SDS PODCAST

EPISODE 913:

LLM PRE-TRAINING AND POST-TRAINING 101, WITH JULIEN LAUNAY

Jon Krohn:	00:00:00	Welcome to episode number 913. My guest in today's episode is Julien Launay. He is unbelievably knowledgeable about training LLMs, the pre-training part, the post-training part. We spend tons of time talking about that so you can get a full understanding of how cutting edge AI models are made and how his startup Adaptive ML allows enterprises to have fine tuned models for their particular use case available much more easily than ever before.
	00:00:30	This episode of SuperDataScience is made possible by Dell, Nvidia and AWS.
	00:00:35	Julian, welcome to the SuperDataScience Podcast.
Julien Launay:	00:00:41	Thank you very much. Happy to be here today.
Jon Krohn:	00:00:43	Yeah, it's great to have you in person in New York.
Julien Launay:	00:00:46	Totally. Yeah.
Jon Krohn:	00:00:46	And so it doesn't sound like you have a New York accent, though.
Julien Launay:	00:00:49	I don't. I come from France. I'm spotted within the first few minutes. I come from France. I actually moved to New York just a few months ago, so it's very recent for me. I think just over two months today.
Jon Krohn:	00:01:00	Welcome. How are you finding it?
Julien Launay:	00:01:01	It's really good. I think people ask me this a lot and I think it's an interesting question. I think it would be really hard not to enjoy New York. I feel every time I feel very boring saying, oh, it's really good. It's really good. I don't really know, honestly what negative things is. It's an amazing city.

Jon Krohn: 00:01:15 I think it's basically infinite because of its size, the restaurant turnover, new galleries opening. There's always new things to be doing, but I think especially in your first few months like this, it's so exciting. Like wow, a neighborhood like this. I had no idea it existed.

Julien Launay: 00:01:28 Yeah, yeah, it's really amazing. Very diverse, lots of stuff to do. It's kind of endless. There is always activity. It's super nice, really, really nice place.

Jon Krohn: 00:01:36 Nice. And so speaking of exploration, actually, you're our guest on the show because you wrote a bestselling book absolutely 10 years ago called The Guide Minecraft. So it's a Minecraft guide that you wrote in high school, is that right?

Julien Launay: 00:01:52 Exactly, yeah, in high school. So I used to play way too much Minecraft. I guess like many people of my generation or so, maybe also of the new generation, apparently it's making a comeback and I used to write for a website called Minecraft Affair, so friend domain and one day Pearson editor contacted us and was like, oh, we could do a guidebook. It's having a lot of success and ended up being part of this project, and surprisingly it ended up becoming, I think so year it came out, it's ended up becoming one of the bestseller in French, which is really funny because I can say that I owe the bestseller in French while it's a video game book. So your mileage,

Jon Krohn: 00:02:30 And actually this book didn't actually show up in our research of you, but you mentioned it before we started recording, however, it is kind of interesting because this kind of, did this get you in interested in programming in the first place?

Julien Launay: 00:02:42 Yeah, yeah, so not so much a book per se, but definitely Minecraft definitely got me into programming like plugins and modes, all of that sort of stuff. I used to run a few

servers, a few very large servers with a fund, and this was, although I think very interestingly, this was a time where Minecraft professionalized, whereas Earth was starting to be very large with tens of thousands of players that started making a lot of money actually on the side, the ecosystem started picking up, which in itself by the way, is an entirely other story. I think the world of Minecraft is actually fascinating, even from a business perspective and how it grew and all of that. But yes, that was very much the beginning of this and ended up doing some mudding, some all of this learning, Java doing this and spending once again, too much time on this. Maybe school performance dropped a bit because of that, but in the end it all worked out.

Jon Krohn: 00:03:33 It seems to be paying off. You are co-founder now and CEO of a firm called Adaptive ml, who are makers of something called the Adaptive Engine, a flywheel for enterprise ai, which continuously evaluates tunes and serves large language models, LLMs, so they're uniquely adapted to a business using smaller cost efficient models. Before we get too much into Adaptive, your company, I'd love for you to talk about based on your rich experience at Hugging Face, also at a company called Light On that we'll talk a lot about more later in the episode. Through that experience, you have tons of experience in creating LLMs that are useful for real life and at the biggest scale that LLMs come. So I'd love for you to start off by providing us with an overview of the steps involved in creating an LLM like pre-training and reinforcement learning.

Julien Launay: 00:04:27 Yeah, it's a very timely question as well, given that I think these steps are blending a bit these days. So take everything that I say with a crane of, so there's always nuance in this, but very broadly speaking, the way that historically large language models have been approached. First is through a pre-training phase, which is the bulk

historically of where the computer has been spent. Pre-training is during pre-training. We essentially collect data from all over the web, pretty much every books, every paper, pretty much nearly at the scale of modern pre-training, nearly every text in existence. I think it sounds very grandiose, but it's not far from being true and even nowadays, images, videos and all of this, and essentially the model is trained to very robustly predict the next world, predicts the next token. This is a step that is built to be scalable, to run at scale, that are essentially everything we have ever produced on tens or hundreds of thousands of GPUs these days.

00:05:25 But pre-training is only a first step because immediately after pre-training models are actually a bit weldy. If you take really pure pre-training and you try your model immediately after, it's not going to be very interactive with you. It's not going to be chatty, it's not going to answer your questions necessarily in the way that you expect. I think a failure mode that we used to see a lot immediately after pre training is let's say I ask a model a question and instead of answering the questions, the model will come up with 10 more questions that are similar.

00:05:53 And so reason why is because in pre-training data, this is a likely to have a list of question asked to have the answer following the question. And this led to the development of second phase in model training, which is called post-training. And the idea of post-training is to own in sharpen the model to really fit how it's going to be used, which typically means making it a good chat assistant or something like that. And the methods that you use during post training typically differ. I mean, strictly speaking, you could do post training in the same way you do, but with just data that is specialized, maybe just only transcripts of chats and continue doing pre-training on transcripts, chat only, and you would defacto be doing a post training towards the chat model.

But very often people like the big success of post training has been the use of reinforcement learning. So essentially enabling models to learn not from an explicit demonstration of what they should be doing, which is what supervised engineering on what pre training are, but instead from a feedback about how are they doing? So the model generates an answer and then from a human from another model or from many different possibilities, the model gets a feedback of this is good, this is bad, and just based on this positive or negative signal, the model learns to improve.

Jon Krohn: 00:07:08 So this is the experience that a lot of us will have had in chat GPT where there's a thumbs up or a thumbs down that you can click after you get a response and that can then be used as a training data for this post-training phase, and that'd be reinforcement learning from human feedback RLHF.

Julien Launay: 00:07:23 Yeah, from a very, very high level point of view, this is an example of the sort of data you could be leveraging to power this phase of post-training. I think what's really interesting is right now I'm giving a description where pre-training and post-training are very separate things, so reality is much less so these days first because now pre-training is very dynamic where you shift the data distribution, so you might start with the lower quality data, the more bulk data, and as you advance through steps of pre-training, you will focus more on higher quality data, maybe more code, more mathematics, more. It could be many more chat data, more of the higher stuff that you consider high quality. I put quality quotes because the definition of quality is more subject that we could spend hard on and post-training itself. Even now people are starting to do reinforcement learning during the pre-training step or starting at some point where they start to incorporate mixed plans. The two, it used to be that post-training was a much smaller spend and

pre-training, most of the money used to go to pre-training and to the millions, tens of millions, hundreds of millions of dollars used to go there. But now if you look at recent papers like qmi or even GR four, not really paper, but more something that they mentioned, which is that they spent nearly as much on post-training as pre-training. So there's massive scaling up of this post-training phase.

