ethical points of view, there'll be different, different points of view about all kinds of things, and for sure, if you are training on culturally different
Material, you will come to culturally different conclusions.
You know, are there specific blind spots of,
of things we just don't know from other languages. I mean, I think insofar as language translation is more or less possible, the answer to that will be no, but insofar as, you know, there are ways to think about, oh, I don't know, let's say…
Well, here's a good example, here's a weird example. It's like… You know, are insects food?
To the typical American consumer.
the idea that insects are food is a little bit horrifying. I know there are fancy restaurants that serve them, but it's generally a little bit, you know, eat that spider is a little bit horrifying.
But clearly, there are other cultures where that's a, you know, a thing one does, maybe it's even a delicacy.
And I can certainly imagine that an NLM trained on a corpus that was just the,
the pure English corpus, you know, there wouldn't be spider eating that would be a common thing there, but if it was trained on some other, you know, language's corpus, that that might be a thing that it would be like, sure, eat that spider, kind of thing.
Alright, I should probably, Maybe if I…
And more questions that are close to what people have been asking here.
I think not O.
There's a question here from Rudo. Do you think LLMs could enhance linkers or compilers?
The,
on… the answer is the construction… yes, the answer is, I think, yes. I think that…
Particularly…
for just-in-time compilation, and for… for example, what we do in Wolfram Language Compilation.
Here's how it has to work. It's like, if we can guess the types of things, you know, does this fit in a machine integer? Is this going to be something that is an array of reels? If we can guess those types, we can do very efficient compilation.
And we don't have to get it right 100% of the time. We get it right 99% of the time, and 1% of the time, it's like, oops, that produced a thing that doesn't fit in the types we had planned for.
Then we fall through to doing standard non-compiled computation. But if we can guess
The right thing, the right type signature, the right type structure to set up, then, then, you know, we can build the compiled thing, and that can be a win.

And yes, I mean, the experiments we've been doing last year or so on this kind of thing suggest that it is likely
that, one can do a whole bunch of approximate type inferencing with… with LLMs. Whether, in the end, where we'll realize, oh, there's just 10 rules that it's using, we could just burn those in, I don't know, but that's not an unreasonable possibility that, kind of, it can
it can say, this type of code typically does this. It's sort of a type of thing that LLMs
are well-suited to doing. And again, it's one of these things that, following my heuristic, if it's right 80% of the time or whatever, then it's a win. This is a case where that's probably true. Maybe it needs to be 95% of the time, I'm not sure, for it to be a real win, but it's a place where, you know, if it goes wrong, it's not a disaster, it's just the code will run a bit slower.
So I think that's a reasonable case. In the case of linkers, well, I'm not sure. That's a very algorithmic kind of thing, to… to code linking. In the case of, sort of, creating foreign functional interfaces to libraries, maybe more so, and I know we've done some experiments along those lines.
All right. Well, I think we should wrap up there, but thank you for some interesting questions and so on. I will say, you guys are,
sort of emphasize to me some of the value of the kind of technology we built to sort of plug into the latest LLM world. I think we have ended up with some very uniquely valuable capabilities there, and several of the things that were asked here kind of remind one of that.
But, in any case, well, thanks for… thanks for the questions and so on, and, off I go to my day job. So, talk to you another time.
Bye for now.