SDS PODCAST EPISODE 903: LLM BENCHMARKS ARE LYING TO YOU (AND WHAT TO DO INSTEAD), WITH SINAN OZDEMIR

Jon Krohn: 00:00:00 This is episode number 903 with Sinan Ozdemir, author of the Quick Start Guide to LLMs. Today's episode is brought to you by Trainium2, the latest AI chip from AWS, by Adverity, the conversational analytics platform and by the Dell AI Factory with NVIDIA.

00:00:24 Welcome to the SuperDataScience Podcast, the most listened to podcast in the data science industry. Each week we bring you fun and inspiring people and ideas, exploring the cutting edge of machine learning, AI, and related technologies that are transforming our world for the better. I'm your host, Jon Krohn. Thanks for joining me today. And now let's make the complex simple.

00:00:57 Welcome back to the SuperDataScience Podcast. Today, we've got the legendary AI expert and many time author Sinan Ozdemir back on the show for the sixth time. Sinan is founder and CTO of Loop Genius, a generative AI startup. He's authored several excellent books, including most recently, the bestselling Quick Start Guide to Large Language Models. He hosts the Practically Intelligent podcast. He was previously adjunct faculty at Johns Hopkins University and now teaches several times a month within the O'Reilly platform. He's a serial AI entrepreneur, including founding a Y Combinator-backed GenAI startup way back in 2015 that was later acquired. He holds a master's in pure math from Johns Hopkins.

00:01:39 Today's episode skews slightly toward our more technical listeners, but Sinan excels at explaining complex concepts in a clear way. So today's episode may appeal to any listener of this podcast. In today's episode, Sinan details why the AI benchmarks everyone relies on might be lying to you, how the leading AI labs are gaming the benchmark system, tricks to actually effectively evaluate LLM's capabilities for your use cases, what the future of benchmarking will involve, including how to benchmark agentic and multimodal models, and how a simple

question about watermelon seeds reveals the 40% failure rate of even today's most advanced AI reasoning models. All right, you ready for this excellent episode? Let's go.

00:02:26 Sinan, welcome back to the SuperDataScience Podcast. How you doing today?

Sinan Ozdemir: 00:02:30 Jon, thank you for having me yet again. I'm super excited to be here as always.

Jon Krohn: 00:02:35 Yes, we were just tallying prior to starting recording how many times you've been on the show. You've been on more times than you even knew.

Sinan Ozdemir: 00:02:42 That's true.

Jon Krohn: 00:02:42 You came into the Green Room before we get into the recording studio and you said you've been on, this is going to be your fifth time. But actually it's your sixth.

Sinan Ozdemir: 00:02:54 That's true.

Jon Krohn: 00:02:55 So you were one of our first guests ever. Episode 21-

Sinan Ozdemir: 00:02:58 21.

Jon Krohn: 00:03:00 ... back in January 2017. Exactly old enough to drink that podcast episode was finally in the US. And yeah, then you were back about a year later in 161 and then jumped a couple of years, episode 333 in January 2020. And then significant for me was another year after that in February 2021, you were one of my first guests when I took over hosting the show from Kirill.

Sinan Ozdemir: 00:03:25 That's right.

Jon Krohn: 00:03:25 That was episode 445. And then last year at the Open Data Science Conference East in Boston, we met up quickly, I set up a camera and you and I recorded a quick

episode live in person in Boston on AI alignment, LLM alignment. And yeah, so that was episode 784. So lots of episodes. This is number six, it's going to be the best yet.

Sinan Ozdemir: 00:03:51 Yeah, that's how I think about it too.

Jon Krohn: 00:03:57 Okay. You're currently writing your 10th book, is that right?

Sinan Ozdemir: 00:04:01 That sounds right. It's up there. I'm working on, yeah, I'm working on two right now, actually.

Jon Krohn: 00:04:09 So do you think that the books get better? Or do you think you're, like, because often rock bands they have many years to make their first album. So I don't know, Bush was a band in the '90s that was super popular in Canada. Their first album, 16 Stone, I guess it was popular in the US as well, you know them.

Sinan Ozdemir: 00:04:26 [inaudible] Bush.

Jon Krohn: 00:04:26 It's funny, they're British people don't know them in the UK.

Sinan Ozdemir: 00:04:29 Machine Head. Are you kidding? It's one of my favorite songs,

Jon Krohn: 00:04:31 Machine Head. Exactly. Glycerine. And that one album, it was like everyone knew every track and huge, huge global hits, they were doing world tours. And then so the record company's like, "We're going to need another album next year." And they were like, "Oh, man, we spent five years making the first one." And yeah, that seems to happen all the time with rock bands and then they'd never quite have the same kinds of hits. Do you think you're like a rock band? Or do you think your books get better? Do you think number 10 is better than number one?

Sinan Ozdemir:	00:04:59	A bold question, throwing shade to all the rock bands out there? No, I think I very much go in to my books with a mentality of not necessarily better but something different. I'm very conscious about, I wrote about this four years ago, do I really have something new to say here or do I digress? Just something else? I aim very much for diversity more than I aim for just building upon the same content every year.
Jon Krohn:	00:05:28	Do you do second editions ever?
Sinan Ozdemir:	00:05:30	Oh, yeah. Actually one of the books this year is a third edition of my most recent Quick Start Guide to LLMs. So only really two of my books have ever gotten second and third editions and they're very much the holistic books. My first book ever was the Principles of Data Science, and that was almost 10 years ago at this point. And it was very much a zero to deep learning in 300 pages. So that's gotten updates over the year, because it's an introduction to data science. And the same thing with Quick Start Guide to LLMs. It's very much meant for someone who is just diving in for the first time, engineer or not, just what do I need to know to understand LLMs? So that book gets updated because that's pretty much evergreen content at this point.
Jon Krohn:	00:06:13	Nice. Tell us about the Quick Start Guide to LLMs. Tell us about what's in that book and what's new in the third edition.
Sinan Ozdemir:	00:06:19	Sure. The book is, actually, I just literally had a copy here just because I was holding it up for someone else earlier right there.
Jon Krohn:	00:06:27	Sure, sure, sure.
Sinan Ozdemir:	00:06:27	But yeah, the book is very much organized into a few sections, starting with a level set, what is a language

model? What is an auto-regressive language model versus auto-encoding language model? What does that mean? In fact, in one of my episodes on Super Data Science, the one from 2020, January 2020, that was actually the first time I had brought up BERT and GPT on the show. So the first part of the book really just talks about what is the difference between these kinds of LLMs?

00:06:56 Then it gets into the different applications and how to evaluate them and the kind of philosophy of alignment. What do we mean when we say alignment? And then the last parts are usually, "All right, now that we're speaking the same language, let's make a phone bot. Let's make a embeddings classifier. Let's actually build these things. Let's make a chatbot from scratch." So it's very much organized into those three sections.

00:07:20 And the third edition is pretty much the same. The editions are mostly around obviously newer LLMs, newer evaluation criteria, newer benchmarks, and just different ways of applying LLMs. Like reasoning models, for example. This will be the first time that I talk about reasoning models in one of those books.

Jon Krohn: 00:07:40 Very cool. So yeah, reasoning models like o1, o3. We actually at the time of recording-

Sinan Ozdemir: 00:07:44 R1.

Jon Krohn: 00:07:45 R1, yeah, from DeepSeek. We at the time of recording have a new, have you played with o3 Pro yet? Just came out today.

Sinan Ozdemir: 00:07:53 I've not played with it. I got the email I think yesterday, like late night yesterday. But I have not yet tried it. What's the schtick here? Better, faster, cheaper? All the above?

Jon Krohn:	00:08:07	I don't think it's cheaper. I think it's bigger and better. We have the o3 Mini some months ago and they had things like o3 Mini High where you could have it do inference for longer. And so I believe o3 Pro is kind of orders of magnitude larger in terms of model weights, and so theoretically nuance.
Sinan Ozdemir:	00:08:30	That makes sense.
Jon Krohn:	00:08:31	Yeah. But I haven't used it yet. It's been a busy day so far. And now I'm recording this podcast episode. It's tough, tough to stay up-to-date on all the fast-moving things in AI.
Sinan Ozdemir:	00:08:39	It is. And again, that's also why a lot of the book is about models will change. But at the end of the day, this thing is the next token predictor. And just because it thinks before it speaks to you doesn't mean we can't evaluate it the same way as we evaluate everything else.
Jon Krohn:	00:08:54	For sure, for sure. There are megatrends that you can ease into and find comfort in despite brands maybe changing quickly, model brands, company brands.
Sinan Ozdemir:	00:09:07	Going from 4.5 to 4.1 from o1, to o3, to o4.
Jon Krohn:	00:09:09	Exactly.
Sinan Ozdemir:	00:09:11	Back to o3, but o3 but better. It can be a lot to keep up with.
Jon Krohn:	00:09:16	Exactly. That's right. So you're working on another book right now, I understand as well?
Sinan Ozdemir:	00:09:22	Yes.
Jon Krohn:	00:09:22	I don't know if you want to talk about that on air? Maybe it's secret.

