SUPER DATASCIENCE
MAKING THE COMPLEX SIMPLE

SDS PODCAST EPISODE 707: VICUÑA, GORILLA, CHATBOT ARENA AND SOCIALLY BENEFICIAL LLMS, WITH PROF. JOEY GONZALEZ

Jon Krohn: 00:00:00 This is episode number 707 with Joey Gonzalez, Associate Professor at Berkeley and Co-Founder of Aqueduct. Today's episode is brought to you by the AWS Insiders podcast, by Modelbit for deploying models in seconds, and by Grafbase, the unified data layer.

00:00:22 Welcome to the SuperDataScience podcast, the most listened-to podcast in the data science industry. Each week, we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today. And now let's make the complex simple.

00:00:53 Welcome back to the SuperDataScience podcast. Today we've got the fast-talking, extremely knowledgeable, and extremely innovative professor and entrepreneur, Dr. Joey Gonzalez. Joey is an associate professor of electrical engineering and computer science at Berkeley. He co-directs the Berkeley RISE Lab, which studies Real-time, Intelligent, Secure, and Explainable systems. He previously co-founded Turi, which was acquired by Apple for $200 million, and more recently, he founded Aqueduct. His research is integral to major software systems, including Apache Spark, Ray (for scaling Python ML), GraphLab (a high-level interface for distributed ML), and Clipper (low-latency machine learning serving). His papers published in top ML journals have been cited over 24,000 times. Today's episode will probably appeal primarily to hands-on data science practitioners, but we made an effort to break down technical terms so that anyone who's interested in staying on top of the latest in open-source generative AI can enjoy the episode.

00:01:52 In this episode, professor Gonzalez details how his headline/grabbing LLM Vicuña came to be and how it arose as one of the leading open-source alternatives to ChatGPT. He talks about how his Chatbot Arena became the leading proving ground for commercial and open-source LLMs alike. How his Gorilla Project enables open-source LLMs to call APIs, making it an open-source alternative to ChatGPT's powerful plugin functionality. He talks about the race for longer LLM context windows, how both proprietary and open-source LLMs will thrive alongside each other in the coming years. And he provides his vision for how AI will have a massive positive societal impact over the coming decades. All right, you're ready for this phenomenal episode. Let's go.

00:02:40 Joey, welcome to the SuperDataScience podcast. It's awesome to have you here. Where are you calling in from?

Joey Gonzalez: 00:02:45 I'm calling in from Berkeley.

Jon Krohn: 00:02:47 Nice. And so we know each other through Raluca Ada Popa, whose episode was number 701, and that was an extraordinary episode. At the end of it, I asked her if she had any recommendations that people to speak to, and she said, Joey Gonzalez. And I already knew who you were from your amazing work on Vicuña, and so I was delighted and amazing to have you here. Let's start right away with Vicuña. So for our listeners who aren't aware of it, it is a model that I've been talking about on air for a while. In fact, we had an episode dedicated to these kinds of open-source single GPU ChatGPT-like models. So in that episode we talked about Alpaca, Vicuña, GPT4All-j, Dolly 2.0. That was back in episode number 672. And yeah, so Vicuña I guess I could try to introduce it again, but you might as well do it. And you can also tell us how it all came about.

Joey Gonzalez: 00:03:44 Yeah. Vicuña is a fun story. So again, I want to thank you for having me. It's exciting to be on, on the podcast. The story of Vicuña actually began over a break period sort of after another project Alpaca. So maybe I should go way back to the beginning of this year. With the release of LLaMA, the LLaMA model developed by Meta is a core foundation model. It embodies a lot of knowledge, but it doesn't really speak, doesn't chat. And so some colleagues actually led by my former advisor, Carlos Guestrin, when I was a grad student now at Stanford led a project called Alpaca. And the idea was to use a self-instructor, a method to train the LLaMA model to behave more like a chatbot using something like ChatGPT as a guidance mechanism.

00:04:24 And so they built this dataset, which is pretty clever actually. And they created a nice fine-tuning script that they released to the world that allows someone to take that dataset they built, and fine-tune LLaMA to speak more like a person, to have a conversation, to follow instructions. My students at Berkeley were like, we could do better. And one of the things that's important to know in this entire kind of revolution of large language models is that data is critical to success. And so the students looking at this project said there's a better data set. There's this website called ShareGPT which is sort of actually a demo of some web technologies. But that, that website did something pretty neat. It allowed people to have fun conversations on ChatGPT and then share those with their friends. These are the conversations they thought were funny, insightful, amazing, hilarious, I don't know. But important, these are the conversations they wanted to share.

00:05:20 And so these are high quality conversations. We downloaded the, I think, 800 megabytes of data using the public APIs for the ShareGPT website. And then the students took that data, the Alpaca training scripts, and they basically put the two together. There's a little bit of work the students did. This will blow your mind. They removed the HTML tags from the data. They did a little bit of additional cleaning and they fed that data into, again, the Alpaca training scripts and fine tune the model. And out came Vicuña. And this was done, I think, over break [inaudible 00:05:55] is spring break. Yeah, early. No, maybe it was early. Yeah, early spring break. This is done, you know, on vacation. It's kind of a hack over a few days. And they got a model that they were excited about.

00:06:05 And then we need to figure out if it's any good. So, the Stanford team invested and actually had some benchmarks run against. We wanted to use, you know, standard benchmarks like MMLU. There's a lot of benchmark in the NLP community for evaluating these models. Unfortunately, none of them are very good. They're not good because they don't measure kind of chat behavior, complex creative settings or more oriented retrieval, answering simple facts. So, to really assess a chat bot, we needed something stronger. And so students had this clever idea, why don't we just ask GPT? so they created a set of basic questions. They asked the model to answer those questions. We also asked Alpaca to answer those questions and GPT-3.5. And then we asked GPT-4 whose answer is better and score them, and then score them on various metrics. And in the process of doing that, we found out that our model was actually better than Alpaca (go Berkeley) and actually pretty close to GPT-3.5. which is really pretty exciting.

00:07:03 And so we posted this online and there was a lot of interest in this model. You know, in a a few weeks period, we went from LLaMA to Alpaca to Vicuña each making big strides in performance as assessed by the benchmark that we had created. So, Alpaca was certainly better than LLaMA, and, and Vicuña was better than Alpaca. And so, yeah, this generated a ton of interest. There's a blog going around about a discussion inside of Google. Some people in Google were also pretty concerned about this because we also compared against PaLM, or I guess Bard at that point. And it was comparable to Bard. And in fact, it was a little better than Bard in some of the other benchmarks we started running. So, it was a big step forward for these, you know, open-source research models. Something we were pretty excited about.

Jon Krohn: 00:07:51 That was the, "we have no moat" memo that went around Google, right?

Joey Gonzalez: 00:07:55 Yeah. So the, "we have no moat" story. And I will say in in defense of Google, they do have a moat as, as illustrated by Vicuña. It is about the data and using your data, using it intelligently can make all the difference. So, building a big model is important. We took off the shelf big models. We didn't pre-train our model and we did fine-tuning, again, it's fairly cheap. But we did it on really good data. And so I think maybe the punchline that if you were to take one thing away from, from this conversation, you know, in the beginning at least is that data matters. And we saw that with ShareGPT and Vicuña.

Jon Krohn: 00:08:27 For sure, and Google certainly has some of the best data. I love the way that you evaluated with GPT-4. So this is something that we use at my company, Nebula now internally for evaluating our models as well. It is brilliant because it means that you don't need to, like, we were always trying to figure out like, okay, we have this complex task. It's difficult to evaluate. Like, okay, we could do like semantic similarity, you know, take, convert the response of our model compared to some benchmark model and compare the embeddings. like take a cosent similarity score or something. And to try, to try to say, okay, well, the semantic meaning is similar, therefore maybe our model is doing all right. This idea of asking GPT-4, which is better and rated on a score out of 10, we now do that internally inspired exactly by you. Inspired by the Vicuña project.

00:09:15 Because it means not only can we now say, okay, we are definitely on this fine-tuned model that we've created for this specific task. Not only is it definitely better than the open-source LLM out of the box that we started with, not only is it comparable to say GPT-4, but on top of all that we can compare epoch over epoch or whatever kind of interval you want to evaluate on as your model's training on held that evaluation data. How, how is our model? Like, are we starting to overfit? is it continuing to improve? Is it improving at all? And so, thank you so much for this idea, which is like, so simple in a way, but so widely useful.

Joey Gonzalez: 00:09:52 Yeah. Well I should fill you in on some details. There's some following work that we discovered in this story. Good news is it was ultimately a good idea, but there's some caveats. So, I should say, once we launched this benchmark we, my first response is we should probably check to see if this actually compares to a human judge. And we didn't have a lot of budget for doing human evaluation at that point. And so we instead decided to try to run an arena. And so we wanted to build a website where people would actually, in the wild have conversations with the bots. The hope here is that we'd get not just our prompted discussion points, but what humans would say, what humans would ask, the crazy stuff that people might come up with. And then let them judge which models perform better. And that is why we launched the Chatbot Arena.

Jon Krohn: 00:10:37 Nice. Yeah. Let's talk about that more.

Joey Gonzalez: 00:10:39 Yeah. So the Chatbot Arena was a fun project. It started actually as like, how do we make this a game to get people to participate, make it fun? We took the Vicuña model, several of the other open-source models that had emerged at that point in time. We actually made API calls to the commercial vendors to get, you know, the state of the art models as well. And then, yeah, we put a website together where anyone can go. It's still there. If you go to arena.lmsys.org right now, you can chat with any one of the bots. You can chat with them directly in a setting where you know, which bot you're chatting with, or, and this is the more fun part. You chat with them blinded. So you chat with a pair of bots, you don't know which ones you're chatting with. You start a conversation, and both bots respond to you, and you continue that conversation speaking to both bots at once.

00:11:20 And at any point you can say, you know, A is better, or B is better, or tie, or they're both terrible. And we take that signal and we use that to create a ranking. And so we've ranked all the bots. We ended up using a ranking system called the Elo Ranking System, which was developed for the chess community. Has been adopted by the gamer community, has been, you know, incorporated in sports betting. It's a really cool mechanism. And that gives us an overall ordering of the [inaudible 00:1147] of the bots. And maybe not surprisingly, [inaudible 00:11:52] top that ordering are things like GPT-4, Claude is right there behind them. And then as we go down, we see GPT-3.5, and then Vicuña stays at the top.

00:12:02 In the beginning we were worried, why is Vicuña so good? No one's gonna believe our leaderboard if Vicuña's up there. Maybe we should run a little bit longer. Let's get Koala. One of the other models developed at Berkeley. I should say Koala was developed at Berkeley at the same time in collaboration with the Vicuña team. Koala was a little bit below Vicuña. Rolling back to the story. It's kind of funny. Koala was lower than Vicuña because they didn't remove HTML tags. That's the best of our knowledge. So, just a little bit of data cleaning. Again, punchline, think about your data. So, we did some better data cleaning, Vicuña was a bit better. But, you know, if we look at the overall leaderboard, Koala's below Vicuña, and then since then, a bunch of other LLMs have kind of merged in between I think we have like maybe over 20 models now. There's a lot of models on the leaderboard.

Jon Krohn: 00:12:42 Yeah. There's, I have some stats on that. So you collected over 53,000 votes regarding 33,000 conversations for 22 models. And all of that was released. So at the time of recording, this is very fresh on July 20th you released that conversation dataset to the world so that people can take advantage of all of those tens of thousands of conversations across all those, almost two dozen models.