Jon Krohn: 00:08:47 Yeah, exactly. That seems to have allowed grok four, for example, to be able to get the highest score yet on humanity's last exam, at least at the time of you and me recording this,

Julien Launay: 00:08:57 Which might change in a week with another model coming out, it's always moving. But yes, definitely. I think one of the reasons ROCK four has been so impressive on many of the benchmarks is a larger part of post training that goes into its larger focus on reinforcement learning, but I would say obviously props to the GR team for being some of the first to put out this sort of artifacts, but I think there is a lot more coming. I think that shift has been happening beyond the shadows for a while and now is getting fully, fully executed. And I think most of the model models are going to be going through much more extensive than pre-training. And part of the reason why is also because post-training data is, if you think about it, I don't want say more plentiful because it's a very complicated subject, but is you can generate new data and new problems that the model is going to solve.

00:09:51 When I was mentioning the feedback before, the thumbs up, thumbs down, a big trend that people are probably aware of with dip with Dipe R one and was verifiable rewards where essentially the model solves the mathematics problem or submits an mathematics problem and then that answers get evaluated and if it's right, that's a positive signal. If it's wrong, that's a negative signal. And obviously these sort of things are

very scalable. Mathematics problem code, like code problems, same thing like tests for codes, you could use that as a signal. So it's very easy to imagine, for instance, mining all of the GitHub repositories that are available, pulling all of the tests from them, having the model right code that needs to pass this test and using that as a signal at a very large scale. And this is something that frontal labs do and it's extremely effective. So there's a plurality of signal that you can use that is massive and I think now people are very focused on scaling this massive environment in which to run the models to get these signals.

Jon Krohn: 00:10:49 Very cool. And so we've talked about reinforcement learning now in this post-training step, and we've talked about RL HF where you have human feedback like the thumbs up, thumbs down, what other kinds of reinforcement learning approaches are out

Julien Launay: 00:11:01 There? Yeah, so totally. So there is historically the big one, the big first one, the big acronym that caught up a lot was LLHF, which is reinforcement learning from human feedback where you are using your typically annotators company like scale recently semi acquired I guess by meta companies like Surge as well, which has been the news a lot, essentially having annotators give this up times down or different forms of feedback. Obviously human data is only so much scalable at some point having armies of people annotating data is not an infinite source or something that really is desirable on getting the model to be more competent. So people are started looking for years into ways to get better signals. So one of them was what we just mentioned, verifiable rewards. So some people could this LVF or you see also a lot in the literature l ef.

00:11:52 So from execution feedback because you are executing what the model is producing, testing the result in an

environment, looking at that result and being like, okay, based on that I'm giving a reward or not. And by the way, this execution feedback if you think about it is if you go back to the roots of reinforcement learning when people used to do Alpha go or even before the Atari games, this is essentially execution feedback as the model plays a game, if it succeeds at the game, then it gets a reward. So it's a much more classical setting actually if you think about it in some way, that's the second category, all of this verifiable reward or execution feedback. But obviously not everything is verifiable. Actually a lot of tasks that we do with the models are not necessarily strictly speaking, verifiable. If you think,

Jon Krohn: 00:12:41 I think one of the reasons why Atari was such a great place to start was because of how verifiable it was because point scores that you're trying to optimize that that's a very clear reward function.

Julien Launay: 00:12:49 A game is very obvious. A game like Alpha Go, it's very obvious at the end if you win, you lose or if it's a tie, but there are many tasks that are not like this. Maybe it could be writing a report or it could be pretty much a lot of natural language task. And this is where I think in terms of scalability and access that has been really, really successful as well is LLIF where you use AI feedback. So feedback from another model and it's kind of this observation, which in insight I think it is very funny because in insight it's very obvious all of these things when you look at them in insight, you're like, oh, it's obvious. But when they were getting started, it's like wow, it's magic that it works at all, which is to use another model to review the output.

00:13:29 So for instance saying, oh, let's say that you're doing summarization, very basic task, but is a summary that's been generated factual, does it stick to the fact of the original text? Is it formatted in the right way? Is it all of

this kind of plurality of things you would want to see out of your summary? And what's really interesting is that this is obviously very scalable because this from another model and you can run models infinitely as many times as you want. And so right now this sort of AI feedback, which some people also bundle up into the idea of synthetic data of training based on data that is produced by other models also we're seeing a very big, very big growth and very big success because it works so well and it's obviously very, it's a great way to scale beyond just having the thumbs up, thumbs down from expert originators to potentially reserving the human for the much more expert stuff and then having a baseline from other models, the model which might be specialized and also having the verifiable rewards as for our task that can be verified, big mix of everything.

Jon Krohn: 00:14:39 About a year ago I did experiments internally at a company that I worked at where we had a very specialized task and it was enormously painful for humans experts to review that. It took them so long and they expressed real disdain for having to do this task because it was so challenging. And so we thought, well, what if we could use at that time GPT four instead of the humans? And so we needed the humans to do enough that we had a sample that we could compare and GPT-4, they were comparable. It was the same quality of results were indistinguishable. And so we were like, perfect, this means we can now scale up to as many samples as we want.

Julien Launay: 00:15:21 Yeah, yeah, totally. And I think this is actually very interesting that you mentioned with especially comparison with human is that a lot of people have pushed back on, well, I would say synthetic data as a whole, but on data that is model generated because they're like, oh, this is going to be degenerate data. This is going to fold on collapse into something that's bad. And

that's possible. You can do this sort of data the wrong way and you can completely mess it up as always, but in general, it actually works really well and I think people have this idea of human data as being very perfect, but actually if you look at the data that comes out of the typical annotation contract, and I won't cite any, but it's actually not necessarily the quality that you think it is, it takes a lot of review to get it right.

00:16:03 There's a lot of issues and there are plenty of studies on what people call inter rate agreement rate, which is how much if you submit to two different annotators how much they agree in the rating, maybe if it's a rating on a like RT scale from one to seven or if it just thumbs up, thumbs down. And the numbers obviously are very task dependent, but when you see them, they're actually crazy. Actually it's a lot of noise. There is a massive amount of noise and when you measure actually the same sort of agreement rate with models or between models and humans, you see actually numbers that line up where essentially the quality that comes out of the model is as good as what comes out of sensors. Obviously not true of every tasks are task, if the judge model is completely capable, judge model is obviously not going to be good at this, but there is is another side to this con, which is that verification is much easier than generation, so it's much easier for a model, a posterior to come and to check your results than it is to produce it. And that's something that's very powerful and that's probably one of the foundation of why this works so well.