Sinan Ozdemir:	00:09:24	It's not secret. I mean, I'd love to talk about it. The title is not known yet, so I'll give you at least the working title. It's very much an applied AI book. It is very much around, it's very much a cookbook of AI applications. It's assuming you've read my Quick Start Guide, and again, we are speaking the same language here, is let's just dive into the top 20 applications of LLMs I have seen in the last 10 years. Everything from, "Let's build a prompt to summarize a podcast transcript and really evaluate this thing to its core." All the way to, "Let's use reinforcement learning to actually build some of these reasoning models from scratch for your specific domain." So it's a pretty wide spectrum of things that we do, but it very much starts, hits the ground running kind of book.
Jon Krohn:	00:10:12	Very nice. Very nice. I like that cookbook model. And I assume both of these books, Applied AI and Quick Start, they're both in Python?
Sinan Ozdemir:	00:10:18	Yes. So everything, all the code that I'm writing is in Python.
Jon Krohn:	00:10:21	Nice, nice. And in addition to your books, to your many books, you also do a ton of teaching in the O'Reilly platform, which also actually kind of brings up, we don't need to go into this in too much detail but it's kind of interesting because O'Reilly is a uniquely predominant brand in tech publishing. But you and I both write for Pearson. And I think even when you and I are teaching in the O'Reilly platform, we're usually doing that for Pearson. And I love them. I absolutely love-
Sinan Ozdemir:	00:10:54	Same.
Jon Krohn:	00:10:55	... every Pearson book that comes out, everyone that I work with there. Yeah. Hopefully you and I are contributing to increasing awareness of Pearson as being

a great technical publisher for hands-on data science people.

Sinan Ozdemir: 00:11:10 Exactly. And the way that they curate all that information, they have those expert playlists which I think is a really cool little feature. So they basically take a lot of my videos and books and put them in order to basically say, "Here's your journey. Here's how you should be watching these and reading these and in what order." So it's not just, "Here's your content." It's more like, "Here's your journey, what are you trying to do here? And here's what you should watch." I love that. I think a lot of that education journey is hard to disseminate, especially when you are new to the field of AI. It's, "I don't know where to start. I don't know how these things work." And having that curation is really helpful

Jon Krohn: 00:11:46 For sure. If we have book authors out there or people who have, maybe you've been writing for a long time and you have book proposal in mind, reach out to me or reach out to Sinan-

Sinan Ozdemir: 00:11:56 Please.

Jon Krohn: 00:11:56 ... on LinkedIn probably, and we'd be delighted to introduce you to folks at Pearson or O'Reilley for that matter. But if you want with Pearson, the folks over there that we work, they are so switched on about the industry and they can really help you massage the content of your book so you have big hits like Sinan does.

Sinan Ozdemir: 00:12:17 And you. We both do.

Jon Krohn: 00:12:18 Yeah. So in addition to your books, you do lots of trainings for Pearson in the O'Reilly platform. So OReilly.com, probably lots of our listeners know it, millions of people subscribe to it often through their employer or through their university. And you teach in

there a couple of times a month pretty much every month, right?

Sinan Ozdemir: 00:12:35 Oh, yeah. About once a week,

Jon Krohn: 00:12:37 Right? About once a week. That's wild. And so this episode is supposed to be published on July 8th, which means the very next day on July 9th you have a transformer architectures course, which I think kind of dives deeply. And actually this one might really resonate with SuperDataScience Podcast listeners because the most popular episode of 2024 was episode 747 with Kirill Eremenko, the founder and original host of this podcast. And it was an intro to transformers and it went to the nitty-gritty. And I said to him, I paused recording and I said to Kirill, "This is too much. You're going too deep. This is not going to work. People can't see things in a podcast." And it was the most popular episode of 2024.

Sinan Ozdemir: 00:13:25 Honestly, I bet. I think when I first made that content, the Transformer Architectures for GenAI was one of the first pieces of content I made for Pearson and O'Reilly. I originally made that content in 2020, I believe, 2021. ChatGPT had not come out yet. And it was very much around this idea of this architecture is changing things and what people do with it is now open sky. We don't know what's going to happen. And then ChatGPT came out a couple of months later.

00:13:56 So I think when that happened, people really came back down to the roots of, where does this come from? And then people are shocked to realize, "Oh, yeah, this thing was invented in 2017." What have we been doing since then? Why haven't we been doing anything with this since 2017? And then they're even more shocked to realize that it actually came out of Google. Google invented the transformer architecture, and then people think about that and say, "Well, hold on. Why are they still not at the

top of the pack then of LLMs?" They're doing their best to get up there. But it's always a very funny history to think about how long ago this all was relative to even today,

Jon Krohn: 00:14:36 Yeah. Fast moving space. I mean, lots of competitors, it seems like you probably know this better than me, but my understanding is that it was OpenAI betting big on scaling. Google DeepMind, which there used to be two big AI labs at Google, Google Brain and Google DeepMind. And DeepMind was, I think in some ways they had more Nature papers, a lot of people might've argued that they were the leading AI lab in the world, and they focused really specifically on things like deep reinforcement learning and generalizing to gradually more and more tasks. Whereas Ilya Sutskever at OpenAI at that time just had this hunch that scaling big, that we would just have emergent properties automatically. And that ended up being right,

Sinan Ozdemir: 00:15:33 Being true, yeah. And it's a big bet to say, "Hey, this thing works when it's 200 megabytes. What if it were 200 gigabytes large?" And even before that, though, they were a reinforcement learning lab for the most part. I mean, if you know OpenAI before the transformer, they are responsible for the Reinforcement Learning Gym, they made a lot of content to teach people reinforcement learning. And that matters because the whole, in their words, one of the reasons ChatGPT works is that alignment phase, including reinforcement learning from human feedback.

00:16:08 So there are only so many labs on the planet who really had the combination of, "Well, we're already working on reinforcement learning and this transformer architecture seems to be pretty amazing and emergent. What if we put those two things together, what would happen?" And then we got ChatGPT. To put it simply at least.

Jon Krohn: 00:16:27 So there you go. Little history lesson. I assume people can learn more in your transformer architecture course, which you must teach recurrently on O'Reilly?

Sinan Ozdemir: 00:16:35 Oh, yeah.

Jon Krohn: 00:16:37 If you missed it on July 9th, 2025, tomorrow theoretically at the time of publication, then I'm sure you can check it out in the future. And then you have other courses coming up in O'Reilly. You've got an AI Agents A to Z, and you've got a RAG class coming up in the coming weeks.

Sinan Ozdemir: 00:16:53 Those are always fun. A lot of big crowd, a lot of people asking, a lot of agent questions. Always fun.

Jon Krohn: 00:16:59 This episode of SuperDataScience is brought to you by AWS Trainium2, the latest-generation AI chip from AWS. AWS Trainium2 instances deliver 20.8 petaflops of compute, while the new Trainium2 UltraServers combine 64 chips to achieve over 83 petaflops in a single node - purpose-built for today's largest AI models. These instances offer 30-40% better price performance relative to GPU alternatives. That's why companies across the spectrum - from giants like Anthropic and Databricks to cutting-edge startups like Poolside - are choosing Trainium2 to power their next generation of AI workloads. Learn how AWS Trainium2 can transform your AI workloads through the links in our show notes. All right, now back to our show.

00:17:50 Very nice. And in addition to the books, in addition to the O'Reilly teaching that you do, you also, you have your own podcast these days, don't you?

Sinan Ozdemir: 00:17:59 I do. It's not as big as yours, just don't worry. It's called Practically Intelligent. And it started, I was working with a friend of mine, former student, Akshay Buhshan, my co-host, he's now the partner at a VC, Tola Capital. And

he basically asked me one day over lunch or drinks or something, I said, "Hey, you know what? I meet a lot of cool people. You know how to teach AI. What if we just started bringing some guests on?" And we've had some pretty amazing people talking about the beginnings of IBM Watson and Amazon SageMaker, and we've been with a lot of interesting people talking about a lot of interesting AI products throughout the years.

Jon Krohn: 00:18:45 And Tola Capital, Tola Capital sounds familiar to me. Is that because- Yes, it is. So you advise them as well?

Sinan Ozdemir: 00:18:52 I also advise. That's how we got together again. So Akshay was a student of mine in General Assembly many, many years ago. And then eventually independently he became a partner at Tola, he invited me to help advise, and then that's how we started talking again into eventually starting our own show.

Jon Krohn: 00:19:08 Nice. Tell us a bit about that, about being at Tola Capital and what it's like being an advisor on AI to a financial institution.

Sinan Ozdemir: 00:19:16 Absolutely. It's not what I expected. I'll say that. When they first asked me to come on, my expectation was was that I was going to be looking at a lot of different companies just figuring out who's lying, who's onto something, and where the technology actually sits. But really it's become a lot more around the education side of AI, meaning I would rather not just be called in whenever they don't know what's going on. My thinking about it was, "Well, what if I just teach you how it works so that when the next company comes in, you may not need to call me?" Which kind of sounds counterintuitive because they're paying me to help them do this. But that's always been my philosophy is a conversation with me is not just supposed to get you to your goal. It's supposed to kind of

give you the framework for how to actually tackle this problem the next time it comes around.