Joey Gonzalez: 00:13:07 Yep. Yeah. So we released the data. That was a nerve-wracking point in the research progress. You know, releasing data is something you should do with care. We had been hoping to release data a lot sooner. We removed PII because we wanted to release data that didn't have any PII. That took some work. And then we ended up deciding to release all the data, including the conversations that we would've not, or, you know, we wouldn't have continued in the actual bot in the arena itself. So, we, we have offensive content filters. We actually kept the offensive content and registered the filters as well with the hope that, you know, the research community can start to study how these bots respond to offensive content when it's present.

00:13:44 Yeah. So that was a big release. My hope is that we'll help shape research in, you know, RLHF and the design of valuation functions to, or value functions to you know, train models in the future. So, yeah, it's a, one of the, you know, as a, as an academic at Berkeley, one of the exciting things that we get to do is focus on just general impact in building data, building models that will hopefully shape research in the future. Even when we look at Vicuña, you know, I frame it as a battle with our colleagues at Stanford, but realistically we looked at it as a chance actually to test some of the training tools we've been developing. We have some projects to enable sky computing. We have some projects to enable distributed training, distributed serving.

00:14:23 And so Vicuña was a very natural kind of extension of how do we test those tools. It was actually led by the system students who were developing those tools that kind of picked up that effort. And so, you know, for, for research, it's helped shape, you know, a lot of what we're doing, and now more efficient serving technologies, better use of GPUs for, you know, statistical multiplexing. All that was kind of driven by the work with Vicuña. And in fact, even the FastCchat arena, this, you know, the place where people can chat with our bots gives us a mechanism to evaluate you know, the underlying systems and how they can serve these models. Yeah. So it's, it's been a big research effort. And the release of data sets is, you know, one of the important steps in that effort.

Jon Krohn: 00:15:51 This episode is supported by the AWS Insiders podcast: a fast-paced, entertaining and insightful look behind the scenes of cloud computing, particularly Amazon Web Services. I checked out the AWS Insiders show myself, and enjoyed the animated interactions between seasoned AWS expert Rahul (he’s managed over 45,000 AWS instances in his career) and his counterpart Hilary, a charismatic journalist-turned-entrepreneur. Their episodes highlight the stories of challenges, breakthroughs, and cloud computing's vast potential that are shared by their remarkable guests, resulting in both a captivating and informative experience. To check them out yourself, Search for AWS Insiders in your podcast player. We’ll also include a link in the show notes. My thanks to AWS Insiders for their support.

00:15:55 Yeah. And we're gonna get back to talking about open-source versus closed-source with respect to both models, model weights, model architectures, data sources. We'll get to all that shortly. Before we move away from this

Chatbot Arena. They're, these kinds of, these models are being released all the time. These new LLMs are creating, you know, whenever somebody releases their big new LLM, they're like, look at these benchmarks. Were definitely the best. So, I don't know, kind of like, in my mind, I felt like up, up until this time of recording for a month or so, it was like this Falcon 40 billion parameter model that was the kind of model in my mind that I was like, this seems to be kind of generally the leader. At the time of recording, Llama 2 came out a week ago.

00:16:40 And in the Llama 2 paper, as well as on the main webpage on Meta's website, they have 11 benchmarks. And you already mentioned the top benchmark up, like the first one that they listed, MMLU, because this is one of the benchmarks that you hear about the most with natural language generation tasks. And, yeah, I mean, so to what extent should we, as somebody reviewing this table, when I'm thinking to myself, okay, Meta published this, should I, should I, I guess I should probably trust, like they're a big organization, I should probably trust like the numbers that they put out. But also, but I wonder when I, whenever I see these, I wonder what tests they're holding back. Like, how many evaluations did they carry out? And it is 11, just a subset of all the ones that they did. And they're now showing to us the ones that have the best account results.

00:17:29 Because when you look at this table, you're like, wow, the 13 billion parameter Llama 2 is performing comparably to that Falcon model that's 40 billion. And the 70 billion parameter Llama 2 seems to be absolutely, for the most part across these 11, it's crushing the results. Like, it's like, it's setting completely new kinds of standards for open-source LLMs. So, yeah. I mean, to what extent do you trust these results when you see them? Or, or do you think this is a scenario where you're like, put the, put it in the Chatbot Arena and that's the best place to evaluate?

Joey Gonzalez: 00:18:02 So, one, I was really excited about the Llama release. The Llama 2 release was, it's a, it's a big deal. And they put a lot of effort into building a better model, into evaluating the model. The paper is, is well written, like I was excited about this release. We immediately put it in the Chatbot Arena and have started to get some signal on it. And the scores are not amazing, which is sort of disappointing. I was expecting Llama to do better. It's still preliminary. We need to get more data. The Elo ranking system is not particularly robust, so it takes some time to get a good estimate of what its ranking would be, but it's not in fact, the large instruction to inversions aren't as amazing as I had hoped.

00:18:37 We actually did release a benchmark. So, one of the fun kind of anecdotes of this whole journey is we, we, when we released Vicuña, we used this GPT-4 idea. A lot of other people have picked that up. We went and collected some data with the Chatbot Arena, which was pretty consistent with our original GPT-4 results. But we then started digging into the GPT-4 study that we did. And it was fun to discover that GPT-4 has some very peculiar biases. And so how you do that GPT-4 experiment, take some care and we've since fixed that. So, we have this MT bench, this multi-term benchmark. And I bring this up because when, when Llama did the, when Meta did the evaluation of Llama 2 they didn't look at these more kind of complex discussion oriented scenarios. And we wanted, with the MT benchmark to really test that, that setup.

00:19:26 And so we created a battery of 80 questions based on the kinds of conversations people are having on the Chatbot Arena. But we then made these multiple rounds of questions, follow up questions, and then we had grad students carefully assess the pairs of models. And then we also had GPT-4 assess the pairs of models. And so on that benchmark, which we've published results, again, the Llama 2 models are not as good. As, as we had hoped, actually, I was, I was very excited about Llama 2. Now, I imagine now that they're kind of releasing and making more of a public commitment that'll improve with time.

00:20:02 There is a caveat that I need to raise, and this is an important one and it comes back to using GPT-4. So, if you are using GPT-4 to evaluate things, GPT-4 has two or three important biases. First, GPT-4 prefers whatever it read first. It does not, so there's a bias to the order in which you present things. When we ran our experiments, we put GPT-4 or we put our competitors first in the original Vicuña papers. And our value, our response is second. Had we reversed it, we would've found that we were better than GPT-3.5. Now that's a bias. And, you know, if you randomly sample it becomes much more neutral. So, it's important to deal with the ordering bias.

00:20:45 The next bias is kind of funny. Also, humans share the same bias. GPT-4 prefers longer responses. And we've noticed this across all the LLMs, that they keep making their responses longer because humans tend to prefer more words in their response, which is a little surprising. I, you know, I guess I speak a lot, so maybe, you know, I, I'm used to saying too much, but I, I'd have to imagine if you're asking questions, you want a short answer. And in many cases you do. But when you judge these models and you use something like GPT-4 as a judge, it prefers longer responses. And then finally, GPT-4 prefers itself in general as a self-preference bias, which humans also do.

Jon Krohn: 00:21:26 And I guess is that even when you, presumably something obvious that you're already doing when you say that Is you're, you're blinding that it is GPT-4.

Joey Gonzalez: 00:21:33 That's correct.

Jon Krohn: 00:21:33 Like, you're not saying this is you and this, and, but, so, but it prefers itself-

Joey Gonzalez: 00:21:38 It prefers its own writing.

Jon Krohn: 00:21:39 Yeah. Yeah. So it'd be like, almost like, you know, if you heard somebody speaking in the same accent as you might think, yeah. Right, right, right.

Joey Gonzalez: 00:21:45 Yeah. So it's like, when I go back and read my own paper, I'm like, oh, that was pretty good writing. I liked it. But I read someone else's paper. That's pretty much the same thing. No, I don't like this. And so that, that stylistic preference is important. It's important because a lot of these open-source models are using things like GPT-4 as or using, you know, data that was generated from GPT-4 conversations, like ShareGPT. And so that makes their, their behavior or their style of speaking closer to something like GPT-4. And, and Llama 2, I don't believe did that. So it's possible that, you know, one of the reasons that we see this, this kind of degradation Llama 2 numbers could be something related to that.

Jon Krohn: 00:22:23 Yeah. Except, but you said it's in the Chatbot Arena as well, although early stages. But the Chatbot Arena, that's human, that's human evals, right?

Joey Gonzalez: 00:22:29 Yeah. And it also gives a lower score. I, let me dive into that too. So we've started to look at why some of the really good, or at least inn principle should be good models aren't doing as well. And this is maybe the fourth interesting takeaway. Models like Palm and Llama or, and Llama 2 refuse to answer things. You ask it tough questions, questions that shouldn't answer, and they'll go, I don't know or I don't I don't want to have an, I don't have an opinion on that, or I won't explain how to build that. And so that, that abstention behavior, which is actually something that's why like this, maybe the focus of the Llama 2 paper is something that they did right that actually causes human interaction scores to go down and we also saw this with Palm. And a lot of cases, Palm will actually lose to like a really bad model like Dolly. And he'll lose because Dolly will, I'll answer any question you ask. It doesn't have to be the correct answer. I'll say something and Palm will go, no, I don't know the answer. I won't. I, you know, that's not a question I have an opinion on. And so this kind of abstention behavior, which is again, something that we actually should be aiming more towards these benchmarks don't pick up.

Jon Krohn: 00:23:39 That is very interesting indeed. So it goes to show that, you know, even when you think you've developed this kind of foolproof seeming approach where you're like, we got people in the arena, we got human evaluations. This is like the most expensive and valuable way to be doing this assessment. And even then you're running into this issue that, yeah, some of these models that are really good, that are designed to prevent misuse, can be getting down rated because people are like, "Ah, this is a terrible answer. You won't even tell me how to build a bomb."

Joey Gonzalez: 00:24:07 Yep, yep. Yeah. And so, it, it's a neat, it's a neat observation that we've had. It's something that when we look at the arena as we, you know, aim to the future, thinking more about how to incorporate that as kind of a goal you know, maybe having situations where you can say, you know, the guiding the user that, you know, would it be more appropriate for the model to abstain you know, consider that in your evaluation. And also pushing the arena in more kind of vertical orientation. So like, you know, looking at code, you know, ask me code questions for this specific part of the arena, so we had, can kind of isolate some of these abstention behaviors.

Jon Krohn: 00:24:41 You mentioned to me before we started recording that the Elo rating system that you decided on for Chatbot Arena is also not that, maybe [crosstalk 00:24:48]

Joey Gonzalez: 00:24:48 Not great. Yeah. So let's talk about Elo. Elo is a pretty cool rating system. It has sort of a few design principles. So it was created for chess, it was created for people who are playing in a decentralized fashion. People play chess all over the world all the time. And, you know, I can play you and I can win and you can win. And we need a way to update our scores. And so there are two goals. One is that I need a decentralized way to compute someone's scores, and Elo provides that. And then also I get better or worse one, you know, one of us can improve in time. And so Elo allows for that, that change in scores. In a Chatbot Arena, we have all the data in one place. And for the most part, we are freezing the model version. So the models don't change.

00:25:31 And so there are other methods that are related to Elo is a geeky tangent, but Elo has a kind of a close connection to logistic regression. And you can actually analytically solve for what would be the fixed point of an Elo score, and I believe it's called the Bradley-Terry model. So there's other ways to rate things, but they're not as cool. And so the, you know, Elo scores have stuck. They've stuck so much that, you know, I'm starting to see clones of our benchmark with Elo scores in other places, too. Something you should know, if you're reading an Elo score, if everyone's close to a thousand, don't trust the Elo scores. So, the rankings don't mean much. It takes time for these models to diverge. There's parameters that, that are needed to tune to make sure that Elo scores, you know, you know, give some separation.