Jon Krohn: 00:17:07 Nice. And now this next question is going to get relatively technical, but our technical episodes are some of our most popular ones, so we're going to dig into it a little bit here and then after that we'll get back to more applications. We'll talk about your company and we'll talk about Adaptive. But really quickly, I want to get into something really technical here. So these different kinds

of approaches, you talked about RLHF where we have the human giving a thumbs up, thumbs down, RL eef where this execution feedback, where there's something kind of innate about what we're evaluating, like an Atari top score that we're trying to reach for or R-L-A-I-F, which we talked about most recently where you're using AI models to kind of give you a thumbs up, thumbs down with an AI system, regardless which kind of those approaches we choose, there's also differences in what reinforcement learning algorithm we select, right? So there's things like POA two C, do you want to tell us about the big ones there?

Julien Launay: 00:17:57 Yeah, yeah, totally. So obviously H-F-I-F-E-F is essentially is changing the data on which you are training, but you could also change a method. We keep talking about this LL, but what is reinforcement learning on a very fundamental basis There is what makes reinforcement learning so different. It's not really, it's a spectrum like everything from SFT to reinforcement learning, you can kind of build step by step and there's really a spectrum of things. The moment at which it exactly becomes reinforcement learning might be

Jon Krohn: 00:18:26 Sft, supervised fine tuning,

Julien Launay: 00:18:27 Supervised fine tuning. Yes, totally. It's like the moment at which the transition might be a bit of a question of where everyone puts it, but it's a very, very general idea. I think the key components to reinforcement learning and then we can go into how it goes into different methods. I think a big first thing is that reinforcement learning typically so will be online. There is a difference in literature between online or offline LL, this is actually one of the very big, let's say theoretical, and I put this in quotes for material. There can be in machine learning, well especially concerning lms, but historically people have argued a lot about what is offline on online and

what does it mean to be offline or online? Well, essentially online means that you are learning based on the sample you just produced. So let's say I have a set of weights of my model, I make an inference, I get an answer to the question, I evaluate that answer or saying, oh, this is good or bad in whatever of the three ways we mentioned before.

00:19:25 And then I use that in the training process to say, okay, so now I update my weight based on that thumbs up, thumbs down, but I do it with fresh data, with data that has just come off the price and then I repeat that process, I update the weight of the model and then I get a new sample. Now what if I accumulate sample and then I train the model a few times and then I start collecting samples again As soon as I'm doing my, I do my first step of training, I'm online, but as soon as I do the second one, I'm not online anymore because the data I've generated doesn't come from the same set of weight. It come from a set of weight that existed before and that hasn't actually produced the final output. Obviously this is a bit of a ship of C kind of thing where one step might be okay, might still be more or less the same thing, but two step is it really the same models, three step, five step and steps and then you veer into aline.

00:20:16 But one of the big success of reinforcement learning is that it is mostly online, but it's mostly proximally online where essentially you have samples that are relatively fresh, you evaluate the samples and you learn from that. And this means something right now when I say this, it sounds very abstract, but actually there is a very, I think easy analogy to say to this is that the samples come from the model and the model gives a suggestion of what it can do and you tell it if it's good or not. If you think we make a parallel with human learning and let's say I'm teaching you a course about general relativity and I'm teaching you something about spinning black holes, care metrics, that

sort of stuff, I could, if I show you an exercise to do and I could show you the solution of the exercise, have you memorize it, just memorize it again and again and again, and then when I present you the exercise, you can run through it exactly the same.

00:21:10 Again, this is essentially what pre-training or supervised, fine tuning do where you are presenting to the model maybe once, maybe twice, maybe it's twice the same samples and the model eventually learns from it. Obviously pre-training still generalizes because pre-training is very diverse In pre-training. I don't just show you one problem, I show you all of the problems that can exist and there is that expectation. But in post-training, if I just show you one exercise and now I tweak something in the exercise, now I say, oh, now actually the black hole is carrying a charge. And so now you are in a completely different setting, you have no idea what to do. You are going to look at me going to reproduce your answer and it's going to be bad. If we were doing reinforcement learning, the way that it'll work is that you would try to do the exercise and then as a teacher I would correct it and I would tell you, okay, this is good, this is not good.

00:22:00 Kind of that iterative process. And this is really fundamentally I think a good mental framework for the difference between supervised venting and reinforcement learning. And that comes to this unlikeness, which is that in reinforcement learning, the samples come from the model itself. So it's always in distribution for the model come within what it's capable to do, and then slowly you are shaping that distribution away towards what you want it to be. Whereas in supervised fine tuning, you are kind of propping down the new distribution, which might be very out of distribution, and you are hoping that as you show more and more simple that are diverse enough, you are going to widen the distribution and hopefully

connect it back to the original knowledge that there is no gap in between. Because if you ask a question that's in between what the model used to know and what you have told the model, well there is no guarantee that you fold that in between.

00:22:52 It's covered. It's something it has learned. So it's kind of like difference between the two, and I think this is a very big aspect of reinforcement learning is that online that learning from trial and error, that part actually touches to the second point, which is something you can somewhat simulate with supervised functioning by filtering the data, but it's also that reinforcement learning learns from a much wider range of signals. So instead of learning from an explicit or you need to imitate that, it's about, oh, this was good, this was good, this was bad, this was maybe okay-ish. There is kind of a ity to this. You can bring this as well a bit to SFT in some ways, but I think it's a big difference at learning from a reward essentially from positive negative things.

Jon Krohn: 00:23:32 I love this example that you just gave talking about teaching me general relativity and how the supervised fine tuning is kind of memorizing a solution and the reinforcement learnings this online way of learning where as I am producing my output, you're providing me feedback and nudging me in the right direction.

Julien Launay: 00:23:51 Totally. And I think it's a very good analogy because I actually think it's quite true to what happens. There is a caveat, obviously an virtual argument to this, and I touch it a bit on it, but I think it worth double clicking. It's like, oh, but pre-training works very clearly. Pre-training works at teaching the model. Many things that, so why it is it's a question of scale is that in pre-training you are showing not just one exercise, but all exercises that are possible in post-training often, I mean you can somewhat afford to do this. We discuss this before post-training is

becoming wider and wider, but when you're specializing a model, you want to be as effective as possible with this to learn as much as possible from every sample that you have. And you might not have the luxury of every cases that is possible that you have in pre-training.

00:24:35 So there is kind of slightly different regime. I put a bit of a star on this because now people run post training at a much larger scale and it works and well has its benefits. So this has a bit of a caveat here, but the fundamental idea of much more generalization from reinforcement learning because of this onlines, because of this trial and error and all of that, I think is very fundamental. And actually when thinking about reinforcement learning research I think is one of the big thing to think about. And to go back to your general question on the different algorithm, so we are a lot like HC is an older one, but we are a lot these days for instance about P-P-O-G-R-P-O-D-P-O, all of this sort of stuff. I think one of they are actually quite similar. The answer is especially PPO and G rpo, there was a big debate in the community, PPO and GRPO in term of these components on onlines and everything share very similar characteristic.