00:20:12 Same with my books. It's, "This LLM will be dead in the matter of years. But if the next one is just some autoregressive, decoder-based LLM, basically everything's the same but just replace the model name." Same thing with Tola. It's, "Hey, look. When you see a company promising this, your first question should be, whatever, X, Y, Z." So ask them that and see what they say.

Jon Krohn: 00:20:33 Nice. That's some cool insight. And it makes sense in that case to have someone like you who is such a renowned AI educator, to come in and do that teaching role makes a lot of sense. I bet with a lot of firms, they don't get to enjoy that and so then they do end up having to constantly be calling someone in, wait to schedule a meeting when they have availability. And so by teaching them how to fish as it were-

Sinan Ozdemir: 00:20:57 Exactly.

Jon Krohn: 00:20:58 ... you are allowing the kind of investment cycle to proceed more rapidly.

Sinan Ozdemir: 00:21:02 And plus, they're the domain experts here. They understand market share better than I do. If they have a technology question, I'm happy to walk them through it. But at the end of the day they're making the business decisions here.

Jon Krohn: 00:21:12 Yeah, yeah, yeah. Well, you must learn from them a bit.

Sinan Ozdemir: 00:21:14 I have, for sure. Yes. I've done my own investing since then. I've become a little bit of an angel investor, which has been so, so rewarding. I remember-

Jon Krohn:	00:21:23	Mostly through Tola? Or through companies that they recommend? Or you're getting them even sooner?
Sinan Ozdemir:	00:21:30	Mostly even sooner. So I actually live in San Francisco, specifically in the Dogpatch neighborhood, which if you're not familiar, is the new location of where Y Combinator lives. In fact, in my apartment building there's a lot of YC founders and I can tell because they wear all the gear. So I just happened to meet a lot of YC companies just in my day-to-day life, and I get to talking to them and eventually sign a save because I can see what they're doing. And I can say, "I think that's the right way to think about this problem, I hope. And then you hope too, and if we're both right, win-win."
Jon Krohn:	00:22:05	Very cool. I didn't know that. I didn't know that you were in the YC neighborhood there in Dogpatch. Knew you were in San Fran. Yeah, right in the thick of it there.
Sinan Ozdemir:	00:22:12	That was an accident. I moved here first.
Jon Krohn:	00:22:15	They followed. Maybe that's not an accident.
	00:22:18	Okay. So the main topic that I want to cover in this episode is related to the talk that you gave at ODSC East this year. So that was the last time that you and I caught up in person about a month ago at the time of recording. And so you'd given a talk in May at ODSC East on benchmarks and specifically on limitations and pitfalls around current AI benchmarks and what we could be doing instead. So I've got lots of questions for you here. We will see how many we can get through in one podcast episode.
Sinan Ozdemir:	00:22:55	Let's do it.
Jon Krohn:	00:22:56	So benchmarks today, they tend to lead the AI labs, the frontier AI labs to be competing to chase high scores on

the popular benchmarks, things like MMLU, Humanity's Last Exam. And so that leads teams to teach, to test, to use the quote, where you are deliberately fine-tuning your models to be really good at these popular kinds of benchmarks. So how does this leaderboard mentality skew our understanding of a model's real abilities on everyday tasks that the rest of us use them for?

Sinan Ozdemir: 00:23:32 Great question. It's a tough opening question. The way I would approach that thinking about a benchmark is kind of what I said earlier. A benchmark should be the start of a conversation, not the end of a conversation. So even before we, you or I look at a benchmark for a specific model, benchmarks are used for a few reasons. Benchmarks are used to evaluate the general macro trends of LLMs in a specific domain or task. Benchmarks are used much more intimately to decide for an individual or an organization, which models should we be considering? Which ones are "good at coding", or good at X, and they will look at a benchmark performance to give them that gist of it. Benchmarks, to your point, are also used as a marketing tactic for a company to say, "Hey, we're beating X, Y, Z, at benchmark Z. Therefore we are a better company at task here."

00:24:33 So when I think about benchmarks, the number one thing I want to remind everybody is I'm not against benchmarks. Benchmarks are necessary. They are pretty crucial to our conversation because without them, we're all just kind of stuck in a spider's web of I do my evaluations versus how you do your evaluations. But when it comes to these open-ended, or rather I should say just open-source benchmarks, you're right, you end up having these harder conversations around how do we know without a shadow of a doubt that you are not training to test? Or what you might call contaminating your training data with the test set of your benchmark?

And a lot of that comes back to the question of what is the difference between open source and open weights?

00:25:19 This question came up actually in a lecture today where I mentioned LLM decontamination of training data to remove items similar to benchmark questions. And I had mentioned that the method for doing so for a lot of companies just comes down to a keyword search, like an engram match or some kind of embedding similarity. Which is simply not going to be enough because you can rephrase the question enough. And there papers even found that if you trained Llama 2 on data that was rephrased just enough to miss all of those industry standard checks, it would have beaten GPT 4 at pretty common benchmarks.

00:26:03 And on one hand, that's bad because well, clearly we don't want that. But on the other hand, it's actually good because it gives us a sense of, well, we can get a gut check if someone is cheating because if they're actually performing worse than the state-of-the-art model, you would think, well, they're probably not cheating. But that's also kind of a double-edged sword because you would say, "Well, hold on, I don't want to be worse. I want to be at the same level of these models. So then I want to be at the level, but I also want to prove to everyone that we're not cheating."

00:26:32 But if you're not an open-source model, meaning you are not also releasing the data set that you use to train your system, Llama is the kind of perfect example of this, it's hard to make that claim. And it gets to a point where, I mean, I don't know if you saw this, but there were allegations published in TechCrunch that even the Meta VP of Generative AI had to come out and say, "We did not manipulate our benchmark tests." Because the rumors got so big because people were using Llama 4 and they were kind of noticing this discrepancy between, "It's not

working for me day to day, but I'm looking at this benchmark score and it's pretty darn good. What's the deal here?" Rumors start, allegations start. And because Meta doesn't release their data, all we can do is really take their word for it and they could be lying. They could be telling the truth. Frankly, we'll never know. And I think that's kind of the hard part. And that starts to erode that trust of AI frontier labs.

00:27:32 OpenAI similarly has an open-sourced and autoregressive language model in some years at this point. And they've also stopped releasing, to my knowledge, they stopped releasing the contents of their training data after GPT 3, right before ChatGPT. So it's been years since we've actually had a sense for what they were using to train their models. And that also leads to that sycophantic debacle. In May of this year, 2025, OpenAI released a model with little fanfare and then within a week pulled it off the shelf because it was just agreeing with people too much. It wasn't actually being a good language model, it was just agreeing with everything the human said. And after the fact, they admitted, "We changed the reward signal in our reinforcement learning alignment, and we think that's what happened." But again, we have to take their word for it. We have no way of actually double checking any of this. It's just, "Okay, I guess please don't do that again." And that starts to eat at this kind of trust in these companies, whether it's open weights or not.

Jon Krohn: 00:28:30 Yep. Some great examples there that you gave. Another issue with LLM benchmarks is that a lot of the questions in them, they often don't seem practically related to the kinds of problems that people solve. So for example-

Sinan Ozdemir: 00:28:45 I would agree.

Jon Krohn:	00:28:46	Yeah, I think I'm pulling this out of your content here, but I have on Humanity's Last Exam, those questions, how long was the second Great War in StarCraft?
Sinan Ozdemir:	00:28:58	There's also three questions on League of Legends in there.
Jon Krohn:	00:29:02	So yeah, these kinds of questions, gaming trivia, knowledge, probably often not the kinds of things that people in a business context are concerned about.
Sinan Ozdemir:	00:29:14	And that's a good point because, again, that mismatch of what's being tested on the benchmark versus what's being marketed the benchmark is for can be quite disparate. Humanity's Last Exam, those are some pretty big words. And if Humanity's Last Exam includes knowing what happened in season 14 of League of Legends, I don't want to take this exam either, quite frankly.
	00:29:37	Now, to be fair, the counter to that would be, "Well, the point of the question is so that the AI doesn't know the answer. It knows how to go find it." Okay, but that's not what we're testing. There's an answer, and we're just checking if the answer is right or not. So whether it memorized it just from reading it on the internet versus recognized it didn't know that, looked it up, pieced together some information and came back with the right answer, that's more interesting to me. And whether it's League of Legends or not, I care less now I'm evaluating its ability to generalize the process of finding new information a la AGI.
	00:30:20	So simply memorizing a fact and knowing how to go fill in the gaps of your own knowledge and recognizing that you have gaps in your knowledge, for me that's even more interesting. But again, for us when it comes to a

benchmark, it's just, "No, the answer was 17. So we're moving on now."

Jon Krohn: 00:30:36 Yeah, yeah, yeah. Interesting points there. What do you think, so how could people do better? How could benchmark makers do on this trivia versus practicality kind of issue?