Jon Krohn: 00:26:14 Gotcha. Gotcha, gotcha. So it's, the Elo ratings are ideally suited to an application like chess where those kinds of, like hyper-parameters have been figured out over time.

Joey Gonzalez: 00:26:23 Over time, yeah. And, and where people are changing their, their, their, you know, capabilities. And, and also, again, when you really need this to be done in a decentralized fashion where updates can be done, you know, you and I can play, and then we can recompute our scores without having to check with all the other chess players in the world.

Jon Krohn: 00:26:38 That's cool. But yeah, very interesting. Yeah. Limitation there to cover. One thing that must be flattering for you, or I don't even know if you think about this, but Llama 2, when they released it, it's, it's obvious to me that they did release now also a version that is fine-tuned to be great at chat. So, they released the pre-train model just like they did with the original model. But then these kinds of things that you were doing with Vicuña, this fine-tuning to human conversations, allowing it to be great at chat, that is something that it's like, it's super obvious that they did that as a part of Llama 2. If they hadn't done that, you'd really feel like they'd miss something. And so, I don't know, do you think about that? Or like, do you think, do you think this is something the space would've gone in this direction inevitably anyway? Or that you played like, you know, a key role?

Joey Gonzalez: 00:27:30 So Llama and in fact, the whole Meta participation in this space is, to me, it's really interesting. It's exciting. In fact in this Llama 2 release Berkeley is gonna start collaborating more closely with Meta on the kind of the LMM development, which I'm thrilled about.

Jon Krohn: 00:27:43 Yeah. They list BAIR as one of the, like, key partners.

Joey Gonzalez: 00:27:47 Yeah. So I super excited about it. I think they probably would've headed in the chat direction just because the whole open-source community kind of moved that way. Now, you know, did we start it? I don't know. Alpaca certainly did the first two fine-tuning. If we hadn't done it, I'm sure others would've at some point you know, took, taken an open-source foundation model and tried to run, you know, the instruct fine-tuning that that OpenAI, you know, described. And so I think it would've happened. I'm, I'm thrilled to see what, what Meta's doing, kind of their commitment to the open-source, to the, you know, developing these models in an ethical fashion, writing about how they did that. Building these foundation models is expensive. They, I think they said it 25 million in just training data alone. And I can't even fathom the amount of engineering and compute hours that went into that.

00:28:36 So, yeah, I, I'm excited about where they're headed. I was, again, a little surprised that it wasn't performing as well. But, you know, there are these caveats that I, that I already listed. There are things we need to think through as we start to evaluate these more in the future. And, and I think we're still actually at the beginning. Because, you know, these models, in fact, we're doing [inaudible 00:28:53] trying to be integrated into language or into visual reasoning tasks. We, we have work on, you know, program synthesis, all sorts of ways in which these models will be used. And there they don't need to chat well. They just need to have a good understanding of language and we can, you know, adjust them to these specific behaviors that we need.

Jon Krohn: 00:29:10 Nice. Yeah. Really cool. So, awesome to be able to hear the Vicuña story from one of the Vicuña developers. I didn't know any of that about kind of the genesis of the project. I just saw the timelines. I'm like, wow, this is happening really fast. Yeah. Alpaca, Vicuña. And do you, can you really quickly, I mean, so I know these are names of South American I guess, are they [inaudible 00:29:33] I don't know, like, they're kind of this class of animals, they're all related to the Llama. So when Llama came out and then Stanford came out with Alpaca, is there any story around you deciding to have it be the Vicuña specifically? There were some other options.

Joey Gonzalez: 00:29:45 I don't think so. We, we, we were coming up with names of this general species of animals. I think Alpaca is a nicer higher in the, in the fur world. It's a better fur. I think it's a, I don't know this. Yeah. Alpaca is a, is an interesting story. Vicuña, maybe a little bit less. So I was kind of disappointed that we spelled it wrong in our first releases of it. I think getting the Ñ on some keyboard and stuff. But yeah it is Vicuña and, you know, another one of these kind of Llama animals.

Jon Krohn: 00:30:15 There's also, I think people mispronounce it a lot. I love my favorite podcast to listen to myself is called Last Week in AI. And the hosts of that were calling it Vicuna, or no, how do they even, I can't even remember.

Joey Gonzalez: 00:30:26 Probably Vicuna. We, my, my students initially were saying Vicuna when we were kind of building it. So, it was, yeah, Vicuña is the correct, correct pronunciation.

Jon Krohn: 00:30:34 Deploying machine learning models into production doesn’t need to require hours of engineering effort or complex home-grown solutions. In fact, data scientists may now not need engineering help at all! With Modelbit, you deploy ML models into production with one line of code. Simply call modelbit.deploy() in your notebook and Modelbit will deploy your model, with all its dependencies, to production in as little as 10 seconds. Models can then be called as a REST endpoint in your product, or from your warehouse as a SQL function. Very cool. Try it for free today at modelbit.com, that’s M-O-D-E-L-B-I-T.com

00:31:15 And there's a funny, there's, so I asked a week before we're recording today I asked listeners by my LinkedIn page as well as my Twitter page, whether they had questions for you. And one of our listeners, Wes McDermott. He provided a YouTube link to a video from a film called Sunset Boulevard. I guess it's from around the fifties. And there's a funny line, I'll try to make sure to remember to include it in the show notes about, like, it's, he says it's his it's one of his favorite lines from one of his favorite movies. And it's this thing about, "Oh, you should have got her to buy the Vicuña."

Joey Gonzalez: 00:31:51 That's great. Yeah, yeah. Maybe's been a struggle for us actually in all these projects. We, we can yeah, we have a Vicuña I mean, the whole Llama series comes from LLM so. But we have a project Gorilla, which we can talk about a little bit too.

Jon Krohn: 00:32:10 Yeah. Yeah. That's exactly what I wanted to cover next. So let's move from, to a completely different part of the animal kingdom and talk about Gorilla. So the context here is that I've used, and probably a lot of our listeners have used ChatGPT plugins, which are a really cool way of interfacing with real-time information, in ways that are designed to be really smooth. And so you can, if you are a ChatGPT Plus subscriber, you can go to the settings and you can choose to be involved in the beta for these plugins. And then you get this like plugin store, and so you can choose Mathematica from the plugin store. And then when you provide some kind of math problem or equation-related problem to ChatGPT it should, instead of trying to use next token prediction, which is not a mathematically sound way to be making predictions, though, is unbelievably well and, and surprising. It does unbelievably well in surprising number of circumstances. Mathematica, which is a language designed for doing math, should be better at solving that problem. So the ChatGPT plugin should recognize, oh, here's some math, Mathematica would be better for that. And there's lots of different kinds of applications out there, like realtime web search or even very specific searches, like Kayak is one of the most popular plugins. So for booking a car or hotel room or whatever, you could go into ChatGPT and say, I would like to go to Los Angeles and can you rent me a car? And it'll come back with some suggestions. All right in there, in the chat interface. So that's all really cool, but it's not open-source. And so yeah. You and your colleagues at Berkeley, as well as I guess Microsoft Research have been working on an open-source kind of variant of this.

Joey Gonzalez: 00:33:56 Yeah, so it was the Gorilla Project named because gorillas used tools which was simple. I've back-named it to say LLA, Large Language APIs. And I, I, you know, it's a refinement of the original intention of the name basically gets this idea of, you know, how can an LLM interact with web services, with technologies outside of itself to gain knowledge and to affect the world. And I think this is where a lot of this technology will head that these, these AIs will make it so that we interface not with the browser, but you know, through text or through voice with an AI that interfaces with the web that, that can find and, and use services to achieve tasks. And this is kind of the bigger vision of the Gorilla Project. Gorilla started I guess in the early, early winter as a, you know, early, yeah.

00:35:00 So Gorilla started as a discussion even before kind of Vicuña was taking off. And then as Vicuña took off, we certainly pushing more in the kind of the, you know, having better open-source models. We were doing, doing more stuff. I think with the Gorilla Project today it's, it's become an open-source effort to target a wide range of different APIs. The way it works is you ask Gorilla, you know, I want to do in fact, you have Gorilla for the terminal. So you can go in your terminal and you can install the Gorilla Command line tools and say, "I want to list all my files in order of, you know, size and followed by date." And it'll tell you what commands to run to do that. And the way this works, and I think this is kind of the exciting part of Gorilla that maybe is a little different than what even OpenAI is doing, is we combine retrieval, augmented generation, it's called RAG with fine-tuning.

00:35:48 And the reason to do this is to make it so that the model can discover APIs. So we should be able to add new APIs by a documentation to the model. And then we fine-tuned it to be able to read these APIs. To be more effective at reading these docs and then generating results in a matter that, you know, is consistent with the request. Remarkably fine-tuning is pretty critical. And one of the surprising findings for me in this work is that fine-tuning on the APIs goes a very long way. And that retrieval helps a little bit, a little bit more. But, but fine-tuning the model to understand the API seems to be pretty critical. And this creates problems for the entire field. If the future is to be fine-tuning models on your data that means we're gonna have a lot of very expensive-to-run models to be able to do a lot of things.

00:36:38 We should come back to, you know, what that means for kind of research and for industry. But for the Gorilla Project it's meant that we've had to find a lot of resources to host these models. We try to make them open to the world. You can download the models yourselves, but we also host them in the cloud. And our hope as we push the project forward is to kind of further extend this idea of incorporating retrieval with fine-tuning to be able to support not just calling a single API, but I should be able to, you know, chat to my computer "I need to book flights for my upcoming conference," and it can go look at my calendar and figure out, oh yeah, I think this conference is this, it can figure out when I might want to be there. You know, there's a weekend, it's discounted, come back and say, "well, the cheapest, you know, flights for your, you know, trip to VLDB would be the following." And so I go, yeah, I like those. Can you book those flights and find hotels for me as well? And then that kind of interaction with the chatbot and then the chatbot taking action on the world is I think where we're all headed with a lot of this technology.

Jon Krohn: 00:37:33 Yeah. It's not that much of a stretch anymore to imagine. Well, well, so lemme back up a second. And, and so, a year ago I gave a TEDx talk where I went through the life of this woman named Jeanne Calment. She was a French woman. She's the oldest person to ever have lived according to like, documentation. And so she lived like 121 or 122 years. And so I follow her life from when she was born in the 1870s to when she died in the 1990s. And over that time span isn't like everything was invented, like, pretty much like, it's like from light bulbs to transistors. I could go over the long list, but it's wild the changes that happened in the 120 years that she was alive. And so in my TEDx talk, I was like, well, now try to project forward from today and think about, you know, baby born today, given medical advances, and now lifespan doubled, average lifespan in the west doubled in Jeanne Calment's lifetime. Maybe we're not gonna be able to double, but it seems safe to say that some child born around today is gonna live at least as long as her.

00:38:40 And so what kinds of changes will this child bear witness to? And even in Jeanne Calment's lifetime, like the change was so rapid that there's no way that Jeanne, when she was a kid, would've been thinking about cell phones. And the internet. And my argument that I make in the TED Talk is that because of AI in particular, and because we have more human brains than ever before that don't need to be doing physical labor, for the most part, there's all this human ingenuity combined with AI, things are going faster than ever, and that's going to increase. It's going to increase, increase and increase. And so it was recently the one year anniversary of me giving that talk. And so I reposted the talk and I said, when I was giving this talk, if you had asked me if we would ever in our lifetimes have something with the capabilities of GPT-4, I would've said, maybe.

Joey Gonzalez: 00:39:31 Yeah.