00:25:24 What they do differently is more in the question of how then do you attribute you have a reward? So I tell you I passed the exercise or you failed the exercise, and there's a question of how do you attribute this to individual steps in the exercise or to individual parts of the messages? And so PPO does this through something that's called a model that calculates an advantage. So the value model that is going to try to go back from a reward, which is sparse in the level of the tokens you have, some tokens are rewarded but not others. Very often might be just a final token, but sometime it might be a bit more dense to a reward that is to value. That's at each token this contributed that much or that. So in PPO, you are training model to do this. Literally you are training a large

language model to do this task, which is an interesting view. In DP, it's a bit different. You essentially do another range of multiple rollouts, but fundamentally this is just a different way to attribute the blame. The fundamental of the methods are still very, very, very, very similar and a lot of similar ideas.

Jon Krohn: 00:26:35 Nice. Very cool. So with that kind of context, that kind of foundation in mind now including getting into the detail a bit of algorithms like P-P-O-G-R, PO and A two C, let's talk about Adaptive. So I mentioned right at the top of the episode something called Adaptive engine, the flywheel for enterprise ai, which is continuously evaluating, tuning and serving LLMs uniquely adapted to a business. Tell us more about that.

Julien Launay: 00:27:01 Yeah, so I think at Adaptive our motivation has been that reinforcement learning is amazing. All of these methods are really amazing and this is even more obvious nowadays I would say in the past few months, but when we got started a year and a half, two years ago, I think it was still true. If you had that experience of going, you are like, oh, these are amazing methods. They can do amazing things and there's clearly a lot of potential in them, but there is a bit of a problem which is that typically they're quite difficult to put in place.

00:27:31 You might remember from reinforcement learning days, AlphaGo or Atari that we mentioned before, and you might also remember that this were very challenging to get right. There was a lot of research around it. Very often, a lot if you have studied ML in a bit more formal setting at school a often seems a bit opac and also when you have experimented with it very often depends on the seed, on the initialization, all of that. It's not as bad with L lms but with LLM, the is more on the engineering because firstly you are going to be blending inference and training because as we mentioned before, we are teaching

the model based on something that has produced. So we are going to have some time to do rollouts or to do predictions and then rate this prediction and then use that as training. So there is not as much as before this dichotomy between all I serve my LLM to millions of users and I train my LM on this cluster.

00:28:18 Now there is a bit more of a combine of the two which poses engineering challenge. Obviously the other aspect of this is that these are complex pipelines. So typically we mention hf, A IF, VF or EF depending on how you want to call it. So this means that during training the model is going to have to interact maybe not with humans because maybe this is something that you will put offline and train reward model, but it'll have to interact with other models, maybe 2, 3, 5, 10 of them. Like today we run pipelines which are like five to 10 AI judges in them and it works perfectly fine, but also environments. So maybe you are going to teach a model to do text two sql, you do this well, you have to run the queries on the S QL database to get the answer to be able to do exchange feedback.

00:29:06 Maybe you teach a model to do rest and so you need to have a REST compiler and maybe you need are teaching the model to use and so you need access to these tools or maybe you are touching the model computer use in which case you need a VM box, like a virtual machine in a box where the OS is running and where the model can go through, oh, I do book click on per point, I open this. So you get all of these things and now suddenly engineering becomes a nightmare. One of the reason pre-training scaled so fast is because pre-training is very simple. It's very straightforward. You have this huge batch of tokens you just predict, predict just the logics, you don't even actually sample. So you just predict the logic, you compare them, you compare the top one to what was in the text and that's it.

00:29:49 So it's very, very easy to make it run at impossibly large scale because fundamentally then yes, there are engineering challenge to distribute computing obviously, but fundamentally the algorithm is very simple, very limited in interaction with external world in code. Whereas with reinforcement learning now you have all of these environments, all of these other models that you need to interact with. The motivation at Adaptive is actually to make all of this easy, we're like we think that reinforcement learning is the way to get the best performance out of a given model for a specific task. We think if you think about it from a par of frontier point of view, reinforcement learning will always get you the best cost to performance compromise always it's like a new par of frontier. So obviously this is very attractive for enterprise adopting AI because either they want a cheaper model, same level of performance, but they want something that runs as efficiently as possible or maybe they want something that's not possible now and so they want more performance reinforcement learning in both cases. Easy answer to get there, but the question is doing this reinforcement learning and so this is what we do at Adaptive, we provide essentially data sense teams with what we call the ops tooling for

Jon Krohn: 00:30:59 Them. Reinforcement learning ops.

Julien Launay: 00:31:01 Exactly. So tooling does, they need to make this super easy and so that they don't have to worry about all of this distribution, this interaction, but they can just focus on the logic. So they can just focus on like, oh, I want this judge that does X, Y, Z. Maybe there is 3, 4, 5, 10, 15 of them. On top of that, I also want an environment in which something gets checked. I want this, I want X, put all of this together, some synthetic data generation as well, a lot of lack of these things and then you don't have to worry about any of the actual implementation or tooling essentially does kind of, I don't want say compile

because that's exactly compilation, but essentially interprets your instruction, your Python recipe and then runs it on the cluster in a distributed way without you having to think about it. That's fundamentally what we do.

Jon Krohn: 00:31:45 So I guess the target, correct me if I'm wrong, but it sounds like on what you're saying so far the current Adaptive platform is designed for people like software developers, ML engineers who want to have reinforcement learning be easier. So your target audience is probably a lot like my listeners in general where they're people who are writing say Python code.

Julien Launay: 00:32:07 Yeah, totally. Absolutely. And most of our users are data scientists will write Python codes to interface with a system. That's totally the target. If we get into the details of the business, it's always obviously a bit more nuance nuance than this. We also sometime work with companies that have much less technical expertise where they don't have a data science team, they still want to achieve something. So maybe either we will do some of the work or we have partners like Deloitte that can do some of that work. So there is always a bit more complexity business wise, but yeah, fundamentally our core idea is to build better tooling for reinforcement learning so that it's easier to get to value and obviously this something right for data scientists.

Jon Krohn: 00:32:48 Nice. And so if we have a listener today that wants to get started with Adaptive, what is that journey like?

Julien Launay: 00:32:54 Yeah, so we are still very enterprise focused. I think when we started our company, which is a bit over a year and a half ago now, one of our thesis was to be very focused on enterprise, on larger enterprise because these companies, when they generate into production, they have a very unique scale, maybe millions, tens, hundreds of million of

users and that comes with obviously costs. There are much larger, some bigger impetus to potentially optimize costs to pack into smaller models. But although that means that you have much more interactions with the model and we say more interactions means more data points for post training. So part of our original thesis was to focus on this sort of company, which mean that right now a lot of our deployments are deploying in our customer infrastructure. We don't really have a cloud available yet, but you can come in with your mom's credit card and just get started. Well not your mom's credit card, your credit card or your company credit card, but this is something that is coming very soon where we want to make this more available. We think also now our tooling is a lot more mature and could be put into more. So the short answer is that at the moment there is, we don't have an immediate general availability of that.

Jon Krohn: 00:33:58 I gotcha. For the, so sounding wants to take advantage of being able to do reinforcement learning more easily. They reach out to a sales team from the Adaptive website. Right

Julien Launay: 00:34:07 Now we are very enterprise focused, which means essentially reach out to a human,

00:34:12 But we are also excited to change that actually in the future and to make our technology more boldly available because we think it is at a point where that can be the case and where everyone should be able to run this sort of stuff. Even for OB projects, actually I think there is, even for this try something, maybe it's really good, maybe you're able to build a model that ends up being much better, much better than what currently exists and maybe that can be the next big startup that you create that you get started. It's part of the tooling for this sort of stuff. I think one of the reason we're expanding now to this is I mentioned something about origin thesis was, oh, you

need all of these human data points because when we got started it wasn't as obvious that IEF and synthetic data would work so well.