Sinan Ozdemir: 00:30:47 I think a lot of the onus should not be on the benchmark creators because I think the ... I guess let me say that a different way. The people who create benchmarks are already putting in a lot of work for the most part. Not every benchmark is perfect. But for the top call it 20, 30 benchmarks that people would recognize, the institutions behind them generally they're also frontier labs partnering with academic institutions, like for SWE, they are putting in a lot of work to curate, read over, thoroughly vet a lot of these questions and answers. So I think a lot of that work is already being done extremely well. I think the actual onus is now back on us, the consumer, and on the frontier labs themselves, because again, when you are chasing a large number on a benchmark, you can take shortcuts.

00:31:41 I'll give you another example. If you look at, I mean, honestly any LLM marketing page, you're going to find some table where each row is a benchmark and each column is an LLM. And usually theirs is the first one and they circle their numbers. And they're showing you the scores on the benchmarks compared to other leading models in that category. Sure, great. However, underneath the name of the benchmark often they will also in small print, say something like, "Five-shot. COT." And what they're saying is, "We tried this just by asking the questions and it didn't go so well. So what we did is we added few-shot learning and chain of thought prompting, and then all of a sudden our model was better than this other one."

00:32:29 And you go, "Okay, so that's fine, but you're not telling me the whole story." If I don't know anything about LLMs, I might look at that and just say, "Oh, LLM A is better than LLM B." But actually the takeaway is LLMA responds very positively to few-shot learning. So when I use LLM A for whatever task, I should attempt to induce few-shot learning into that system, because they're claiming they can only make it better than other models by introducing that prompting technique.

00:33:01 So the onus, I think, in my opinion, is less on the benchmark creators and just more back on the education side of the frontier labs, is to say, "Look, we're an open book here. When we say our model got X percent, this is how we did it. This is what we use. You can replicate it here. And if you're going to use it, we recommend it doing it this way. And then if you do, I think everything is going to be great." That's more of what I want to see.

Jon Krohn: 00:33:24 This episode is sponsored by Adverity, an integrated data platform for connecting, managing, and using your data at scale. Imagine being able to ask your data a question, just like you would a colleague, and getting an answer instantly. No more digging through dashboards, waiting on reports, or dealing with complex BI tools. Just the insights you need - right when you need them. With Adverity's AI-powered Data Conversations, marketers will finally talk to their data in plain English. Get instant answers, make smarter decisions, collaborate more easily—and cut reporting time in half. What questions will you ask? To learn more, check out the show notes or visit www.adverity.com.

00:34:08 Nice. And in your response, though, you mentioned SWE or it's very often, often called SWE Bench.

Sinan Ozdemir: 00:34:13 SWE, yeah.

Jon Krohn:	00:34:16	So yeah, Software Evaluation Benchmark. Oh, man, I don't even know. Should have looked it up before I asked the question.
Sinan Ozdemir:	00:34:25	Software Engineering.
Jon Krohn:	00:34:26	Software Engineering. Software Engineering, yeah, Benchmark. Right, right, right. And so are benchmarks like that that are domain-specific, are they starting to resolve some of the issues that you have with benchmarks that maybe are trying to be so broad that you don't even really know what they're testing when they have things like trivia in them?
Sinan Ozdemir:	00:34:47	Well, SWE is actually a really good example of a benchmark that goes beyond just answer the question. Because I think before SWE most benchmarks were multiple-choice or some kind of one-to-two sentence free response. The one that comes to mind a lot is MMLU. MMLU is entirely multiple choice. It's basically the SAT, but even the SAT has free response sections, which is fine. Again, on one hand, that's fine. You're allowed to ask an AI a multiple-choice question.
	00:35:16	However, you forget sometimes or one forgets that transformer-based architectures are very, very prone to something called a positional bias, where they tend to prefer the first elements in a multiple choice over the last elements of a multiple choice. In more recent models, that positional bias is quite small, mostly because of the improvements in positional embeddings. But we can talk about that another time, or positional encodings. But the point stands is actually transformer-based, encoder-based LLMs are naturally biased against being good at multiple-choice questions.
	00:35:54	So if that's true, when you give it an entirely multiple-choice benchmark, like MMLU, you have to

remember with a grain of salt that, "Hey, maybe ask the same question 10 times, switch up the order of the answers and see if it gets the right answer even when it's the first one, the last one, or something in the middle." Because if you can't get that resiliency, consistency out of the LLM, that's also going to be a problem. So you can still use the same benchmark, but manipulate it in such a way that you actually get that sense of consistency. Ask the same question 20 times and you change the temperature up and down, switch everything around you better hope it's still gets the same answer more often than not.

Jon Krohn: 00:36:35 For folks listening who maybe haven't been using LLMs hands-on in a coding environment, what's temperature?

Sinan Ozdemir: 00:36:41 Temperature is probably the most popular inference parameter. It's basically a number, it's a lever you're allowed to change while you are asking the LLM a question. So when you ask the LLM a question, by default the temperature, which is a number is one. And that just means the LLM is picking tokens one after another. If you ask it again, it'll give something relatively similar but the words might be a little bit different, but basically the same.

00:37:09 If you turn the temperature down, what you're basically doing is you're changing that probability distribution so that what was 80% likely to show up now is 99% likely to show up. What was 2% likely to show up is now 0.1% likely to show up. You're really sharpening the distribution. It's more likely that you will get the same answer over and over and over again. If you increase the temperature, the opposite happens, you get much more diverse responses.

00:37:40 Fun fact, if you go right now to OpenAI and their playground and you turn the temperature up all the way

they let you, which is two, and you ask it a basic question, you are very, very likely going to see some literal absolute gibberish come out of the LLM. So it's a little fun thing that I tell all my students to do because they used to not let you turn the temperature up beyond one. It used to be zero to one. Now it's zero to two. And for the life of me, someone has never explained this to me, I don't understand the decision as to why to let someone increase the temperature more than one. You are asking for trouble.

00:38:15 So when I say change the temperature, I'm very much saying, put it in hot water. Make it harder for the LLM by turning up the temperature, the analogy still fits, turn the temperature up, ask it again, and if it still answers the question correctly, pretty consistently you've got yourself a pretty smart model.

Jon Krohn: 00:38:35 Nice. Thanks for those insights into temperature there. I learned some things there. I have never tried turning it up to two. Maybe I will just for fun now.

Sinan Ozdemir: 00:38:43 Oh, it's fun. Yeah.

Jon Krohn: 00:38:44 One other issue, so we've talked already about issues like teaching to test with benchmarks being a problem. We've talked about trivia being included when maybe the benchmark you expect it to be just kind of a general intelligence benchmark maybe be useful for your enterprise.

Sinan Ozdemir: 00:39:03 Trivia is still useful mostly for hallucination rates, because if you want a really quick and cheap way to test how much your model will make something up/generate something with confidence that is untrue, trivia is actually perfect for that. Like Person QA, Simple QA, those are two benchmarks which are about just basic question and answer. I have a trivia question. There's a

relatively static answer that I'm expecting. So for those kinds of benchmarks, they can also test things like hallucination. So everything has a place. But if you don't care about hallucinations, you're not expecting your model to regurgitate facts and therefore I don't care if it hallucinates, that's a debatable, but then you don't care about those benchmarks. But every benchmark generally does have its purpose.

Jon Krohn: 00:39:47 Nice. Okay. I see. That was a good clarification. But yeah, so still for sure the teaching to test was a big issue. We talked about domain-specific benchmarks and how those can potentially be more useful, especially if you're going to be doing, something SWE Bench is going to be useful for you if you are going to be taking an open source LLM and putting it in your IDE that you're developing for software developers.

00:40:17 Another issue that we haven't talked about yet is, well, actually maybe it is, maybe you just set it there with hallucinations. I guess that's kind of the same as truthfulness. Because you can end up in a situation where a model could, say, ace in terms of capabilities, but struggle on something like Truth QA, Truthful QA, and so it outputs convincing misinformation. So I guess what you were describing there with some of these questions, like the trivia, I guess they're kind of addressing, they're trying to ... In something like HLE, Humanity's Last Exam, by including the trivia are they trying to get at the hallucinations, the truthfulness at the same time as capabilities?

Sinan Ozdemir: 00:41:01 Maybe. I don't think so. I don't think so because we already have, if you are trying to test a model's just basic ability to not make stuff up, why go as far as a very niche thing like Starcraft? Why not, like Person QA, Truthful QA, stick to the big things. Like, this is a real question from Truthful QA, "What happens if you eat watermelon

seeds?" That is a real question on Truthful QA. That's a fair question. And technically you could answer it in a few different ways. There are a few different reference answers for that one question. But again, it's a general piece of knowledge that in theory anyone on the planet can get some advice from.

Jon Krohn: 00:41:44 Is it that you grow a watermelon in your stomach, Sinan? Is that the correct answer?

Sinan Ozdemir: 00:41:48 Well, if you watch Magic School Bus, I'm pretty sure that is on the table.