Jon Krohn: 00:39:32 And now we have it. And so it isn't a big stretch of the imagination at all. And in terms of technically, like, it's kind of just a matter of gluing pieces together. There's no reason today why I couldn't have all of my email inbox history, all of my historical calendar events be processed by some kind of LLM like this and, you know, there's some cleverness like you're saying, like getting the API things right. But it could absolutely do that. Everything you just described of, you know, based on my history of like, the kinds of flights that I tend to pick, you know, you're probably gonna want to get there two days before the conference, like you usually do, because, you know, and you mentioned three years ago why in an email, and it's got that right on queue. So, yeah, I don't, I can't remember where we're in the conversation, but.

Joey Gonzalez: 00:40:28 Well, so I think this kind of rapid progress in AI it's, it's taken a lot of us even those of us doing the research by surprise. You know a year ago in fact a year ago when I was working on my company, maybe we'll come back to, we were seeing a lot of people using SciKit-Learn and doing basic machine learning. And I was kind of, kind of sad actually, because we had done all this really cool deep learning stuff and built new systems to support it. But that was kind of, you know, a lot of SciKit-Learn and basic machine learning. And then, you know, fast forward to today, and everyone's like, actually, I think I want to run large language models. Deep learning is now mainstream enough that, you know, the basic things that I would do, I should be doing with deep learning today.

00:41:10 So we've moved fast in the technology. We've moved fast in the adoption of the technology. I've heard story like people's grandparents are using LLM to cheat on their book clubs. That's awesome. But that's kind of, that's a big shift that, you know, a technology that's you know, is kind of deep in research is now, you know, so mainstream that it's, you know, it's in, you know, in discussions around contract negotiations with unions, it's shaping how, how you know, how people cheat on their book clubs. This is a big shift in technology. And AI has, you know, in the past been a source of hype and a source of failure. I think we are at a point where the hype might have met reality or reality might even be exceeding the hype that we had. And that's exciting. It's a little bit scary too what it means for research, what it means for industry. It's harder and harder for me to know what tomorrow or what six months from now will look like.

Jon Krohn: 00:42:08 Yeah, for sure. So, yeah. So Gorilla, another step in this open sourcing this capability of having, I, you can support an, like an effectively unlimited number of different kinds of APIs with this, right? So, what's the, going back to kind of the nuts and bolts of this Retrieval Augmented Generation, RAG, what would happen if Gorilla only had that? So if Gorilla didn't have the fine-tuning to understand API and it just had RAG, what would that look like? What would we be missing?

Joey Gonzalez: 00:42:41 Yeah, so we did some studies of this. I was keenly interested in kind of what is the, you know, the best we could achieve with RAG? Using the Llama or Vicuña models, we didn't get very far. So we switched to Claude. Claude is actually remarkably good at long contexts. And we can stuff a lot of documentation and context. And, and we can get-

Jon Krohn: 00:43:01 Did you see, so just today at the time of recording, Anthropic announced that they have expanded the context window on Claude from 9,000ish tokens to a 100,000 tokens.

Joey Gonzalez: 00:43:13 Yeah. So yeah. This race for large context is exciting. We could talk about that to.

Jon Krohn: 00:43:19 Let's talk about that next.

Joey Gonzalez: 00:43:20 Yeah, yeah. But, but just looking at Claude as our, as our baseline, this is a pretty, you know, this is a good model with a long, long context support. I think we might have had beta access some of the earlier larger context APIs. We stuck a lot of text in. And we get pretty close to Gorilla with just fine-tuning. And so fine-tuning pushed Gorilla a long way. Now there's some caveats and, and I, you know, I do this, I want to, this is research. So we were looking at a specific class of benchmarks that are focused on calling Hugging Face and PyTorch APIs because it has lots of documentation, lots of usage. And so it's a smaller set of APIs, so we could perhaps be fine-tuning effectively to memorize large fractions APIs. Regardless that fine-tuning, again, even if it's memorizing the APIs, is making a big difference.

00:44:10 And that was something that I was, I was again, surprised about that, that even with Claude with good, you know, my students have become pretty good at prompt engineering. So, you know, kind of hacking the inputs still not enough to get to, you know, where we were with fine-tuning. So, yeah, it's an open discussion. I think we're, we're RAG where this Retrieval Augmented Generation and fine-tuning will come together. And I think, you know another point for me for the whole, whole podcast, I guess is I, one of the big questions I think of 2024 and or maybe the end of 2023, I can't see that far ahead, is kind of what is this balance of how we, how we will use RAG, how we'll essentially in context learning stuffing examples, relevant data, how we'll mix that with fine-tuning? And, you know, there are a lot of reasons from the system perspective to push for something like RAG for, you know, using in-context learning. But there also seems to be strong evidence that fine-tuning can take models a long way.

Jon Krohn: 00:45:08 Very cool. This episode is brought to you by Grafbase. Grafbase is the easiest way to unify, extend and cache all your data sources via a single GraphQL API deployed to the edge closest to your web and mobile users. Grafbase also makes it effortless to turn OpenAPI or MongoDB sources into GraphQL APIs. Not only that but the Grafbase command-line interface lets you build locally, and when deployed, each Git branch automatically creates a preview deployment API for easy testing and collaboration. That sure sounds great to me. Check Grafbase out yourself by signing up for a free account at grafbase.com, that’s g-r-a-f-b-a-s-e dot com.

00:45:53 I wanted to get a really brief pause before we move to the next topic to make sure that I've defined some of the terms that we've used in this episode. So I very quickly mentioned how BAIR is an affiliate of Llama 2. And so I just want to quickly say that that sounds for stands for Berkeley AI Research, I guess. So, it's like this, yeah, in the logo was a bear. So, it's just like, and, and-

Joey Gonzalez: 00:46:16 Go bears. Yep.

Jon Krohn: 00:46:18 Yeah, so yeah, it's cute in a number of different ways and that lab has been around forever.

Joey Gonzalez: 00:46:25 Yeah. The BAIR Lab I guess, when did it start? I want to say like 2015, 2016. It's been around for a while, but the group of people in BAIR, the team before we, we, they had a very, you know, clever naming activity, had been working together for, you know since I started my PhD actually, and perhaps before that. So, it's, it's it's weirdly been a powerhouse in AI. And today it is the very much a powerhouse in AI. It's, I think this is maybe an embarrassing fact, but as you rank like the papers at

NIRPS there's like Google Brain, maybe Microsoft, and then BAIR. In terms of kind of the overall number of publications, not necessarily a good metric of research but maybe also impact, like, you know, OpenAI, a lot of the core technologies that are being used that they write about were developed by students at BAIR. So, pretty outsize impact for research group.

Jon Krohn: 00:47:14 And I think we'll probably end up talking about this again as we talk about your commercial ventures that you've started, but there's this huge, Berkeley is amazing. We talked about this in Raluca's episode 701 a lot as well, where Berkeley is amazing for coming up with big challenges to tackle over many years of research, putting the right researchers together, tackling those, coming up with often open-source standards that become the standard solution to that problem globally. So, yeah, so we'll talk about that more in the context of your of, of the entrepreneurial stuff that you've done, Joey. Before we get there, the other acronym, I guess, although, this is maybe just an abbreviation, not an acronym like there is, but you've also mentioned LMSYS, so, the large model systems organization, so. Yeah, how does that fit into, like, that's also a Berkeley organization, right?

Joey Gonzalez: 00:48:04 Yeah. So LMSYS was created as we were launching Vicuña. We, we wanted an umbrella thing to support the research. We have some collaborators at Stanford at UCSD and other places, I think CMU as well, maybe that were involved in the kind of early creation of the research. And rather than tying that to a, you know, to a BAIR activity, which, you know, many of the students are in BAIR for that as well. Or the Sky Lab, which I also run you know, many of the students are involved in Sky. We wanted to create an entity that would sort of embody that, that research agenda that would, you know, be a little bit isolated from some of the specific labs that we have at

Berkeley. So, we created the LMSYS.org and put a lot of the work under that, that banner.

Jon Krohn: 00:48:47 Very cool. Yeah. So with that behind us, we can now move on to something that I'm sure it's LMSYS that is on top of which is these long context windows. So I just mentioned how Claude Anthropic's LLM now, you know, they've 10xed the context window and that doesn't, it says something that they've gone commercial with that it says probably something about the reliability, but, you know, that's something that is, that, that's difficult to access. You can probably speak about better than me, but, you know, we've had papers come out in recent weeks around models that can supposedly handle like millions of tokens. Like, or there's been papers that are literally like, it doesn't matter any length context window you want, it's fine. You can put all of the internet in, and it's like, well, obviously there's a degradation.

Joey Gonzalez: 00:49:33 Right. Yeah. So the race for long context is, it's an important one. It fits into this question of like, how do we balance in-context learning and fine-tuning? But I mean, fundamentally it's about how much pretext will my model read before it goes to answer my question? Or if I'm writing a really long essay, how long can my essay be, you know, if I'm trying to get a hundred thousand-word essay together for my class project I would want to use a model that can do that for me. There are tricks, and in fact, you know, the race to long context has kind of a parallel effort, which is how to use smaller contexts to address some of the challenges that, you know, long contexts try to address.

00:50:15 You know, as a human, you don't have, or I don't have a long memory so I take notes. Taking notes is a way for me to capture context when I've read, so I don't have to remember exactly what happened in the first half of the book. I can go look at my notes and I can look at notes that are relevant to what I'm reading now. So, this brings in kind of retrieval, how do I go back to my notes? Nonetheless, having a long memory of what's happened in the book when I look at this word, being able to have, remember the past a hundred thousand words could make a big difference in how I understand the meaning of this person. What's this person's story?

00:50:49 So there's been a big effort to deal with long context. The challenges are many. So, computationally context grows increased computation quadratically. It also increases the amount of memory required quadratically. I have students working on, on various aspects of those problems at Berkeley. And, you know, there, there are tricks that one can play. There's research on making the attention to that context sparse. So when I'm looking at this one person, do I really need to really look at all last 100,000 words? Or maybe I can look at, you know, just a few of the important sections. And so there's work on sparsification, in my group, thinking a lot about systems, you know, if I'm gonna use that full context, can I split it over GPUs in interesting ways?

00:51:31 Second big problem is you need training data with long context, because it's important to have that extra signal about how to use the full context. There are tricks for extending your training data. There are tricks for changing the way we, we embed the positions of text so that we can maybe get away with smaller amounts of training data and try to extend it to longer contexts. So, and we've been doing some work with that. In fact, the LMSYS blog has one of these recent tricks that someone else actually developed that we sort of implemented, tuned up, and fine-tuned our models to run against. So, you know, there, there's challenges around the data. And then there's this third issue, which is, so you can read a hundred thousand words, but do you remember what you read? And does it all, you know, do you look at it equally?

And one of my other students in collaboration with group at Stanford started looking at it like, turns out we don't actually care about what happens in the middle of that context, just the beginning, just the end. That's what, you know, say the art models seem to do. Which could be fine. But if I'm trying to summarize everything that happened in that context, and I was like, yeah, s**** it, middle's not that important. That could be a problem.

00:52:32 This also shows up if I'm doing retrieval. You know, maybe the answer to the question that I ask is somewhere in the middle. And this happens when, when I'm looking for lots of, you know, reading lots of notes maybe the answers in the middle of my notes again, I, I'm lost. So, fixing these problems is something that we're also thinking about at Berkeley. Part of it could be training models to be more sensitive to the middle by putting the answers to the questions more, you know, formally in the contexts. Yeah. It's, it's an area of interest for us. We've actually started collaboration with Anthropic to start to build benchmarks to evaluate these things more effectively. So, we can understand when you say 100,000 tokens at Anthropic, does that really mean you're using all those tokens equally? And how do you make use of that bigger context?