00:34:57 I think it was definitely something we had in our roadmap. If you actually go back to our fundraising pitch, we had it as the end of first year kind of thing. But what has really positively surprised us is how well it works, how much leverage the synthetic data give you to go from almost nothing to a lot and these kind of removes, there's still value in having all of these production data points. They have tremendous value, you can do a lot with them, but they are not strictly necessary. You can get started, you can get bootstrapped from much less and you can take a very small model, 8 billion parameter model to be to the frontier performance on the task mostly entirely with synthetic data, which is quite incredible and very easy to do. So this is why now we are also thinking of avoid availability in a way because actually the synthetic data pipelines, anyone can run them, anyone can define them. You don't need all of these users already to take the benefits of that.

Jon Krohn: 00:35:52 Yeah, we had the same example that I was giving earlier where we evaluated, we tested the interrater reliability like you mentioned earlier, between the AI model evaluation and the laborious tedious human evaluation. That was actually for the purpose of what you just said, which was fine tuning an 8 billion parameter model, one that can fit on a single relatively inexpensive GPU and get frontier model performance on just a small set of tasks using these kinds of approaches.

Julien Launay: 00:36:27 Yeah, totally. It works. It works amazingly well. I think the boom in synthetic data, like the success of synthetic data, and when I say synthetic data by the way, it's a very broad word, which means many things and it means to me it means first problem generation sometimes. So it

means creating new sample, creating new scenarios, maybe new scenarios of conversation self play. So maybe simulating a user, having a model stand in as a user to drive a conversation for self play where you have, that's first category, but it also mean all of the a IF components of giving feedback, of reviewing some of these things. It's quite broad, but I think synthetic data has been widely successful. We have a company we worked with, which I can't name, but essentially they were building a chatbot of quite a general use case and the chatbots, they wanted it to have certain traits, certain psychological traits and that sort of stuff.

00:37:27 And to do this we under mostly using synthetic data and we start from something, I think it's about 70, 80 maybe human ated samples. And when I say annotated in this case, actually I don't mean thumbs down. This is something that you find a lot in this reinforcement learning pipeline. We use critics and rights where essentially a human comes in, looks at something that the model has produced and write a critic and a new version of it. So literally in natural language, this is really cool by the way because I think when you ask someone, especially someone skilled, think of a lawyer or think of a psychologist, someone when you ask them to give thumbs up, thumbs down on thousands of samples, I think it's very, they don't like it. I think they feel a bit in the mid factory, they feel like their work is being ized.

00:38:17 I think they don't like it at all, but when you ask them to give feedback about something or write something, actually they really enjoy it, which is very funny. They feel more I think engaged in the process and they feel more in control. So anyway, so we collected something 70 of his annotations in a bunch of different contexts and from the 70 we are able to generate something like 80,000 synthetic conversations through self play, about 80,000 synthetic conversation. Each of this conversation is made

of about 10 turns. So you are looking at nearly a million message. And during the reinforcement learning process itself, we explore multiple possibilities. So for each of this message we might explore five 10 possible answer which gets rewarded by judges. So at the end you are looking from less than 100 sample had something like nearly 10 million data points from which you can learn. So obviously a massive multiplying effect from these synthetic data pipelines.

Jon Krohn: 00:39:22 So that's been a fascinating journey that you've had us on talking about lots of the reasons why Adaptive makes things easier for us and how we can have more powerful models, have smaller models be able to do things that frontier models might otherwise only be capable of. You've described previously Adaptive as a layer on top of foundation models to tailor them to final use cases. Tell us about this being a layer on top.

Julien Launay: 00:39:51 Yeah, so one thing I want to be very clear that we don't do is starting from scratch, I think specialized model in the sense of starting from something from scratch, I think there are use cases where it might make sense, but I think for a vast majority it doesn't because, so reality is that the foundation models are this amazing engine, this treasure of knowledge and of understanding which you can sharpen into exactly what you want. And I think for us what's really important is why we say we are a layer on top is because we start from open source models. So this might be Lama, Quin, Kimi, whichever one is your favorite flavor and whichever one you're allowed to use at work, we start from them and we tune them to perform better. But this is only possible because the base model is already amazing actually.

00:40:36 And if the base model is not good, you don't really get anywhere. And this is back to the example much earlier or we chatted about reinforcement learning used to be

even harder with issue at initialization with seed and that sort of stuff. And part of the reason is because this was reinforcement learning from scratch. And when you're doing reinforcement learning from scratch behavior initially is fundamentally unstable because you are asking a random policy, like a random model to take decisions, but obviously the decisions are random, which is a disaster. Whereas when you start with a large language model, you are starting on easy mode because the model is already incredibly smart. One thing to note about pre-training is we said, oh, the pre-training model, the pre-training model is not chatty. And by the way, something I would invite people to do, it's harder these days because as I mentioned, the lines between pre-training and person are blur, but it's to look at some of this older pre-trained only model.

00:41:29 I think some of the early LAMA might still be available this way, but you can also look at models like GPTJ that were some of the first very big success of open source models that were just pre-trained and you can try to interact with them, obviously come from another generation, much less compute spent, but still you would see it's very different. But anyway, despite that, this models are still amazing compared to a random starting point. They're still amazing. They still contain insane knowledge. If you think about it in pre-training they've seen nearly everything. The knowledge that's contained in this model is insane. So it's mostly a matter of disentangling that knowledge to an extent. Adding some of it as well, I think there's always a debate is post-training actually adding knowledge to the model or not at all? I think it does as well in southern conditions, but essentially of disentangling the knowledge that's in the model, maybe pruning the part that you don't need as much, so facing the part that you need the most and using that as scaffolding to learn even more, to acquire even more capabilities. But this is possible because we

start from an amazing open model and that's kind of the sense that we are layer on top is that we start from this base open source model and we take them to even better performance.

Jon Krohn: 00:42:43 Nice, great example there. It makes a crystal clear and it's interesting how you have all this background in developing frontier models and now you've found this niche allowing other people to leverage that kind of background that you already have and be able to accelerate their own use case development, particularly at the RL stage.

Julien Launay: 00:43:07 Yeah, yeah, totally. Yeah, I think for us a lot of what we do now is bringing this expertise in reinforcement learning to be something that anyone can do. Our view is that reinforcement learning to an extent is still a bit of a frontier subject. It's still a little bit of something that only a maybe more experienced audience gets to experience, but I think it shouldn't be the case. The reality is that these are exceptionally powerful method that should be in the end of every data scientist of everyone must like prompting is in the end of everyone. I think being able to build these pipelines to leverage them should be in the end of everyone to build something really cool. So ultimately that's really our goal is to spread this and we see it. One thing that I find personally very, very exciting when we work with customers that obviously very often on the first use case, we work very closely with them because we teach their teams how to use the tool, how to think differently as well because even teams that have experienced with supervised functioning, I think reinforcement learning asks you to think differently.