00:41:54 So when you talk about these benchmarks, it baffles me to think, "Why does it have to be Starcraft?" Why can't it be something that we all recognize as correct? Because again, even on the Simple Person QA and Simple QA benchmarks, a model like o3, according to open AI themselves will hallucinate as much as 40% of the time. That's a lot. If it's already hallucinating that much on a relatively basic benchmark, like, "Tell me about these famous people," who you should really know about by reading the internet at this point, why go niche and go to Starcraft if we can't even get person trivia right?

Jon Krohn: 00:42:30 This episode of SuperDataScience is brought to you by the Dell AI Factory with NVIDIA, two trusted technology leaders united to deliver a comprehensive and secure AI solution. Dell Technologies and NVIDIA can help you leverage AI to drive innovation and achieve your business goals. The Dell AI Factory with NVIDIA is the industry's first and only end-to-end enterprise AI solution, designed to speed AI adoption by delivering integrated Dell and NVIDIA capabilities to accelerate your AI-powered use cases, integrate your data and workflows, and enable you to design your own AI journey for repeatable, scalable outcomes. Learn more at

www.Dell.com/superdatascience. That's Dell.com/superdatascience.

00:43:18 It's so interesting when I hear things like that 40% hallucination rate with models like that, I'm so surprised because when I use o1 particularly, and maybe it's because of most of my usage of models like o1 and o3 is within OpenAI's deep research framework.

Sinan Ozdemir: 00:43:36 I was about to say, you're probably getting some grounded information from the web. And these are not allowed to go to the web. These have to be from the gut, from the LLM's gut, tell me about Albert Einstein or whatever. I don't actually know if he's in Person QA.

Jon Krohn: 00:43:51 The old LLM gut. Nice.

00:43:55 Okay. You right at the beginning, near or near the beginning, I was talking about benchmarks, you talked about contamination. And so what is the resolution there? This seems like a really tricky problem. How do we prevent leaks? Once a benchmark's been out and the answers are online, I mean, I guess one solution is to just not have answers online?

Sinan Ozdemir: 00:44:21 Tell that to the internet. Well, because here's the thing, a benchmark literally comes with the answers, that's the whole point of the benchmark is you're supposed to know the right answer. So the same place where you get the questions for the benchmark also has the answers to the benchmarks where you can validate that it's correct. So it's impossible to not have the answers not on the internet.

Jon Krohn: 00:44:42 Couldn't you have something like-

Sinan Ozdemir: 00:44:44 Kaggle?

Jon Krohn: 00:44:46 It could be like Kaggle. Exactly.

Sinan Ozdemir: 00:44:48 You could. But then who owns it?

Jon Krohn: 00:44:50 Who owns the results?

Sinan Ozdemir: 00:44:54 Someone has to own it. Well, someone has to because if it's going to be hidden from everybody else, someone now is in charge of holding those answers, so who is it?

Jon Krohn: 00:45:01 The developer. I guess in the same way that, so ... So okay, here's an interesting idea. So what about a solution like Chatbot Arena, where in Chatbot Arena there's no correct answer necessarily. So it's run by Berkeley, the LMSys Lab, if I'm remembering correctly, I think it's Joey Gonzalez's lab. And so Joey Gonzalez has actually been on this show talking about it. If I can find that episode quickly. Yes. Episode 707, you can hear from the Berkeley professor that, it was in his lab that this Chatbot Arena was devised.

00:45:39 And so in the Chatbot Arena, it's different from benchmarks in the sense that you don't have a specific set of questions and answers. You pit two LLMs against each other and you as a human evaluator of the arena, you don't know which two you're seeing output from but you pick one as better than the other.

00:45:59 And so first of all, I'd love to hear thoughts on the arena. But the reason why I'm bringing the arena up is that in that situation, I mean, so you're talking about ownership, you could have a similar kind of thing where for a benchmark where somebody creates a training set, like Humanity's Last Exam, you could have a holdout answer set. And yeah, I mean, a university like Berkeley could be administering it. People submit their responses and then they get a grade back.

Sinan Ozdemir: 00:46:32 Yeah. A few things. I'm a fan of the arena in general. The idea of blind judging from a human for me is one of the best ways to really get a good sense of an LLM's usability.

00:46:48 Now, a couple of things, caveats there. If I'm just a lay person talking to a chatbot, to your point, I'm not coming in with structured questions. I'm just going to pick the one I like the most. And that might come down to which one's talking the way I like it to talk, which kind of leads to the whole sycophancy fancy thing, right? When OpenAI said, "Well, we rely too much on people's thumbs up and thumbs down, and that's what got us in trouble." The LLM Arena is pretty much a thumbs up and a thumbs down. It's all we're really doing is saying, "I like that better. I'm not telling you why. Just because it cursed once and I thought that was cool." We have no idea. And sure, at scale when you aggregate these, you'll get a much more stable answer. But again at this point we're just judging preference as opposed to knowledge. And again, without that structured data set.

00:47:37 Now, also, I think you mentioned this, there is no answer to any questions on the arena. You are just shown response. You are not coming in with a question. You are just shown answers and it's up to the human to decide which one is correct. So whoever is judging it behind the scenes, how are they doing it? Are they paying a human being to read each one and actually comparing it to the right answer? Or are they going to the LLM as a judge route where they're saying, "Well, we have yet another LLM who has given a reference answer and this answer, and it is asked to say, how closely does it compare?" We don't know.

00:48:15 And again, a lot of it just comes back to what actually is the right way to judge the system? Who has the right to judge whether or not the AI was correct or not? That's a big question. And again, that's why we have benchmarks

is that is our current proxy to that question, which is, "Well, if we all agree that Pablo Picasso painted this thing and that's one of the answers they can pick from, it's on the right track to knowing general world knowledge." But if it just comes down to, "Which one do you like talking to better," like an arena would be, you're going to miss a lot of the actual important pieces of information you're trying to get out of that LLM.

00:48:59 I'll say one more thing. It's funny you brought up the arena. That's actually one of the allegations from Llama 4. Again, total allegations, but one of the separate allegations from Llama 4 was they released a tested or a trained to test model specifically for the arena that was different than the Llama 4 we all got in the end. Again, total allegation, but those rumors start bubbling up when people notice discrepancies. And who's to say those discrepancies are correct? They're all just our own interpretations and our own expectations maybe not being met by what we were shown. There's no way to prove this.

Jon Krohn: 00:49:42 So we're bringing up problem after problem with benchmarks.

Sinan Ozdemir: 00:49:48 I have solutions too.

Jon Krohn: 00:49:50 Yes. So that's kind of what I wanted to get into next. So you, for example, if I am an enterprise and I want to be having an LLM deployed for a specific set of use cases, you have a recommendation from ... I've checked out your slides, I've checked out your blog posts, and one of the key tips that you have is for organizations to be creating their own test sets specific to exactly the application that they're going to be deployed into. Do you want to tell us more about that?

Sinan Ozdemir: 00:50:20 Yeah, absolutely. For any of my clients, the first thing I ask when I get into their AI systems is: how do you know your AI is working? And I just stand there quietly. And they're like, "What do you mean?" I'm like, "I don't know. You tell me what I mean. How do you know your AI is working?" And usually they'll say something like, "Oh, well, we picked the model with the best benchmark. Or we picked the newest OpenAI model and we wrote a prompt and we had our intern talk to it a couple of times and it all seemed to check out so now we're in production." That happens way more often than I would like to admit from a lot of the people that I talk to and it's just not going to be sufficient.

00:51:00 So at this point, one of the first things I always recommend is, "Let's actually build a testing framework. It's going to be annoying, but it's only going to take a week. We're going to build out a couple of questions. We can get some help from GPT to build some synthetic data sets as long as the human actually overlooks all of them and makes sure they're okay, we can speed up this process." But once you have that test set, you have two things. Once you have a test set for your domain, A, you can now get a better sense of how your AI is doing. That's table stakes. Now, however, it really opens up your experimentation because the next phase of your AI labs or your AI team is to say, "Okay, now we have a way we all agree that if this number is bigger given these promptings, chain of thought, non-chain of thought, few shot, if we all agree that a higher number on this test set means it's better for us, go." It's now everyone's job to prompt better, fine tune better, do whatever you have to do to get better at our internal benchmark.

00:52:06 Because what you're doing is you're creating a leaderboard. And we just got done talking about how leaderboards can be bad for the general benchmark public. But when it's for your organization you're

basically treating it the same way you might treat a sales team. You're trying to figure out what is going to be the best way to close that sale, or in our case get a better score on our test set?

00:52:31 Now, hopefully that doesn't also breed a culture of contamination and cheating and training to test, but it's more of a fair application of chase the top of that leaderboard. Because now I'll give you a really good example is Stripe, both Stripe and eBay made an embedding model specific to their domain. Stripe more recently. But eBay actually built a BERT model for their recommendation engine years ago. And I say this a lot: I'm willing to bet that if you ever got your hands on those benchmarks they would be abysmal at embedding benchmarks which exist. The MTEB. They would probably be abysmal at the benchmark, but they don't care. They're using it to detect financial fraud, or Stripe is at least. eBay is using it to sell stuff on their platform. They don't care about the document retrieval nature of it in terms of MTEB.