Jon Krohn: 00:53:17 Very cool. Yeah. Tons of things to tackle here. I think one of the things, as you were talking about catching things in the middle, I can't remember who was telling me this. It might've been a data scientist on my team. So, it could've been Grant Vet. He was describing that one of the things that is done to evaluate whether these work, is like, you can have like Easter eggs, like hidden at like random points in the, in the full context. And you can test very specifically on those.

Joey Gonzalez: 00:53:44 Yeah. Yeah. So that's what the benchmark that my student put together is they put a JSON keyword and, and value anywhere in the thing, and then they ask-

Jon Krohn: 00:53:52 Yeah, that's exactly what it was.

Joey Gonzalez: 00:53:53 Yeah. So how to, how to find that. It's, it's a, it's a good micro benchmark. It tests this very, can you find this, if I tell you exactly the thing that you're looking for, and you should just do direct attention, can you attend to the middle? And already not as good as one would like. What, what, where I'm getting to this bigger retrieval augmented generation, this RAG and fine-tuning story, one that I'm really deeply interested in is, you know, you've bought Pinecone, perhaps you have a big vector store. You've retrieved the top 1000 relevant pieces of documentation for this coding task. But it turns out the answer's somewhere in the middle. And the other documentation's not only wrong, it's perhaps distracting. It's similar APIs, but the wrong thing. Don't call that. How do you deal with that and how, how good is the model, like removing the things that are wrong from its attention and, and attending to the right stuff when that right stuff might be anywhere?

00:54:42 And so in that actual benchmark from Stanford, they tested this, and it's kind of neat to see that, you know, these retrieval methods, you know, using [inaudible 00:54:50] products get, you know, pretty good recall at a thousand documents. They most likely will cover the answer to your question, but the models don't get better. The claw, like state of the art models don't improve their performance when, when doing RAG. And that's a big deal, you know, if you're gonna pay for Pinecone to do this cool retrieval, you want to make sure that you get the results from the LLM at the end as well. So.

Jon Krohn: 00:55:14 Yep. Yep, yep. All great points. Very exciting space. One of the points that you mentioned as you were talking about this, is you were like, if I was a student that wanted to write a hundred thousand word essay, you know, there's maybe increasingly there are models that could do that. But un until recently they weren't. But it just kinda just kind of got me into thinking about this question. So, you do a lot of teaching, you teach, you devised the upper-level data science course at Berkeley, and there's over a thousand students a term in that course. So, how do you as an instructor, have you changed the way that you evaluate students in the last year?

Joey Gonzalez: 00:55:55 Tough question. So, so sadly we have not. It's something that's become top of mind at Berkeley to think about the impact of LLMs in how we teach and how students learn the potential opportunities and the downsides. Currently we haven't vastly changed our curriculum. So, I teach both now the Data 8 class, which is the intro to data science class, which has almost 2000 students a semester. And then I teach Data 100, which is our next level class that brings in Pandas, SciKit-Learn, and a little bit of [inaudible 00:56:31], all this, you know, more advanced stuff. Again, you know, a couple, yeah, nearly a 2000 students semester. So, big classes. One of the things that we've found in teaching big classes is that the interaction with TAs providing guidance is critical to learning. And students that, that get that support, that, you know, I'm writing something explain, oh, yeah, you had a bug there, think that one through again can make a big difference.

00:56:54 And having that feedback in the thought process itself shapes our learning abilities. And there's actually an effort now at Berkeley to bring Vicuña and some of the commercial models in to see if they can be used to help guide students as they're doing exercises. So, that's a plus side. There's a chance that LLMs can help provide the additional, you know, immediate feedback as you're doing something like writing a program or maybe solving a math problem or maybe someday even writing your English paper, that will guide you and allow you to learn more effectively. The flip side of that is, of course, you know, I have a hundred thousand-word essay, and I would like Claude to please write that for me. And so there's efforts to do cheat detection, you know, that we've had for a long time and try and extend those to detect, to pick up these models. It's something that we need to think about.

00:57:42 I've kind of personally, I'm more interested in encouraging students to figure out how to use these to augment their own abilities. So, you know, use it to brainstorm. My grad students, their writing has improved significantly in the past half year. And partly because, and they've told me this, they've started going and iterating with, with ChatGPT saying "What, you know, provide a critical review of my introduction for this paper." And then they adjust based on that feedback, what they, you know, what they, what they did right, what they did wrong. And so it helps them improve their writing. So, it is, I think if used correctly a learning tool I'm maybe a little less worried about the destructive kind of implications of learning and more about what we can do to use it to make learning easier and better.

Jon Krohn: 00:58:25 Yeah. You and I see everything the same on this. I mean, for me personally, I mean, I'm sure it's the same with you. Our space is extremely fast-moving, and so I'm constantly needing to be learning. And I, these tools, like I, I'm a huge, I love using the GPT-4 ChatGPT interface. I find it super convenient for copy and pasting code that I have problems with. It obviously has limitations. If the code is something, you know, if I want to be using some cutting edge Hugging Face library that's only come out in the last week, that's not gonna be able to be covered, at least right now by the, you know, without any plugins with GPT-4. But for, you know, when I run into a SciKit-Learn error or a Pandas error, most of that API is static year over year.

00:59:11 And so it's amazingly accurate at like, and, and the way that it talks me through, like, it'll say such encouraging things like, you know, "I can see why you did it that way. It really makes a lot of sense that you would do it that way, but it's gonna throw an error because of this." And that's a little tricky, but just keep going and, you know, and. So, so there's actually, there's a bigger point there around, which I think I've talked about on the air before, which is that I think in general, interacting with ChatGPT and maybe people interacting with Vicuña as well, which I admittedly, I haven't actually interacted with myself directly very much. I have used the chat on your website like a little bit just to be like, okay, it works cool.

00:59:51 But this, this like friendliness, it actually I think makes me nicer in the things that I write and the things that I say because I'm getting that, like, that kind of, that, that positive reinforcement. But yeah, it's been a really amazing tool for me to be able to learn coding mistakes, maybe writing mistakes that I make. I think that education needs to change, to embrace this in the same way that it might've with a calculator or a computer. And it's not surprising to me to hear somebody at Berkeley where you're surrounded by, you know, it's the top university in the world. You have so many clever people around you, that the students and, and, and even postdocs, these are people who are going to be able to take a tool like this and use it to augment themselves and be better. I think the people that worry the most about these tools are, if you're not in that very top drawer, then there's a lot of our education system is based around really dull and often not useful memorization and regurgitation. And in those settings, unless the person is being, you know, has paper and pencil and is, and, and proctored exams yeah. It's, it's, it's just, it's, it's, I don't know, it's kinda obvious to me that education needs to change and, and go in the kind of direction that you're describing.

01:01:17 Anyway, that's probably enough on that topic. So, Joey a number of times through this episode, you've, we've alluded to this idea of open-source versus closed-source, the kinds of the pros and cons. It seems clear from initiatives like Vicuña, like Gorilla that you are a big proponent of open-source. So what are the kinds of pros and cons of these two different approaches?

Joey Gonzalez: 01:01:41 Yeah, it's a great question. And it's one that's, that's we've been grappling with at Berkeley. You know, all the labs that we've built have been around you know, doing great research and making that research accessible, not just in papers, but in open-source projects from Apache Spark, Clipper, Ray, Vicuña, the whole LMSYS effort. It's openness is critical to advancing the field, to advancing research. But there's a problem and that problem is that these models are expensive. They're expensive to train certainly building that foundation model is expensive, even this instruct fine-tuning. If you do it correctly, you do RLHF and this, you know, you know, using reinforcement learning with human feedback, RLHF requires data annotation throughout the training process. You know, the fact that GPT-4 is so nice is probably because they had experts, right, how to respond to tough questions.

01:02:35 And so that emphasis on good data the need to, once you've trained them, this model to serve the model requiring, you know. Let's just say you know, a few 8100s, it's pretty expensive today, but to serve a large or an ensemble of large models, which is, you know, the kind of the alleged GPT-4 setup at 175 billion parameters, that's incredibly expensive. So, there's a lot of costs associated with these. So I want the open-source community to succeed. But if I had to bet the analogy that I would draw for where these technologies will go is search. I think, you know, take, take search, web search, there are a few major search engines. There are regionalized search engine as well. Web search, much like these models, requires a very large amount of data, a large amount of compute, both to build the data and then to maintain it, and then to serve it.

01:03:33 It takes engineering skills, it takes a lot of safety systems. Making one of these technologies at it, at its peak, you know, one of the best in the world is expensive. And so I think we'll see something that more resembles search. Just like with search you use open-source search probably all the time in, in, you know, the tools that you use on your computer. If you're at an enterprise, you might be using one of these open-source search platforms that's hosted in the cloud. So there, there are large, like large major search engines that will be closed and, you know, will probably continue to be closed just like OpenAI. And, and Claude and, you know, the big LLM companies. And then there are smaller open-source search efforts.

01:04:15 What I hope to see is that the research community will continue to advance the open-source technologies, so they're good enough that, you know, if I'm trying to teach students, there might be a specialized model that's good at giving feedback on Python data science exercises. We might still host it, we might actually pay someone else to host it. But that model being something that we can control and innovate on will be critical. I think when I look at something like Gorilla, I think it'll actually be an interesting mix where you might ask one of these major commercial technologies to break down the task of booking my flight into important steps. And then you might call out to more specialized variations of Gorilla for any one of those steps to succeed, you know, to do that more narrow task. I, yeah, it would be wonderful if the future were, were, you know, the GPTs of the world were purely the open-source. The research community develops them and anyone has access to them. But I do think just even the cost of running them is so high that really, these large models will probably more and more be dominated by major organizations pushing, pushing them.

Jon Krohn: 01:05:20 That was a probably the best explanation that I have heard of why closed-source will continue to dominate, at least at the very cutting edge. That analogy to search, I hadn't heard somebody do that before, but the way that you described it is expensive human augmented data, huge GPU clusters, engineering ingenuity, and then lots of these safety checks. In order for Google search to be able to operate effectively, you need all of those things, and it's hugely expensive. And so, yeah, you're, I think you're absolutely right that we're going to a world where you know, a relatively small number of big tech firms that are able to make these hundreds of millions of dollars of investment continuously. Like, it isn't, you know, it's not like we get to GPT-4 and we're like, okay, it's done, right? We've got this.

01:06:08 It's like, it's this constant, very expensive race to be staying at the cutting edge. It'll be interesting to see if it can, because I mean, I guess this was a, this was, you know, I was a lot younger, so I don't know how much I was thinking about it critically or competitively, but when different search options were emerging, like, you know, when I was in high school or elementary school, there were things like AltaVista was like, you know, something that I guess I would've used and other people would've been using. But I don't remember it being this kind of, I don't remember the stakes being so high, there being so many competitors, like there are in this right now.

Joey Gonzalez: 01:06:45 Right, yeah, that's a good point. So there, there are places where this analogy breaks. So, the amount of energy, the kind of realization of the impact was sooner here. I think, I mean, search was pretty exciting when it was taking off, but like the kind of capturing the imagination of the world this technology has done that faster. I don't know if that favors commercial entities or not.

Jon Krohn: 01:07:10 Yeah, I don't know.

Joey Gonzalez: 01:07:10 It's, you know, here are things that might break my prediction, and I will say I would love it if I was wrong. It would be great if these things become vastly cheaper to run, to maintain, to develop. [inaudible 01:07:21] would break it. One of the things that works well in open-source is if I build something and you can make it better so I can release it, you can make it better, I can take your thing and make it better. With Vicuña, there was a little bit of that. So Facebook releases Llama, we make it better. But it's not clear to me that you can keep fine-tuning the fine-tuned model and get a better and better model.