00:44:05 You don't think so much about the data that's going to be the explicit demonstration, but you think more about measurement of success, so you think more about what defines success and how do I measure it. So that might

be something that's verifiable, that may be with an AI judge and then you use that to tune over. It's a bit of a different way I think to think, but yeah, typically we work very closely with 'em on subject and then their teams kind of take it on. We have a customer can name publicly because they appear with that. We work a lot with at t in the US and at t their teams now are using the tool autonomously and it's always amazing when in the sessions that we hold with them, they come and they're like, oh, it and V, and you see the scores and it's really good and you're like, oh, it's amazing you tool. I think it's really fun for data scientists to be able to have this new capabilities to do more

Jon Krohn: 00:44:53 Nice that the teams there at t are all grown up now on your trading.

Julien Launay: 00:44:57 They are really good. Actually, I don't just say this because I can talk about them publicly. I think one of the things that's been really cool in working with at t is the maturity in terms of bringing gene AI to use cases. And when you look, whenever we have meeting with them, I'm always surprised by the penetration of generative AI inside the organization and everywhere in every aspect of the business, they are pushing models to do really amazing things that really create value for the company. And so I think it's really cool to see that because often there's always a discussion of oh, is AI bubble is blah, blah, blah grumpy people. And sometime you might be like, is it whatever? And I think it's definitely not. I think yes, some business are slower in adoption, so real world is always slower in adoption, but there is in companies that are moving forward, insane value being created.

Jon Krohn: 00:45:49 You mentioned there's something that I want to highlight a little bit that these amazing teams that at and t are doing is they're getting that definition of the reward function. That sounds like it's one of the hardest parts.

So you're getting people's mindsets on the reinforcement learning cycle. And if you don't define that reward function, right, your model isn't going to end up doing in production what you hoped it would.

Julien Launay: 00:46:08 Yeah, totally. So I think this is the part where it gets in a different way to think about these problems that in reinforcement learning fundamentally what you're thinking about is what defines success? How do I define a successful outcome or bad outcome for the model and how do I provide the model a signal about this? And that's really what it becomes all about. There's this quote that I really like for enforcement to describe enforcement learning, which is that if you can measure it, you can optimize it. And this is literally true, actually this is one very cheesy, but it's actually literally true in the case of reinforcement learning, which is that as soon as you can measure something, you can use it as a reward to optimize it. And because these methods are so powerful, because this models that we're using are so smart, even if the signal is noisy, even if the signal is kind of removed quite complex and all that, the models are going to find ways the system is going to find a way to optimize for it.

00:47:02 And that's uniquely powerful. So then it becomes entirely a game of how do I define it? And this is a part where reinforcement learning becomes more of like, I like to describe it as a pipeline or as a system because it's not very often success is multifaceted. There is not just one criteria. So it might be somebody that needs to behave in this way, needs to follow these policies, it needs to achieve that x, y, and then it's about finding the signals or even in multi-agent system finding the signals for individual agents of like, okay, these define success as this step. I can check it, I can evaluate it maybe with another model that's a reward, that's good. Then I move to the next step and kind of building these things and

also being able to do it end-to-end as well where ultimately there might be an overall success.

Jon Krohn: 00:47:48 So it's clear that you have a ton of experience. You and the Adaptive team have ton of experience with getting real world use cases spun up, particularly leveraging the RL ops that you guys specialize in at Adaptive. I now have a long question.

00:48:04 There's a lot of context here, so I hope you have a big context window zoom as well as our listeners. It's 1 million because I'm going to dig into a bit about your past prior to what you're doing at Adaptive, but then I'm going to use that to talk about how we can be preparing for the future. So you previously worked as an extreme scale team lead at the AI community hugging face, and prior to that at the Gen AI platform light on, we'll talk, I have another question about Lighton coming up soon and there's a big question mark around scale these days where the idea of bigger compute, bigger networks, bigger data driving, more model capability, one of those things, more compute. Okay, we can just have, you talked about hundreds of thousands of GPUs, you can have a million. Theoretically it is an engineering problem. Same thing with bigger networks.

00:48:57 We can have more model weights or we can have more clever mixture of experts models, but bigger data can be tricky. You already talked about earlier in this episode how the pre-training can involve all the literature that's ever existed, all of the internet. So that can theoretically run into short supply. And so different people have different opinions. Ilias said that if gen AI's fossil fuel is human data on the open internet, we've exhausted our supply. However, other people like Sam Altman, Dario Amodei from Satya Nadella, from OpenAI, Anthropic and Microsoft respectively, they don't seem to think it's a problem that scaling has no end in sight. Synthetic data

seems to be part of the solution there. Do you think that engineering tricks, you've mentioned in past interviews how things like cleaning up data to remove duplicates had a big impact. So do you think that this kind of massaging the data that we have can continue to give us great results going forward regardless of whether we have more of it?

Julien Launay: 00:49:56 I think there is a bit of truth in every one of the statements where definitely in term of readily available data, we are starting to eat the limits where there was a golden age where we are just starting to call the web and starting to improve your color and then you add archive paper and then there was kind of golden age where data sim unlimited and obviously no way is less the case. You can message this data to improve its quality to get better results out of what you get. You can order it differently. You can order it differently during pre-training, maybe put the lower quality data first that you get more impact from the later high quality data. But ultimately at some point we only as humans, we have only produced so many words. So there might be a question of this of do we run out?

00:50:42 I think actually I can't answer the question, have we run out now or when I think it's actually quite complex, but I think I can answer the question of what's next and what's already actually the case, which is we go back to post training where what's very interesting about post training, that post training enables model, and I'm going to use the word of certain because you put in a very elegant way, it enables model to learn from experience. So the models actually do something as we discussed before, get the feedback on that doing and receive that. And this is infinitely scalable because this is essentially the experience of the model in the real world. Yes, currently we do this in simulators, we do this, we formulate this artificial problems, but it's already possible for models to

conduct an experiment in the real world and use that as a reward signal for reinforcement learning.

00:51:32 And the bits of data we can get from this, I mean no, there's no limit. It's practically models can conduct as many trials and many experience in the real world as resources alone. So I think this is a part, if you look from a more very large scaling perspective, this is where enforcement learning is very exciting for this is that actually it's a gateway to a much wider range of signals, much wider ability to learn. And we are seeing this now for a while people were seeing reinforcement learning more as like, oh, this only specialization layer or only post-training layer, but actually it can be the bulk of the resources are going to be spent in the future on post-training, on reinforcement learning because the models are going to learn from trying again and again across billions of virtual environments and eventually also in the real world against trying their experiments, their own ideas. And this is obviously a very big frontier right now and people are pushing really hard on it and it's really, really exciting.

Jon Krohn: 00:52:37 Nice. So that kind of covers the data problem. It sounds like we're good on that front. Let's talk a little bit about compute as well, and this gets into your experience of lighton, which I think is interesting. So land use and power requirements of data centers are getting more and more ambitious. So Mark Zuckerberg recently announced several multi gigawatt clusters including a five gigawatt data center, which would cover more than three quarters of the area of Manhattan where we're recording today in a presentation a few years ago. In 2021 you said that by this year, by 2025, hardware would become the bottleneck. And so you discussed how things like light ons, photonic chips have these kinds of hardware, alternative hardware approaches can be the solution. So

maybe neuromorphic chips or photonics. What do you think about this hardware problem?