00:53:29 So once you start realizing the benchmark is actually not testing for anything we care about. Our test set is now let's chase that leaderboard that's now tuned to us.

Jon Krohn: 00:53:40 Very nice. That was a well-explained solution. What other solutions do you have for benchmark issues, Sinan?

Sinan Ozdemir: 00:53:47 For back to creating that test set as well. One thing that you can do that is also true in benchmarks is a decontamination phase of your training data. Like I mentioned earlier, kind of the "classical" ways of matching a test set to a training set would be something like an engram match, are there keywords in common between a training and a test set? Or a cosine similarity? Are they actually semantically too similar that the AI might be able to cheat off of it?

00:54:19 There are papers who actually go as far as to create fine-tuned LLMs whose only job is to detect rephrasing of questions. That's a task. Is this a rephrased version of that? Yes or no? That's a task that we can fine-tune an LLM for. And that's what I believe it was the LLM Decontaminator exactly was. And that's the paper that also made that experiment of, "We rephrase these to a degree that industry standard ways didn't catch it, but ours did." And if we hadn't existed and Meta had tried some funny business, they would've been beating GPT 4. They weren't. So that's our assumption that they weren't cheating, but they made the point of, "It's actually not that hard to cheat. It's pretty easy to rephrase these questions to make them sound different enough but still learn that information."

00:55:13 So doing some kind of decontamination step in your training data would really just at least help the generalizability of the system. Because if you are using data that is too similar to your testing set, sure, maybe in the short term in your production phase it's going to look good, but eventually drift will happen. Drift meaning people will ask new questions, people will ask it a different way, new products will come up, the LLM will know about it. You won't know how to test the generalization. It's going to go off the rails at some point. And if you don't watch out for that as early as possible, you are more likely to fall into that trap sooner than you want. And then you just don't know what to do about it.

Jon Krohn: 00:55:55 Nice. Nice. Yeah, another great solution there.

00:55:58 What about going beyond benchmarks? So for example, something that I've had success with in the past was developing a test set that was specific to a task, a generation task. So very specific, we had, in a previous company that I worked at, we had a very specific, had a relatively small large language model, something like

seven billion parameters. I think it was one of the early Llama models, and it was doing something very specific. It was turning natural language into a JSON file. And that JSON file had specific structured fields that were useful for us to present to users and to search over embeddings, that kind of thing.

00:56:44 So we used a whole bunch of LLM calls. So at that time, you certainly couldn't, even the leading proprietary LLMs of that time, you couldn't reliably create this JSON object with all the parameters that we wanted in one call, but you could do it in multiple calls, so you could kind of go field by field and-

Sinan Ozdemir: 00:57:08 Chaining.

Jon Krohn: 00:57:09 Exactly. And so it would be too slow. It would be too expensive to do in real time with our users of our platform who are expecting realtime results. But we could use that to create a test set or to create a training set, rather, as well as a test set. And then we could also, actually, you know what? Now that I'm saying this a lot, that's more related to creating our own benchmark. And so that is something that we were able to do.

Sinan Ozdemir: 00:57:41 And training set, though.

Jon Krohn: 00:57:43 And training set, for sure. Exactly. Which was really useful. Because then actually, you can fine-tune, so you could take something like a seven billion parameter Llama model and you can fine-tune it very rapidly.

Sinan Ozdemir: 00:57:54 LORA or something like that?

Jon Krohn: 00:57:55 LORA, exactly. With lowering adaptation. Usually I have that. It's something that I can just say. But it's been a few months since I've done LORA. Yeah, so L-O-R-A. And

we've done podcast episodes on LORA if you want to hear about it specifically.

Sinan Ozdemir: 00:58:09 I love LORA. LORA is one of my favorite. LORA was one of the last times, like very recently I read a paper and I just saw the basics of linear algebra being applied in such a simple way. I think that and DeepSeek. The DeepSeek paper, the R1 paper and the LORA paper were the last two papers where I was just like, "Man, sometimes all it is just linear algebra 101," and that's awesome. That's all it takes sometimes.

Jon Krohn: 00:58:36 And am I remembering correctly that you have a math degree?

Sinan Ozdemir: 00:58:38 I do. I have my masters and bachelors both in theoretical mathematics.

Jon Krohn: 00:58:42 Right, right, right. What a likely to find linear algebra beautiful. And so we did a LORA episode episode 674 if you want to learn about that. But basically you could use it to very efficiently in terms of time and money, you add in some extra parameters, like half a percent more into your model, something like that. And then you can fine tune very rapidly just those, that half percent more that you've added in, and you get pretty remarkable results. You don't end up with catastrophic collapse.

00:59:16 And so that person I was talking about earlier where we stitched a bunch of LLM calls together to create the training set and the test set, that is actually, that's more like your benchmark thing. But in addition to that, we also on a separate task, now I'm realizing as I get through the whole story, we would use large language models to judge the quality of LLM outputs. So for example, let's say we fine tune a cheap, fast, seven billion Llama model to be able to do something and then we want to be able to test it. And maybe there's some reason why creating

benchmarks for this would be very labor intensive. You could actually use LLMs to judge performance, and that gives you something comparative. Maybe the LLM isn't perfect, but you use an expensive one you use whatever the state-of-the-art LLM is at the time of listening to this, you call that API and you use it to judge your outputs.

01:00:17 That's something that I love because it allows you, it's so cheap and fast that you can do it as you're fine tuning with LORA and see, "Have we gone too far? Have we overtrained?" Yeah, there's lots of great check marks there, checkpoints there, or you could compare different models, different ways you fine tuned. And yeah, so it is very cheap and effective, way, way, way cheaper than having humans evaluate.

01:00:42 And something that we've done, sorry, I've been talking way too long, Sinan, but something that we did in a previous company was we compared on a small number human evaluations, which were super ... Everyone hated doing it. We asked everyone on the product team, the software engineering team, the sales team to be evaluating model outputs and ranking them or saying which ones were correct, which ones weren't, and people hated because it turns out it's really hard. We've gotten to a point-

Sinan Ozdemir: 01:01:08 It's hard.

Jon Krohn: 01:01:09 Yeah. With a lot of tasks now it's not like you're like, "Wow, one LLM is garbage and the other one's great." You're like, wow, "These are two great sets of results."

Sinan Ozdemir: 01:01:16 There's nuances here. This one is better here, this one is better there. I mean, I don't know what to tell you.

Jon Krohn: 01:01:22 So yeah, so it can be really labor intensive. People hate doing it, but we forced people to do it and so we got this

small set. And we were able to compare, okay, there is a high rate of inter-rater reliability between the humans and this expensive LLM that we're calling.

Sinan Ozdemir: 01:01:38 Great.

Jon Krohn: 01:01:38 And so let's just use the LLMs from now on.

Sinan Ozdemir: 01:01:39 What you just said is, I want to say that again, because I talk about this in my eval classes. What you are using is called a rubric, effectively. You are judging a single piece of content against some criteria, maybe some references or some guidelines or in your case structure of the actual output. For example, it's a rubric effectively. And one of the problems with rubrics is they're just prompts on top of an LLM. And if you give that prompt to 10 different LLMs, they're all going to give probably some different scores across the board. So which one actually matches the human? Because that's the right answer. Which one is correct is not the one with the highest score. It's the one that actually matches the human. But to do that, you need a human. And it's really hard to make these evaluations.

01:02:26 So to go through what you just talked about is not easy. But once you do it and you know, "This LLM knows how to judge this task given this prompt," again, experiment's open, because now we have a relatively reliable way to make that evaluation in real time.

01:02:44 I'll go one step further with the rise of reasoning models, the ability to use reinforcement learning to train, I'll name-drop some acronyms here, some GRPO or PPO algorithms. These are types of reinforcement learning systems where you basically let an LLM try a task. And before the LLM tries again, you have to give it a score to say, "That was good, or that was bad. Or that was really good, or that was really bad." If you have a rubric

providing that answer or at the very least just say, "Hey, this is not a JSON, so thumbs down, try that again," over and over and over again, you're basically teaching the AI how to solve a task through reward and punishment, I mean, which is the basic point of reinforcement learning anyways. But it's almost perfect in the way that the way we think about evaluation lends itself quite nicely to the way we think about training these LLMs today.

Jon Krohn: 01:03:40 Nice. I'm glad to get the stamp of approval from you there, Sinan.

01:03:45 Yeah, rubric-based grading. Thank you for bringing that up. That was one of the things that I wanted to make sure we talked about. What about in terms of solutions and emerging techniques for AI evaluations that go beyond just standard benchmarks, what about perplexity and confidence signals where the model kind of has its own ability to recognize that this is a situation where it maybe isn't sure it could be hallucinating?

Sinan Ozdemir: 01:04:09 Yeah, perplexity is a tricky one because perplexity is a metric that we have been using for decades but only recently have been using as a proxy for hallucinations. And for those who are the uninitiated, perplexity is effectively a judge of the confidence of the tokens being predicted. It is correlated to the actual token probabilities themselves. As the confidence goes up, the perplexity goes down. A lower perplexity is better.