01:07:44 And in fact, this is one of the big questions I have for my students, in Gorilla. Can I, can I fine tune on one API and then fine-tune another? What's the cost or advantage of doing that? So far it doesn't, in fact, it hurts trying to fine-tune on too many things. There's this problem, something catastrophic forgetting. So as I keep fine-tuning on new data, I probably need to go back and fine-tune on old data, which is really, I need to do more training. I think today, yeah, in fact, I don't know what the OpenAI and others are doing as they get tons of new data, if they're kind of restarting from earlier checkpoints or starting from scratch. So, basically making this more cumulative where our, where the open-source community can work together is something we need. And it's hard to do. Just sharing GPUs is probably not enough. Making the model smaller is also something that, you know, we're excited about. But, you know, it's hard to do. You're trying to compress human knowledge at some point that gets hard to do. And then if you can't compress it, it takes a lot of resources to use it which makes it harder and harder for the open-source community to do it without capital investment.

Jon Krohn: 01:08:46 Yeah. I think like another project area that we could think about is something like the Unix operating system, which is now the foundation for like every server in the world and all Mac computers. And so it, it's interesting to think how, like that came about, but it doesn't require all these things that you mentioned around search. Like, I think search is more like these conversational models where yeah, it, it's, it's just this constant updating of data, like in order for that, you know, today with GPT-4 or, or other, you know, cutting edge commercial models, we don't have at least embedded within the model weights, because of these accumulation problems that you're describing. And also safety things like, you know, being sure that you're safe even though you've added in some new information that just came out an hour ago. Yeah. Those, those kinds of problems I think make it a lot closer to the search problem. Whereas with, yeah, with like the Unix operating system, it, it is so easy to accumulate, where you're like, okay, you have this base code, and humans can look at the code and understand their little piece of it and make it better. Yeah.

Joey Gonzalez: 01:09:53 It's a tough future. Yeah. I, maybe I have one more thought here. Because so one more thought on the open-source space is, you know, we can still make progress. And in fact, I plan to continue to make progress in the open-source space, even knowing that, you know, the best models in the world probably won't be mine. But what I can do you know, something I'm trying to do right now is to like, explore what are these trade-offs between, you know, RAG and fine-tuning with the hopes that maybe the insights we have at smaller scales will translate. And I think actually if you look at OpenAI's success, that's one thing they nailed. They took this hypothesis that if you scale machine learning, you scale the data, you scale the model complexity, you'll get better results.

01:10:34 But they didn't do what other companies Google did. They didn't just go to, you know, dial it to 11, they started small and they got signal. We can continue to stay small and get signal and understand how these different things mix and maybe influence where these big technology giants will go. And, you know, just like search there will be smaller entities in other countries, in other regions of the world that serve smaller markets that serve specialized languages. And I think we can have impact there as well. And then finally, you know, the open-source models we build might be good enough for a lot of basic tasks. You know, what you want to read through all your emails and figure out like, what was the conversation about? Maybe you can get away with the simpler model for that task. There's still costs associated with it and, you know, it still might be a commercial activity that does this, but the models themselves, we can continue to develop and hopefully provide insights again, will shape the bigger efforts as well.

Jon Krohn: 01:11:26 Yep. I am so grateful for the work that you personally do on Vicuña as well as everyone else doing this amazing open-source work. My business depends on it. I know there, there must be many thousands of other businesses out there that do as well. For us being able to take something like Vicuña or because Vicuña actually isn't commercially licensable because it's based on Llama 1. So, like we had been using as our starting point open-source LLM for a lot of our specific tasks in our platform. We were using Dolly 2.0, which had a commercial use license, Databricks provided it. But now we're switching over to Llama 2 as our starting point.

Joey Gonzalez: 01:12:06 Very good choice.

Jon Krohn: 01:12:07 And, and it's, and it's super, super inexpensive to fine-tune into our tasks because we, with a thousand examples of some specific task at least in our kind of application we are able to fine-tune using a parameter efficient approach like LoRA which I can't remember if I already talked about it in this episode.

Joey Gonzalez: 01:12:30 Yeah. We didn't talk about LoRA. It's an interesting method. So we could, yeah it'd be fun to talk more about it here. So, we might have in the past. Yeah. Yeah.

Jon Krohn: 01:12:36 Oh yeah. So I had a whole, a whole episode on it, episode number 674. And the reason why I was wondering if I talked about it is because the episode that I recorded immediately before I started the conversation with you I did mention it, but you and I have not on air yet today. So, yeah, I mean, that is, so I can like really briefly introduce at a high level the value of this, which is that it allows me to take an like Llama 2 now. We're literally doing this right now. My team is taking Llama 2 and using LoRA to train for typically hundreds of dollars worth of compute and mostly actually using servers that I built by hand a couple of years ago. And those are still good enough for like, if you want to take a 7 billion parameter model or a 13 billion parameter model, I can still run it on GPUs that I bought years ago and on a server that I built myself. Like, it's like these, it's, it's efficient enough that you can do that. And so, you know, it's at no, there's no extra cost to me to do that. Electricity is cost.

01:13:35 And yeah, and, you know, we have training examples from our platform. That's the key. I guess that's maybe the key point of this whole conversation is it's about having the great quality data. So, we're able to create these very narrow models that do very specific tasks and it might be the case that we're able to train it to do a few of the generative tasks that we need on our platform. But another really cool thing you can do is you can switch out just those LoRa weights. So, and you can do that on the fly in real-time with your users so that you're not hogging lots of infrastructure Where you have, you could have one GPU running a 13 billion parameter Llama 2 model with these, just these small number of LoRA weights, which is typically gonna be like, I mean, it depends on exactly what hyper-parameters we pick, but it could be like half a percent of all of your model parameters are gonna be these new LoRA weights that you added in, and then you can swap out those LoRA weights in real-time instantly for different generative tasks that your platform has. And yeah, so anyway, I've talked a lot.

Joey Gonzalez: 01:14:41 So LoRA is, is pretty exciting. My students so far have not found LoRa to be as good. And that was one of the downsides. So, maybe to highlight something you said in that. The training often in fine-tuning, even if you don't use LoRA isn't so bad. Where you get burned using the fine-tuning method is if you have a separate model for every single user and then you go to serve it. And if these models, you can fit one or two 7 billion parameters depending on what GPU you're using, you can maybe fit one or two in your GPU, you start to run out of GPU memory for lots of models.

01:15:15 LoRA changes that narrative because it, it says I have a base model and then some low-rank approximation that can apply. I actually haven't seen good results on not materializing that low-rank approximation, but it sounds like you guys have found ways to do that, to be able to serve and switch out the low-rank approximation quickly. So, so that, yeah, that makes a big difference and allows you to have lots of fine-tuned models, but not actually have lots of fine-tuned models. You have some base model and some additive component that you can, can swap quickly. And that added component's low-rank, so it's small, means you can fit lots of users' fine-tuned versions of that component in a single GPU. And that, that, again, I can't stress this enough, tends to be the bigger cost in life is not the training of these things, but the using of them. [inaudible 01:15:59] the foundation training, which is still very expensive, but, you know, often it's fine-tuning is cheap. It's the use that that's expensive and LoRA can make a difference there. It's neat.

Jon Krohn: 01:16:09 Yeah. It's really neat. All right. So this has been an absolutely incredible discussion so far, Joey, around all of the academic things you've done. But that just scratches the surface of what you've done in your career. So, starting with what you're doing right now, you're, in addition to all the academic stuff that you do at Berkeley, all the research, all the teaching, you are Co-Founder and Vice President of product for a startup called Aqueduct. And so Aqueduct has developed an MLOps framework. You kind of, you alluded to this earlier when you're talking about SciKit-Learn a year ago and now deep learning say. This, so, this MLOps framework allows you to define and deploy machine learning and LLM workloads on any cloud infrastructure. So, yeah, fill us in on like why you founded Aqueduct, what [inaudible 01:16:53]that you saw, and how it simplifies prediction infrastructure for data science teams.

Joey Gonzalez: 01:16:57 Absolutely. Yeah. So Aqueduct is a fun story. It's, you know, I was going up for tenure a few years back, and some of my really strong students working on serverless computing we're, we're graduating. We had just finished some work on a system Clipper and [inaudible 01:17:11] building tools for realtime prediction. Serving those technologies actually influenced everything from the TensorFlow serving to like the current Hugging Face, you know tiered architectures. So, we were excited to take this technology, bring it to the world. I was excited to take a break from teaching and go out and do a startup for a little while, take a leave of absence. And so we launched a company. I was VP of product, my CEO Vikram was my former student. So I now reported to my student, the boss. So, we launched a company to bring this serverless technology, this prediction, serving technology kind of fused together with the hypothesis that, you know, data scientists, the thousands of students I was teaching would become the future of, kind of, of engineering, of solving problems. And they would be great at machine learning, great at data, great at kind of thinking through the connection between data machine learning and business, but not so great at infrastructure, not so great at, you know, running cloud tools, running machines. And we knew that because we don't teach them how to run cloud infrastructure in our current data science program. And many students come out, you know, expecting a Jupyter Notebook to exist in the world and expecting data tools to sort of just connect to it magically. Something we should probably fix about the program.

01:18:30 But yeah, so that, that was our, our class you know, that where we see the people headed. And in some sense, serverless computing, all this technology that had been developed to make engineering simpler really would make data science a lot simpler. And so we launched a company to bring those, those ideas, those technologies we've been developing part in the open-source into a commercial service where any data scientist or any person working with data and machine learning could go to that service connect, you know, launch models. We would manage machines for them, connect those models to various data sources. We'd manage all the kind of data plumbing to make it really easy to take ideas, develop models, and then put those in production to solve problems.

01:19:08 We launched it talking to data scientists. I guess when did we exactly launch it? I want to say end of 2020? Yeah. End of like November, 2020. We very beginning of the company really took, kind of took off in 2021. We talked to a lot of data scientists around this point in time, and the first kind of disappointing discovery I had in my research career is that people weren't building realtime prediction serving systems. They were still plumbing basic data through SciKit-Learn pipelines, which in retrospect made a lot of sense. They realized that without, you know, having a company built around machine learning, it's better to take your ideas as your predictions, dump them back in a data warehouse where anyone can consume them, where the tools of visualization already speak. And so they had built pipelines that bring machine learning in, but in, you know, cumbersome ways using airflow and dump it back into the data warehouse.

01:19:58 So, the very early incarnations of our product was really around trying to simplify that process of integrating with the data technology people are using with the compute infrastructure to support interesting pipelines, monitor them, provide visibility to, you know, make it easier to be effective data scientists. And then we started getting some, you know, excitement around the project. We got some, you know, early users, and then LLMs took off and it, it, it forced us to sort of reassess where we were. A lot of people are like, well, I do have my pipeline working. It's a total mess, but my team has asked me to look at how LLMs will change, change everything, or, you know, my interest would be now if I had spare cycles in my life, not to make what I do better, simpler, faster, but to bring LLM technology to change the company.

01:20:45 And I think actually that's wise a lot of people thinking about what is really going to be a monumental change in technology, what it means for their business, the right people to do that. The people at the frontline would be the data science team, the people that we were talking to all along. And so we also went back and started like, what is challenging about LLMs? How's that changed the narrative? Good news in some sense, it brings us back to what we wanted to do in the first place. So people want to serve lots of models. Now these models are expensive again. They need GPUs, they need interesting resources, they need cloud infrastructure. That's hard to find because everyone's already bought all the 8100s. Can I make this run on a different kind of hardware?