Julien Launay: 00:53:26 Yeah, so for context, I used to work when I started, when I did my PhD in France, you can do an industrial PhD where you work with a company, very good system, very surprising that it wasn't invented in the US of all places. But this company lighter on now, which now mostly does gen ai but used to develop a chip which worked with photons instead of electrons or with light essentially to do computation to certain type of computation and viscom with a bunch of advantages like such on poor consumption, on parallelism, on things you can do. So it's alternative means of computation sometime related to neuromorphic, that sort of stuff. And yes, so today is the bottleneck is compute definitely. I think this is very obvious given the money that people are spending towards trying to get more, given the insane valuation of Nvidia, which is going to continue to increase.

00:54:14 So the bottleneck is compute. Obviously we might ask do we need a new compute padding? This is actually a subject on which I'm very bearish personally and which I have, and maybe this is because I got burnt once and so I think it's really difficult to bring a new hardware padding to life, the current hardware padding as its issues. It's use a shitload of energy, blah, blah, blah. It's very rigid, but it also has tremendous advantages in that it works really well. You can implement this algorithm very effectively. It works really well and there's a lot of money that goes into it. If you think about the latest cheap home Nvidia, if you think about the GB 200 or the full rack, now it's an entire rack, but if you think even about just the chip, it's probably the most complex object that has ever been built by humankind.

00:55:07 If you hold one in your hands, even an H 100 or B 200, you are probably holding the sum total of all of human

achievement. Like all of human achievement has picked to this thing, which is absolutely insane in terms of engineering to get there at nvidia, it take a decade to build a generation of ship from tion to implementing some of the r and d that is coming out of T-S-M-C-S ml, others to actually building the compilers for this chip, the first step outs and all this over a decade of building and account of the r and d that goes behind into extreme and all of this insane chain of technology for a competitor to come for n alterna mean of competing to come. Well, you have to reproduce all of that and I think this is going to take a while. I think the reality is that this is really hard to get at.

00:55:55 I think all current silicon based padding, more current padding of computing is really good. Is there room for improvement for more specialized hardware for that sort of stuff? Yes, and to an extent the GPUs are already extensively specialized. They're not really GPUs anymore. They're already extensively specialized to machine learning and even people, some people say, oh, it need to be specialized to transformer, but this is already happening. If you look into the instruction sets that you have on these GPOs operations that are increasingly becoming specialized to this, they're thinking about or adding some specific instance in the attention. You have the soft max which has an pronunciation phase, so our instruction set for this, they're thinking about or can we increase a bit on the chip? So part that is dedicated to this so that we get a bit more stupid with this, it's better.

00:56:43 So there's already all of this already goes into thinking at Nvidia, so I think there is already that specialization motion is moving. I will bring another point of view to this, which is something I've been thinking about increasingly recently when chatting with France of, oh, what do you think of? Should we change tokenization? Do we need photonic computing, quantum computing for

that? I think about it in term of do we need it to get to AGI slash ai? Is it something we're going to discover by ourself that we need to figure out by ourself to get there, or is it something that later we are going to figure out with the support of general intelligent or super intelligent system Because I think general inte system probably already exists or it be something that we're going to figure it out with these systems and my thinking on the subjects of photonic chips or even quantum computing that we as humans don't really need to worry about this right now. I think that we already have the capability, what we have, the technology that we have are already in us to take us to the level where we will build system that will help us build this. I'm sure in a century I'm sure we will use photonic chip. I'm sure we will use quantum chips and all of that, but we will have built them with the help of what we're creating currently.

Jon Krohn: 00:57:57 Yeah, so basically to kind of summarize your big idea there, we can use these chips that were originally designed for graphics processing and we can leverage those at huge scale to create an AI system so powerful that it helps us,

Julien Launay: 00:58:12 That it'll help us

Jon Krohn: 00:58:12 To crack all these other things.

Julien Launay: 00:58:13 Exactly. That will help us do scientific research and everything. And a lot of the way that I think about these questions these days, what are the innovation that we still need to do to bootstrap the system that will then help us to get even more? And there is stuff left to do. This is not a negative point of view. There is nothing left to do. No, there is stuff left to do. I think we're thinking of reinforcement learning just before. There's plenty of stuff to do in that direction of how do we scale this? How do we enable models to experiment in the real world At some

point there is obviously a bottleneck in the real world if you are a material scientist or a biologist, you conduct experiment in the real world. You don't just sit at your laptop all day. Models currently cannot do this. There is no way currently for a model to run a biological chart in a scalable way. So there is no way for a model to run.

Jon Krohn: 00:58:59 There's a tiny bit of prototyping in that space where you have on relatively small scales an AI system that can control a wet

Julien Launay: 00:59:08 Lab. Exactly. It's starting. Yeah, exactly. With weight labs or even for material science

00:59:13 And it's starting, but now people I think want to scale this because this is one of the next bottleneck is how do we enable models to run experiments in a scalable way in the real world. It's super exciting. I find the first early experiments in this that you mentioned to be really key, I think how do we scale this? How do we make a weight lab, a material science lab or whatever else, something that's addressable to a model can be easily reset, that can be easily experimented with in a safe way. I think these are really big challenges. So these are subject that I think for instance, we'll need still a lot of innovation and will be key.

Jon Krohn: 00:59:50 So it seems like you spent a lot of time thinking about these powerful AI systems. We might call them artificial general intelligence if it's kind of at our level or above us, artificial super intelligence and helping us with these kinds of problems, handling material sciences problems, biological problems. And so this is something in my mind, I'd love to hear what you think about this. In my mind, it's always seemed to me like having an AI system, having this a GI kind of system. It isn't the singularity that we can't really see beyond that unleashes. Yes, things will be very different, but some things will still take a lot of time.

It's not like instantly overnight cancer is solved because you have to run experiments on probably humans and tissues and other animals and that could take decades. So you can have hunches, the AI might be able to have insights by taking papers from all different kinds of fields and having insights that humans might not have had. But then we still need to run the experiments and those could take a long time.

Julien Launay: 01:00:57 Yeah, totally. Yeah. I think one of the aspects is that I think the super intelligence that will create will at first be very spiky. There will be domains where they will be disproportionately superintelligent compared to other, for instance, I think mathematics is a really good example of this where we can build formal verification system, we can build all of this. So getting to mathematical super intelligence can happen in a box, literally can happen in a completely closed box. You could build a super intelligence in terms of mathematics, building a biology, super intelligence. Some people have a different view of this. Some people think that computer driven biology simulation and everything will be in us, but some others, the state of the science at the moment is that you need experiments. And so we might build a system that is beyond genius developed mathematics that can describe mathematics.