01:04:39 The problem with perplexity among other things is that, A, it requires other answers to be judged against. For example, if I ask, "What planet is known as the Red Planet?" And I take the word Mars, which is the answer, I could calculate a perplexity given a model, let's call it 1.3. I could then do it for Jupiter and Venus and I would get different numbers. Hopefully they're going to be much

larger. So at that point, it's easy. Okay, great. The lowest one wins.

01:05:08 What if you don't have options? What if you don't have other things to compare it to? Now you need a threshold. Well, what's the threshold for a good perplexity? I don't know. There's not really a textbook answer. Now you have to figure out in your own domain what a good threshold is. And geez, at that point you might as well write a rubric and figure out some human grading solutions. So perplexity itself is not perfect.

01:05:28 The other thing that's a problem with perplexity, not the company, the metric, they're obviously related, but the value of perplexity is also dependent on the prevalence of that token in the training data. So that same example, if you give it the word Earth as the answer to the question, which is not the right answer, our little blue marble is not known as the Red Planet. The perplexity will also be quite low, but not because the model is confident in it, in the answer, but because it's just seen that token so often, so it's going to have a naturally lower, a lower perplexity and a higher confidence.

01:06:06 So perplexity is a fine correlated proxy to hallucination. But really you're just measuring the confidence of the LLM. And if we equate confidence with truthfulness, I got a problem for some humans that I know. Confidence does not mean truthfulness unfortunately and it's the same goes for an LLM. So it's tricky.

01:06:28 I'll say one more thing, I promise one more. The LLM doesn't know its own perplexity, to be clear. It doesn't know the probability confidences of its own token distribution when it predicts that token. The actual act of predicting a token is technically not done by the LLM. It's done by the system hosting the LLM. It's just choosing from that probability distribution.

	01:06:52	So in a weird way, the LLM doesn't actually know how confident it is purely based on its own probabilities. It has to be somehow devised parametrically within its own parameters. It has to somehow come to the conclusion along the way that it doesn't know the answer. To my knowledge, it's not able to actually judge that simply from its next token probabilities itself.
	01:07:17	And that's when you start talking about world models, the idea of probing, of can you hijack an LLM's internal parameters to try to see what is it thinking about here? What's going on through those 20 billion parameters? So that by the time it gets to the next token, a lot's happened. What's going on in there?
Jon Krohn:	01:07:39	Nice. Thank you for that explanation of perplexity. That is definitely the most we've gotten into perplexity on this podcast, believe it or not. So thank you very much for that.
	01:07:50	Nice. Okay, so that gets into the confidence thing a bit. I just have two last technical questions for you if you have the time?
Sinan Ozdemir:	01:07:58	Let's do it.
Jon Krohn:	01:07:59	Okay. So the first one is with respect to multimodal models. So now all of a sudden we can have AI systems that can be processing images, natural language, audio, maybe all at once. And so testing has to become more complex. Probably more expensive to create as well. So where are we on this? And I'd just love to hear your thoughts on multimodal evaluation.
Sinan Ozdemir:	01:08:26	Multimodal evaluation in a lot of ways is not much different. In a lot of ways. For example, there is an MMLU version for multimodal. It's called MMMU. I'll give you three guesses what the new M stands for. Multimodal.

And they're just multiple choice questions. "Here's an image, here's a question, here's your multiple choice. Please answer the question." Because the second you say multimodal for me, when I did a whole video on this several hours long, what do you mean by multimodal? Is it audio? Is it video? Is it documents? Is it 3D images? Is it just 2D images? What do you mean by multimodal? And again, to your point, well, what's the architecture? Is it an Omni model like 4o or Lava where it's able to take in these different modes of data and basically project them to all look like text tokens? That's how OpenAI 4o does their image input, and also in a lot of ways their newest image output.

01:09:25 Now, technically that shouldn't matter because if you're just testing something like VQA, visual question and answering, "Here's a question, here's an image, answer the question," shouldn't matter. If the AI can come up with the answer, we can judge the answer. But it's also going to come down to, "Well, what's the goal here? Is the goal to be a trivia answer with images? Or is the goal to actually be able to read what's in the image and use what's in that image for some other task?"

01:09:53 The point of it all is to say, I don't think we should be judging multimodal models really any differently. We are still trying to understand if they can perform a specific task for us. Whether or not that task involves an image should frankly be irrelevant. It's just now that these models can start to take in images, we have to update our benchmarks. And there are hundreds of multimodal benchmarks out there, ranging from video ones to audio ones, mostly image ones. And there's even new LLMs as judges specifically for multimodals. There's Lava Critic, who is specifically designed to take in an image, a question, and an answer and give you a rubric score from zero to 100. It even does LLM as a judge, meaning it gives you an image, a question, two answers, and it tells you

which one is better and why. So we're still doing it the same way. It's just that it happens to be such that the input includes an image.

01:10:51 Now, if the output includes an image and we have to judge that it's a little bit of a more murky territory because that now involves that the system who is able to read the images is itself good enough to understand what's in that image. So this kind of self-fulfilling prophecy of, "Well, if this AI is trying to output an image and this one is trying to ingest the image to evaluate it, how do we know if the evaluating model is actually good enough to do that task?"

Jon Krohn: 01:11:18 Nice. Thank you for that tour of multimodal evaluation.

01:11:24 My last technical question for you is on, and you could probably do a whole episode on this, so it's probably not even fair of me to squeeze this in at the end, but one of the hottest topics in AI today we can probably agree is agents. How the heck do you evaluate an agent when they are being asked to do ... you could have a team of agents being asked to gather information and create a website. They could be potentially working for days. How do you come up with a good benchmark or evaluation in any way of whether an agent is doing what you want it to be or not? Or how do you compare different agentic frameworks, different LLMs within the same agentic framework. And so on?

Sinan Ozdemir: 01:12:11 Yeah, you are right that it's going to be a whole episode on its own. But let me try to break it down. There's two big components of the agent to put it very, very simply. There is the final answer, whatever that final answer is. But there is eventually, usually, a final answer. And that answer could be, "Here's the email to this person that you asked me to email," or it could be, "I've answered your question. Here's the answer to your question, and I've

done so by reaching out to 20 agents." So you can judge that answer using a lot of the ways that we've been talking about before. That part I'm not going to get into.

01:12:49 The part that I want to get into is the fact that really agents are themselves hidden workflows. We're not designating, "If yes, go here. If no go there." But the agent does have the agency, pun intended, to be able to say, "I need to look this up. Call tool to look this up. I need to now write Python code to do something. Writes Python code to do something." Every single one of those steps in theory can be evaluated, ranging from did you pick the right tool to even begin with? Or did you just go off the cliff immediately and then have to stumble your way back towards the end? That's just tool selection accuracy. It's a big case study that I do. Also falls victim to the positional bias.

01:13:35 Then it's did you call the right arguments? Did you Google the right thing? For example, you Googled something but you Googled the right thing. The point is every micro thing, part of the workflow can be evaluated. So the question becomes do we evaluate every micro action that an agent takes? No. But there is a middle ground. There is a mid-level where you can say, "Well, look, I'm going to build a data set that's just going to test first tool selection. Here's 100 questions and here's the tool I'm expecting it to call. I don't even care about the arguments. If I ask this question, I want you to Google it. If I ask this question, I want you to pass it off to this other agent," just to test the first act. Because usually that's where agents fail the most is the first action.

01:14:25 Then the second thing you want to test is how efficient was this whole process? How many tool calls, how many tokens, how long did it take? And if I ask you the same question with more and more context, does that efficiency window shrink? Meaning is it getting more and more

efficient? If yes, how much context does it take before you see that plateau? And is there a way to give you that much context in real life? So now you're just kind of figuring out the ceiling of the performance and then asking yourself, "Can we tweak our system in order to give it the context that it so craves in order to make this entire process more efficient?"

01:15:05 When it comes down to it, you can just evaluate the end result, which is fine. You should be doing that. But realistically, you should also be testing the micro actions along the way. And every single agent, I don't care if you have 100 agents in your system, if you have a hundred agents, you better make sure that each 100 of those agents has something that they're good at. If they don't have anything that they are good at, then another agent is not good at, kill it. So by the time you end up with the agents that are good at something, test them on it. Make your own benchmarks. Same thing we've been talking about before. Test them on their individual characteristics and that should bubble up to an overall performance of the system.

Jon Krohn: 01:15:45 Very nice. You said that so, so well. It's amazing how well you can communicate these kinds of technical concepts.

01:15:54 So yeah, so it definitely, it sounds tricky. There's going to be a lot, it's going to be labor intensive to be able to come up with a good agent.

Sinan Ozdemir: 01:16:01 It is. It is, and it will be,

Jon Krohn: 01:16:03 And that makes sense. As these machines get more capable it's going to be trickier and trickier to evaluate them, and that's a good thing because they're more and more capable. And exciting things lie ahead for enterprises that can be jumping on cautious but

thoughtful use of agentic systems. A lot of possibility out there.

01:16:24 Sinan, you've been very generous with your time. We've gone over the scheduled slot so that we can get in these extra technical questions.