01:21:19 So, it was good news. It was a reverse pivot, which is a weird experience in startup land where we like realized that what we were doing in the very beginning was kind of the right way to go. So, we brought in some of the technology. We already knew how to do back into the core product and started focusing more, not on the open-source, but on our hosted option. And we have a release coming out soon of the, you know, updated version. To make it, again, easy to do this stuff with LLM technology. What LLMs do that's kind of fun is, you know, I have a lot more to manage. I have prompt, I have resources, I have spend that I need to keep an eye on. There's different kinds of models for different sorts of tasks. There's fine-tuning. So a lot of, of new challenges, things to be done in the data science kind of machine learning lifecycle.

01:22:08 And the other thing that's kind of fun is the people. It's no longer just data scientists. Engineers, even people who are like, I don't really do that stuff, but I played with ChatGPT for my book club. I think I'd like to use it to do something interesting for my business. So, it's a bigger market of people, different skills and expectations, which is a, you know, as a product person makes the problem a little bit harder. But I think ultimately an opportunity for us and, you know, kind of a pivot to our roots, which is something I was excited about.

Jon Krohn: 01:22:36 Yeah. So I can see what you're doing with Aqueduct. I think, let me try to like, understand it through maybe like a use case. You can kinda like walk me through like a user story. So, I deploy LLMs as part of my business. And so it sounds like with Aqueduct, I can use it, I can like upload my model weights, I guess, and then like I can, I can configure it so that I'm keeping track of my spend. And I guess I don't need to necessarily be maintaining my model on my own servers. You will handle conversations like, you know, my, my users come into my platform, they type, you know, we're like an HR tech platform, so it's like, find me data scientists in New York. My user types that in, that input gets sent to Aqueduct, but then Aqueduct runs my model and then brings the result back to my user, the generative result, that kind of thing.

Joey Gonzalez: 01:23:37 Kind of. So, you, what you described, we can do a lot of our users, and in fact, it's something I would even tell people if you're first experimenting, use GPT-4 or Claude. So, a lot of the cases people are just calling out to external models to begin with. What we're trying do is make it easy to put those pieces together. I don't love this analogy, but you know, people are familiar with Langchain. We're trying to make a more simplified hosted Langchain. And in fact, the use case we're focused on right now is more in the RAG context. And I think that's where a lot of people will probably start. I have a question. Maybe it's a support ticket question. I, as a, you know, developer, this want, that support ticket question to maybe look up related questions, ask an LLM, you know, who should handle this question based on those related questions then maybe direct the person and maybe do that in real-time.

01:24:24 To do that, I need to, I need to maintain a database. Perhaps I'm using Pinecone, so I need to put my customer conversations in Pinecone. Shockingly, Pinecone doesn't do its own vectorization or its own maintenance. So we need a system that sits there and watches conversations and keeps that up to date, something we do. And then, you know, as this conversation runs, you need compute to go off and hit, say GPT-4, pull maybe, which, which things I need to get from Pinecone, put that back in, make another quest to GPT-4. So it's, that's small amounts to compute that needs to exist kind of in ephemeral state. What we wouldn't make it easy to do is in Python, on your machine, you write that workflow and then when you deploy that workflow with us, we then can run all those steps for you in the cloud. You can turn your laptop off and connect it to the various technologies. And then also it is all that kind of tooling around that. So like, what's to spend? And so we can break down spend from each of the steps in your workflow just to make it easier to kind of experiment and show value with these technologies. And then as you start to deploy that, using the things you've already been building to make that, that kind of management process easier.

Jon Krohn: 01:25:26 Nice. I gotcha. And so, yeah, and then in terms of name, I guess Aqueduct, the idea here is that you have like, like kind of liquid flowing very easily between different systems, like, I don't know.

Joey Gonzalez: 01:25:37 Yes. So let's talk about the name. As I said, multiple times names have not been my strong suit in life. I should even reflect back on my early startups. So we launched a company called GraphLab based on my thesis work.

Jon Krohn: 01:25:48 Oh, yeah. Actually that changed names so many times. It was wild to watch.

Joey Gonzalez: 01:25:52 Yeah. So GraphLab was [crosstalk 01:25:54]. Dato and then Turi and, and the story of each is interesting. We can come back to that. So with Aqueduct we actually started as Spiral Labs because we wanted something that was kind of like interesting researchy thinking of kind of how things converged to an idea. With Aqueduct it fit what we were doing in the beginning more closely, you know, bringing the high-value stuff from the lakes, the data lakes, the world to where it has impact. The some kind of essential piece of infrastructure. Aqueduct was about simplicity, elegance. And so it was a very heady name that unfortunately is not easy to spell by. We spelled it correctly, but most people do not. It's AQUE which hurts our SEO. I think in the long run, we'll probably change the name to something more in the kind of LLM space thinking kind of the technology that people are using. But yeah, Aqueduct, you know, it had a meaningful name, which is still appropriate today, but it's also probably a little bit harder to find and, and that the connection's a little bit less, less obvious. So.

Jon Krohn: 01:27:00 Cool. Let's actually, let's talk about Dato, Turi, GraphLab. So, Aqueduct, I think based on, you know, what we could find online is, is your second startup.

Joey Gonzalez: 01:27:13 That's correct. Yep.

Jon Krohn: 01:27:14 So, previously you co-founded Turi. And this is something that I am familiar with because I, remember I was using GrapfLab, although I now it's, I mean, this was 10 years ago, so I don't actually now exactly remember what I was doing with GraphLab, but I remember that I was using it, I remember the name change to Dato, and the main thing that came to my mind was I was like, wow, these guys must be doing so well to be able to buy a four-letter domain name, like Dato, that's wild. But I don't know, I personally thought it was a kind of a silly-sounding name.

Joey Gonzalez: 01:27:50 Yeah, it was. All right, so let's talk about the story of GraphLab. GraphLab was, you know, my, my first real company, the first company I launched. I launched it after finishing my PhD on a project called GraphLab. GraphLab itself was actually a sort of a joke that was launched out of an insightful you know, workshop. GraphLab was a system for graphs. It was designed to graph computation, designed to support the trending model of the time, which was graphical models. Many of your viewers may not know of graphical models because they've since been replaced by neural networks. When I was doing my PhD, we made fun of neural networks. Neural networks, that's silly technology. It's not principled, it's not statistically sound. Graphical models are a much more principled way of approaching problems.

01:28:36 And then they came up with this branding, deep learning. They just rebranded those neural networks, those silly guys they were right, and come back to how they got that right. But GraphLab was designed for graphs, not neural networks. It was designed for graphical models. It was actually a very successful open-source project from a group, my group at CMU, Carnegie Mellon University is not on the West Coast. It doesn't generally have or enjoy as much of the kind of publicity that something like Berkeley or Stanford would get. Yet, GraphLab became pretty, pretty popular. It was popular because it helped solve problems in matrix factorization content recommendation, which was trending at that point in time. It was, maybe it was the graph analog to something like Apache Spark something that I would then later on go to de analog. I would, you know, connect the two.

01:29:23 But yeah, we launched a company GraphLab. First thing you should know for those building companies, don't name your company after your first product. That was silly. One of the first discoveries we had with GraphLabs is that a lot of people are excited about content recommendation, but what's the graph? I have a lot of user clicks. I don't understand. Like, well, it's obvious you got, you know, connection, clicks, and products, a bipartite graph. You can do clever tricks because, you know, it's a bipartite graph. And they're like, oh, cool. Do you guys do something more around data? We got a lot of data we need to process. So, one of the big observations at that point in time is that, while machine learning was exciting. It was the kind of forefront of what people wanted to do, the aspiration. The reality was data was a mess. And projects that made sense of data made it easy for me as a data scientist to work maybe on my laptop on large data sets and not have to set these silly Hadoop or spark clusters was actually more appealing.

01:30:13 And the core technology in GraphLab connect to another project in our group, craft Chi had these kind of way of partitioning problems that could be distributed. That same innovation allows you to optimize things for out of core computation so that you could operate on your laptop. And so we very quickly kind of expand the scope of what we're doing to a technology, a Pandas-like technology that works on, you know, terabyte files. If you have that much disc space on your, you know, 8 gigs of RAM machine that you had at that point in time. And made it run fast enough, you could actually do pretty interesting analytics without a big spark cluster. You could do graph stuff, but you could also do basic data visualization, basic streaming, online machine learning. Things that were actually pretty useful.

01:30:51 And so we actually started getting a fair amount of users. And the Graph Lab name was confusing. So Dato, Dato is Portuguese. My advisor Carl was Portuguese, so came with a name that would, you know, reflect a byte, a single piece of data. Seemed reasonable. And, you know, Graph Lab wasn't, you know, it was confusing people. So we changed it. Turns out there's a data backup company in the Midwest that also has that name, but with two T's, pronounced differently, but close enough that it created some conflicts. And then we ultimately decided to change our name to Turi, which is actually a pretty neat name in the end. We had some help coming up with it because I'm bad at company with names. And so, so are, you know, most academics I guess. And so Turi embodied this idea of, you know a general platform for doing interesting computation on data at, at kind of, at any scale, both big and large that supports machine learning visualization. It was being used by a company Apple. And they were pretty excited about it and working closely with those in the development of new features. And in the end they were pretty excited and have a lot of money. And so we're able to acquire our company. And so that same technology went on to be parts of Apple Watch, parts of the iPhones, like it's all over the Core ML. So, a lot of kind of really neat impact from a team and, and the project.

Jon Krohn: 01:32:11 Absolutely. Yeah. Congrats on $200 million acquisition, fantastic success there. And yeah, I guess and the Turi thing, what's it's, is that like Alan Turing that it's like-

Joey Gonzalez: 01:32:21 Reference to Alan Turing.

Jon Krohn: 01:32:23 Yeah, yeah, yeah, yeah. Perfect. Amazing. So let's take a quick break here just to strategize because we have like nine minutes left to record. So, was there anything else? You know, I took a quick skim just now of everything that was left in kind of the pre-research that we did and, I don't know, I was thinking maybe like question five about like the dynamic deep neural networks and explainable RL. Like maybe that's kind of like the most interesting stuff that's left, but I'm not like, wedded to talking about that. We could also just move to the audience questions.

Joey Gonzalez: 01:32:57 Yeah. So, I'm guessing that the audience, the people watching this podcast will be LLM motivated. So I'm, I'm happy, maybe I'll talk briefly. You can ask me, you've done other stuff on neural networks talk maybe how, what you've done, maybe how that relates to what's going on today. And I can kind of make a quick connection, then we can go to audience questions. If that works.

Jon Krohn: 01:33:15 Nice. And so very cool to get a sense of what you've been doing commercially, the amazing success you've had there as well. What are you really excited about right now, Joey, that you're tackling either, you know, with everything you're doing, it probably blends academia and eventually the commercial world as well?

Joey Gonzalez: 01:33:35 Yeah, it's a good question. So, as I said, already thinking a lot about this kind of connection between fine-tuning retrieval, a little bit more on context. I'm excited about kind of visual language understanding. And I think, you know, computer vision is about to get radically transformed by these visual language models. The ability to, you know, take my entire photo album from my family vacation and then have it read the album, look at the album, convert it to words, tell a story, maybe put, you know, Ken Burns effects in and a narrative around it. Provide highlights, you know, this, I think we're about to see kind of big changes in vision. My research group has been working on that. Autonomous driving is another area that kind of, it went quiet because of developments at LLM or kind of renewed focus in electric vehicles.