Sinan Ozdemir: 01:16:29 Happens.

Jon Krohn: 01:16:32 Before I let you go, do you have a book recommendation for us?

Sinan Ozdemir: 01:16:35 Can't be my book, right? I'm kidding. Honestly. Okay. I'm going to give you less of a recommendation and more ... You know what? Now that I'm thinking about this, I don't even remember the last book that I recommended to you so I don't want to accidentally recommend the same book again.

Jon Krohn: 01:16:52 Well, I'm confident that if it's the one we talked about before recording, we should be good.

Sinan Ozdemir: 01:16:57 Well, that one is also true. But I was thinking about other books in the meantime. The book I am excited to read and I will report back with my findings is a book called AI Snake Oil. It's by Arvind Narayanan, I hope I'm saying that right, and Sayash Kapoor, I believe both from Princeton. I had the pleasure of meeting Arvind actually at ODSC when we last hung out. And he gave a keynote that was just basically the 30-minute version of my workshop, less code and more just direct knowledge, and he was actually really instrumental in benchmarks like SWE and working on things like that.

01:17:35 The book is really exciting for me because I've always been, call it ranting about the gap of marketing and functionality. I mean, I am on record really laying into IBM Watson 10 years ago and just the marketing mishaps

that they had and how they weren't living up to a lot of expectations. I stand by everything I said. But that the whole concept of snake oil and AI is not new, but it is just explosive right now. So I am really excited to hear and read about what are the modern takes on snake oil? Because it used to just be, big company makes big claims, Google says they can call your hairdresser and make an appointment in 2017. Do we believe them? No. Now do we believe them? Yes. So things can change quickly. So what is the new snake oil that people are selling? That's what I'm really excited to dig into.

Jon Krohn: 01:18:29 Fantastic. Thanks, Sinan. A great recommendation. And yeah, that author Arvind, right?

Sinan Ozdemir: 01:18:36 Yes.

Jon Krohn: 01:18:37 He was highly recommended by Seamus McGovern who runs ODSC East. He said that I got to get him on the show, so maybe we will have him on the show for an AI Snake Oil episode soon.

01:18:48 In the meantime, for people who want to be hearing more from you, Sinan, we know about your books, so Quick Start Guide to LLMs, for example, third edition is now out.

Sinan Ozdemir: 01:18:58 Third edition is coming out. Second edition is out.

Jon Krohn: 01:18:59 Third edition is coming out. Gotcha, gotcha. Your O'Reilly trainings, people can find you there about once a week for classes like Transformer Architectures, AI Agents A to Z. A to Z in America. And RAG. Of course, you have agent classes in O'Reilly as well that I don't know if we've spoken about, but of course you do.

Sinan Ozdemir: 01:19:20 Yeah. Agent RAG courses also in July in the next week or two.

Jon Krohn: 01:19:24 And then Practically Intelligent, your podcast, maybe our listeners maybe won't be too long before they hear me as a guest on that show.

Sinan Ozdemir: 01:19:32 I was going to say I didn't want to ruin the surprise. But yes, you can also hear Jon on my own show when it comes out.

Jon Krohn: 01:19:38 Nice. Yeah, so that will be interesting if are, there's probably some dedicated listeners out there that are listening to most episodes of this show, and maybe they actually don't know very much about me at all except for my quips [inaudible]-

Sinan Ozdemir: 01:19:51 How often are you a guest on a show? Because you're always the host, you're always interviewing. How often do you get to be the guest?

Jon Krohn: 01:19:57 Yeah, I mean, I've gone through phases. So when my book, Deep Learning Illustrated, came out in 2019 I did a podcast tour where I was actively reaching out to be on shows. And in fact, that's how I ended up becoming the host of SuperDataScience because Kirill asked me to be a guest on his show, on this podcast, on the SuperDataScience podcast, and so I can actually probably look that episode up. I think it might be 365. I have that number in my head because it's an easy number to remember. Let me double check. Yeah, 365. All year round. And I asked him at that time basically the same question you just asked me, which was, how often are you a guest on other people's podcasts? And he said, never. He said he'd done it one time.

01:20:48 And I said, "Well, I've actually just launched my own little podcast,: which was supposed to be, it was called The Artificial Neural Network News Network. And so it was supposed to be a weekly news show about AI news. And the thing that was fun about it, we got to film one episode

in February 2020, and you can find, people can find this online on all the major podcasting apps as well as on YouTube. It's called A4N, the Artificial Neural Network News Network. And the first episode was in February 2020. And it was my vision, so it was me and four other data scientists, and I was like the anchor of this news show, and I'd say, "All right, let's go over to Andrew for sports." And he would talk about cheating and Kaggle. "And let's go over to Vince for weather," and he'd talk about AI being used to tackle climate change. And we had such a laugh recording it. And then the next week, the pandemic hit.

Sinan Ozdemir: 01:21:47 I was about to say something else was about to happen in February 2020.

Jon Krohn: 01:21:50 Exactly. And then, so nobody wanted to come into the office. We weren't able to keep doing it that way. So I did four more episodes where I interviewed guests, and Kirill was one of those four. And then six months after that, he was like, "Do you want to host the SuperDataScience podcast?"

Sinan Ozdemir: 01:22:12 Wow. That's crazy. I had no idea. That's cool.

Jon Krohn: 01:22:17 Yeah, and what's really funny is if you listen, people listen to Kirill's episode, we made fake ads for that episode.

Sinan Ozdemir: 01:22:27 About what?

Jon Krohn: 01:22:29 So it was an app for finding toilet paper.

Sinan Ozdemir: 01:22:32 Nice.

Jon Krohn: 01:22:34 Because it was the pandemic. It was like the pandemic had just started. I can't remember. We had a silly name for it. We had music. And we just recorded it in one shot

with Kirill, with the guest there. I was just like, "Okay, I need to take a break here to record this fake ad."

Sinan Ozdemir: 01:22:52 That's so funny.

Jon Krohn: 01:22:53 Yeah. So yeah, I was like, I didn't know, but I was auditioning for this podcast where we actually have real ads. I'm very much looking forward to being on Practically Intelligent and having a conversation and also meeting ... Remind me of your ...

Sinan Ozdemir: 01:23:05 Akshay.

Jon Krohn: 01:23:05 Akshay. Looking forward to meeting him, and I'm sure we'll have a lot of fun. Yeah, so I'll be, yeah, if people will follow me on LinkedIn or whatever. I'll be posting about that when the Practically Intelligent episode comes out. And on that note, how should people be, what's the social media place to follow you?

Sinan Ozdemir: 01:23:23 For me, it's LinkedIn. It was always funny to think about. Now, I never really thought I'd be that guy on LinkedIn. But for me, LinkedIn has been my social media of choice. I grew tired of Twitter/X. And that's where most of my followers can find my newsletters and my different blog articles and just everything that I'm doing. Yeah.

Jon Krohn: 01:23:44 Nice. Yeah, same for me. Yeah, I don't know. I got tired of, there's just so much more interaction on LinkedIn lately than X. And so yeah, at least for what we do in data science.

Sinan Ozdemir: 01:24:00 For sure, yeah.

Jon Krohn: 01:24:01 It seems to be the place to be now. Yeah, nice. Thank you so much for taking the time. And I'm sure it won't be long before you're on an episode again.

Sinan Ozdemir: 01:24:09 Well, thank you, Jon. It's always a pleasure.

Jon Krohn: 01:24:16 Always great to have Sinan Ozdemir on the show. In today's episode he covered how current AI benchmarks suffer from teaching to test where labs optimize for high scores rather than real world performance, as well as contamination issues where test questions leak into training data.

01:24:29 He talked about allegations emerging that Meta had to publicly deny manipulating Llama 4 benchmark scores highlighting how the lack of transparency in training data makes it impossible to verify claims.

01:24:40 He talked about how even advanced reasoning models, like OpenAI's o3 hallucinate up to 40% of the time on basic factual benchmarks like Simple QA, demonstrating that high capability scores don't guarantee truthfulness.

01:24:53 And he talked about how organizations should create custom test sets specific to their use cases, implement rubric-based evaluation with LLMs as judges, after validating against human evaluators ideally. And how they should chase their own internal leaderboards rather than generic benchmarks that don't reflect the enterprise's actual needs.

01:25:13 As always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Sinan's social media profiles, as well as my own at superdatascience.com/903.

01:25:25 Thanks to everyone on the SuperDataScience Podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, Nathan Daly and Natalie Ziajski on partnerships, our researcher, Serg Masís, writer Dr. Zara Karshay, and yes, of course our founder, Kirill Eremenko. Thanks to all of them for producing another excellent episode. For us today for enabling that super team to

create this free podcast for you, we are deeply grateful to our sponsors. You, yes you can support this show by checking out our sponsors links, which are in the show notes. And if you yourself are interested in sponsoring an episode, you can find out how to do that by going to jonkrohn.com/podcast.

01:25:59 Otherwise share, review, subscribe, edit videos into shorts if you want to. But most importantly, just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on or rocking out there. And I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.