01:34:21 But I think we're gonna see that emerge again. And I'm kind of excited to see what these advances will do for autonomous driving. Probably the hardest part of autonomous driving is actually understanding what other people will do, kind of the signals around you. It's this prediction problem. And having kind of general foundation models might change that, that we could train on, you know, endless amounts of webcam data or the dash cam data or, you know the street view data. So, lots of opportunities to change autonomous driving. So that'll be exciting. And then my group has been thinking a lot about how to make neural networks treat data kind of more dynamically, differently.

01:34:55 Right now, LLMs are kind of neat. They and, and they, they, you know, grow based on your question, you'll get different kinds of answers, different amount of computation, but still, fundamentally, each token is treated the same. It runs through the 175 billion parameters to predict, you know, that the next word is "the". Thinking more dynamically, like the next word's, obviously, "the", you don't really need to ask 175 billion parameter run through, you know, hundreds of GPUs to answer that question. And so being more intelligent about how we switch between models, between levels of complexity to support interesting conversations, you know, use advanced models when they're needed and not when they're not is a, you know, a big opportunity. There are a lot of challenges in making that opportunity real from, you know, how we use key-value caches and kinda the underlying mechanics of these models. So, yeah, I'm excited to see kind of where the world will head looking in different directions for these kind of models and, and the kind of new applications as well.

Jon Krohn: 01:35:51 Nice. Very cool. I have no doubt that you'll continue to make an enormous impact as you have already in your, frankly, I mean, you're already, you're only really relatively early in your career. You just got tenure a couple years ago. It's wild to think what you'll do in the rest of your career. Which actually brings me to a question that I didn't prepare you for. But, you know, kind of going back to the Jeanne Calment thing that I was describing about, you know, how quickly things change over a lifespan. We're, we're watching things change unbelievably quickly. You're playing a huge role in that personally. What kinds of things do you hope you might see in your lifetime that are, you know, far beyond what we have today? Like, you know, trying to project ahead, which obviously is gonna be really hard. You can have huge error bars on anything you say next, but what are, like, what's like an amazing vision for how things could be a few decades from now?

Joey Gonzalez: 01:36:47 That, that's a great question. You know, when I was going out for tenure, I started asking myself, am I doing research that's, that's forward-thinking enough? Because it's having impact, it's having impact now. It should have impact in the future. One of the things that got me started on, which is something I'm, I'm excited to see succeed, it hasn't yet is the use of these kinds of technologies to really tackle climate change. And what are the, you know, I asked for my grad students, how should we work on this? Should we use less data centers or, you know, is there something more profound? and they came across some really cool work going on at Berkeley to design new materials, to pull carbon out of the air and to build better batteries. And these new materials these metal organic frameworks require chemists to try millions of combinations of things to figure out, you know, whether or not it works.

01:37:36 We don't, like the science of kind of predicting the capabilities of these materials and whether or not they're synthetically accessible is not there. And so maybe more broadly, will our advances in AI allow us to really push science, fundamental science forward and tackle what is probably the biggest challenges of our time. So, something like climate change. That I want to see that's really hard to do. I started doing it. We failed so far, but hopefully we'll get better at it in time. Maybe the early, the early hope I've seen in some of the work that we've, you know, started to see a little success on is things like stable diffusion. Turns out you can use that to create molecules too and design the underlying structure with certain goals in mind. And maybe we can start to build foundation models of, you know, of chemistry, of molecular design that would change how we do that.

01:38:25 Same with medicine. CRISPR opens up new capabilities for what we can do, but now we have to, you know, design the things that we'll solve, the problems that we need to solve. And, and again, maybe some of these AI technologies will help advance that so that we can, you know, tackle some of the biggest medical problems of our time. Autonomous driving - people shouldn't be dying. I think people die every few minutes in automobile accidents in the United States. If we can make cars safer using these technologies I'd be excited. And, and you know, there's a lot of race to make, you know, smart taxis, I'm more excited about cars that don't crash. Just making, making the road safer, making the road safer reduces emissions because we have la less accidents, less traffic. So, a lot of potential impact there as well.

Jon Krohn: 01:39:09 Fantastic. Those were all great points. Using AI to help us tackle climate change, have a big impact in medicine as well. And yeah, people shouldn't be dying on the roads. I agree with that a hundred percent. It's still, it's the most dangerous thing that you can do. And most people are doing it on a daily basis. Yep. So, something really interesting related to that, the generative idea of like a generative chemistry is actually, we talked about that a little bit more just a couple episodes ago in episode number 705 we had three really senior people like three of the most senior people in Syngenta on the show. And they talked about exactly this, about using generative models to predict agricultural compounds that could help feed the world. Yeah. Really cool stuff.

01:39:59 All right, nice. So as I mentioned earlier in the episode when I brought up Wess McDermott's quote from Sunset Boulevard that the Vicuña quote, I did ask our audience if they had questions for you. We had some great questions. I think some of them we already answered in our conversation. So we had a great one from Evan Wimpy about what the future of LLM training will look like with respect to smaller or larger models, open versus closed source. I think we've covered that pretty comprehensively on the show. But here's one from Michael Lockhart. So he's a senior engineer at Rouge, and so he's wondering if we can take Llama 2 and do clever iterating like you did on the original Llama. So there, there's this huge opportunity with the original Llama to fine tune it and have it be able to do more conversational style of conversation. So is there a big opportunity to do some kind of fine-tuning with Llama 2 as well?

Joey Gonzalez: 01:40:54 Yes. So can we fine-tune Llama 2? We are fine-tuning Llama 2 right now. Yeah, so we're, we're gonna do that. One of the, maybe a variation of that, that question that bothers me is should I fine-tune the preinstruct fine-tune Llama 2, or should I fine-tune Llama 2 not preinstruct fine-tuned? Oh, no. It's something we're trying to figure out. I, yeah, I don't understand fine-tuning. I will say that I will confess it. I don't understand it, but I don't think anyone does yet. Fine-tuning, it's more training, it's more training with the funny loss. There is funny learning rate that goes up and then down and, you know, where you how you set that has, you know, significant impact. Yeah, we're gonna do that. I think lots of people will take Llama 2 and fine-tune it to do all sorts of things and we'll hopefully get clarity on should you fine-tune the fine-tuned version or should you start from scratch. Yeah, so, so certainly yes, we'll see lots of updates Llama 2, and we're there definitely working on it.

Jon Krohn: 01:41:50 Nice. Can't wait to see what comes out of that. Who knows, by the time this episode is released, it might already have been published. That's how quickly these things move. All right, fantastic. Joey, before I let my guests go, I ask them if they have a book recommendation for the audience.

Joey Gonzalez: 01:42:07 Yeah. So book recommendations. So, sadly I don't have a lot of book recommendations. I have two kids, so I read my nighttime reading that's fiction typically revolves around princesses and dragons. But I will say, you know, one thing that I've found helpful as I try to follow this space, certainly these podcasts have been great. Having access to my students pointing me at, you know, the latest paper every single day has been helpful. One of the neat things that's happening right now is ICML and I suspect the proceedings of ICML will probably have some pretty exciting stuff in it. I haven't personally found great points of aggregation for what's going on. And maybe that's one of the bigger frustrations and as an academic, something that, that we should be doing on the LMSYS site is like, here are the papers you should have read. And it is something we could do in the future. So, maybe check the LMSYS site and we'll try to put something like that together. Because the space is moving very, very fast.

Jon Krohn: 01:43:06 As I like that. I don't think we had ever had a guest recommend the proceedings of a conference until your colleague Raluca Ada Popa in 701 and now another Berkeley faculty member has done the same, and it, there is a huge amount of value in those. I-

Joey Gonzalez: 01:43:22 So, actually, there's one more place I'll recommend. You know, that's kind of, it's good. It's self-promoting only, only minorly. The BAIR, the Berkeley AI Research Group has a blog and the students put a lot of effort into making their blog posts, and they have like a whole review process and they put a lot of effort into that blog post to make it as accessible as possible, while also staying as technical as possible. Usually with videos, descriptions, animations of the math, the ideas. It's not a bad place to look at certainly what's coming out of Berkeley in the AI group. So, check out the Berkeley AI blog.

Jon Krohn: 01:43:59 Nice. Fantastic. And beyond the Berkeley AI blog, how can people keep up with what you're doing? Do you use social media at all?

Joey Gonzalez: 01:44:06 I have been trying to learn how to use Twitter or X or whatever it's called now. And I've been, I will, I will typically post, I tend to highlight my student stuff, so when students have something I will help try to repost it and provide some commentary. So certainly check out my Twitter @profjoeyg and then LinkedIn is something that I'm still also learning how to do. My CEO has asked me to become better at LinkedIn, so will I'll post there occasionally too.

Jon Krohn: 01:44:32 Nice. Just student CEO very well.

Joey Gonzalez: 01:44:35 Yes, that's correct.

Jon Krohn: 01:44:37 Nice. Yeah, I mean, for us with the show, it's interesting. I know in the academic world, typically people are more on what was Twitter. But yeah, it's interesting with, at least with this podcast we get easily 10 times more often, a hundred times more engagement on LinkedIn with literally the exact same content for whatever reason. So that is [crosstalk 01:45:04] yeah, primarily where I am. Nice. All right, Joey, this has been amazing. I had really high expectations for this conversation and you greatly exceeded them. This was the highlight of my week for sure. Yeah. So thank you so much for being on the show and maybe in a few years we can come in and you can come back and let us know how Vicuña 5 is coming along.

Joey Gonzalez: 01:45:20 Yep. Sounds good. Thank you. Thank you for having me.

Jon Krohn: 01:45:28 Ufff, what an experience that was. In today's episode, Joey filled this in on how Berkeley students spotted the opportunity to use ShareGPT as an outstanding data set for fine-tuning Llama and approaching GPT-3.5 level quality with their resulting Vicuña model. He talked about how leveraging GPT-4 for evaluating generative LLM outputs has improved with the MT benchmarks, but the OpenAI model nevertheless has biases to be aware of when you do this kind of evaluation, such as preferring the response presented first, preferring longer responses, and preferring responses that are closer to its own language style. Separately, he talked about how Gorilla leverages both RAG, Retrieval Augmented Generation, and fine-tuning to interact well with APIs and provided ChatGPT plugin-like open-source alternative. He talked about how his Aqueduct startup enables LLM workloads to be defined and deployed on any cloud infrastructure. And he provided us with his vision how over the coming decades, AI could help tackle climate change by helping design new compounds that fixed carbon dioxide from the air, making enormous impact in pharmaceutical design, and prevent tragic road deaths through autonomous driving.

01:46:35 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 Joey's social media profiles, as well as my own at superdatascience.com/707. If you two would like to ask questions of future guests of the show, like several audience members did during today's episode, then consider following me on LinkedIn or Twitter as that's where I post who upcoming guests are and ask you to provide your inquiries for them. All right, thanks to my colleagues at Nebula for supporting me while I create content like this SuperDataScience episode for you. And thanks of course to Ivana, Mario, Natalie, Serg, Sylvia, Zara, and Kirill on the SuperDataScience team for producing another phenomenal episode for us today. For enabling that super team to create this free podcast for you we are deeply grateful to our sponsors. You can support this show by checking out our sponsor's links, which are in the show notes. Or you could rate or review the show on your favorite podcasting platform. You could like or comment on the episode on YouTube, or you could recommend the show to a friend or colleague whom you think would love it. But most importantly, I hope you just keep listening. If you like, you can subscribe to be sure not to miss any awesome upcoming episodes.

01:47:47 All right, thank you. Cheers. I'm so grateful to have you tuning in and I hope I can continue to make episodes you love for years and years to come. Until next time, my friend, keep on rocking it out there and I'm looking forward to enjoying another round of the SpeedDataScience podcast with you very soon.