01:01:46 That's way beyond our ability to understand. But at the same time, if you ask it to do even the simplest of medicine development might not be that amazing or the bottleneck might be really good at making up the plan at updating you on the plan once you give it the result. And it's going to be like, okay, so now we should try this, blah, blah, blah, but still be bottleneck. So totally. I think it's a very realistic feature. I think something that's very clear I think now is that the closer we get, I think it'll definitely happen. I think we'll definitely build within probably the next five years. That's my personal bet, but maybe even

10 years for if you're a bit more British, we'll build super agent system, but this super agent system will not be super agent in everything out of the box.

01:02:28 I think they should be very messy actually. I think it'll be a very messy time because in some domains we will do more progress probably in the space of a year than we have done in the space of all of the existence of our civilization, which will be astonishing like discovery that we can barely imagine. And in some others we'll barely move. In some others it'll be like, oh, a new flu come around, well still have to do the work to come up with a vaccine for this year because the system doesn't do that yet. So I think it'll be very messy. I think it's one of the nuance that I would bring to the stories that you often read about fast takeoff and everything is the messiness of it. And it's very hard to predict which direction is going to work so well, which is not maybe some of them. Yes, indeed simulation will work very well for some things and maybe for something we'll be able to do tremendous progress just in the box without going to experiments. And maybe in some other fields we will disparately need the experiments to be able to make forward for us. I think it'll be very, very integral, very messy in many ways.

Jon Krohn: 01:03:25 Fascinating. And so it sounds like with me, this super intelligence system that we are careening towards in five to 10 years, in your view, it is largely a positive thing for humankind.

Julien Launay: 01:03:38 This is a very complex topic. I'm personally, maybe I'm a more optimistic person. I personally think yes, it's very positive. I think I view scientific progress. I view progress in general as one of the main driver of what we do. I think it's a view that maybe not shared by everyone, but personally I think it's some of the most beautiful achievement of human gun is progress and understanding of our universe. So I think not only being

able to create intelligence, to understand intelligence, to create it understanding might come after creating it, which is kind of funny, but being able to do this is really beautiful. I think it's amazing. It's something amazing that we are doing and I personally think it'll be positive. I think that, so it'll be challenges, will there, will it create big societal change that might create unrest and everything? Yes, that's very likely.

01:04:30 But I think on the longer timescale of, I think the five, 10 years where it happens are going to be very messy for sure. But I think the time after that is going to be a time of probably the best time ever. I mean it's always the case I think. So next year is always better than the previous one in human story more or less give or take a few accidents. But I think overall this trend is always positive because I think our poor guys gives us more freedom, enables us to do more, to give us more freedom, to have more independence. And so I think that's going to be very positive. But obviously this is not to say that there might not be some complexities along the way that are problems that we need to solve. Obviously there's a lot of stuff to figure out.

Jon Krohn: 01:05:11 Articulately said, I couldn't agree more with everything that you said. We're on exactly the same page. You're preaching to the choir as it were, at least with me and probably with a lot of our audience as well. Before I let you go, Julianne, this has been a fascinating conversation. I know that you read a lot of sci-fi books. I think your book recommendation might be in that vein, which, so it kind of gives us, we've just been talking kind of sci-fi a little bit in real life, like real life sci-fi

Julien Launay: 01:05:38 In a way. In a way it's sometime actually a reflection recently when reading sci-fi books, reading depiction of artificial intelligence that they're actually, they fall short of the reality of what's happening. I think there are very

few books that actually where you read them and you are like now faced with lms what we have. I'm like, oh, actually what we have is better. The life surpassed fiction

01:06:02 On the book recommendation saying it's a book. Have people that have heard me before will know, will say that I'm obsessed. It's a book I recommend a lot. I heard a lot of sci-fi and I personally have a fascination with alien contact. I think one of the most interesting subjects in sci-fi is the idea of alien contact of contact with an intelligence that is different than ours. And I think based on our previous conversation, you might understand why I think the idea of different form of intelligence and how we might interface with them is a very fascinating topic. There's a really good book called Blind Sight from Peter Watts, which is essentially an alien context story and I won't spoil it, but humans actually in the book are very different. It's in the future. So humans themselves or intelligence have, and I say intelligences plural because they have evolved in different ways, but also one of the alien is extraordinarily alien and kind raise the question of, okay, how do you interface with that? How do you interact with that? And I find this to be a very fascinating topic. So yeah, it's often a book that I recommend and we recommend it again. I think it's an amazing book.

Jon Krohn: 01:07:06 Yeah, actually aliens came up in our research, so as usual, our researcher s MACIs brought up way more topics than I could possibly cover, but it does help me interview you even the questions that we don't get to. But actually you've talked about aliens before in the context of pre-training, creating something like aliens of extraordinary intelligence, yet little understanding and then the reinforcement learning, the post-training that comes later allows you to transform those aliens into helpful, grounded assistance.

Julien Launay:	01:07:33	Yes, I think this is an idea that people used to frame under, it's a bit less popular this day, but the meme with the sugar, the sugars or the sugar, I dunno how you pronounce it. I
Jon Krohn:	01:07:43	Dunno what word you're saying.
Julien Launay:	01:07:44	It's like, I think it's from Lovecraft, so it's this weird creator. No, no, no, no. It's a specific creature from it's sugars or sugars. I don't remember
	01:07:57	Pons. But anyway, there's this idea I think people were comparing for a while, models large language model with it. And there is a few memes on this of alignment is just putting just a mask on the terrible creator, on the very frightening and terrible creator. And this kind of comes from that immediately after pre-training. The models are very strange. They know a lot about us obviously because we train them on everything we have ever done. So how could they be so different from us? But the interface with us in a very weird way and some of post training is about aligning this and yeah, that's an idea that comes from here.
Jon Krohn:	01:08:36	Yeah, and you had the word exactly right there. It was one I wasn't familiar with. It seems like it's related in the kind of lovecraft universe, HP Lovecraft universe to kullu in some way. But Shog goth, S-H-O-G-G-O-T-H, I'll have links to images of them in the show notes,
Julien Launay:	01:08:50	Plenty of memes in machine learning about them.
Jon Krohn:	01:08:55	Nice. Fantastic. Julian, this has been amazing for people who want to hear more of your brilliant thoughts after this episode, how can they follow you?

Julien Launay:	01:09:01	Yeah, I mean it's a very boring corporate way on LinkedIn, but otherwise on Twitter I'm at sleepy Lolo, which I think we can put a link instead of
Jon Krohn:	01:09:12	Sleepy Hollow.
Julien Launay:	01:09:13	Sleepy Lolo. Sleepy Lolo, L-I-P-P-Y-L-O-L-O, which is a very long backstory, but yeah.
Jon Krohn:	01:09:21	Okay. Yeah, we'll have that in the show notes. Exactly. I'm sure we'll arrange that. Thank you so much, Julian. It has been a treat to have you here. I learned so much. Thank you.
Julien Launay:	01:09:29	Thank you very much. Good.
Jon Krohn:	01:09:34	What an exceptional conversation with the brilliant Julien Launay. In today's episode, Julian covered the evolution from pre-training that's predicting next tokens on webscale data to post-training. That's reinforcement learning as the dominant phase of LLM development. He talked about how Adaptive ML'S platform makes reinforcement learning accessible to data scientists enabling companies like at t to autonomously tune smaller models to frontier performance. He went in detail on the three types of reinforcement learning feedback. That's RLHF from human feedback, like human thumbs out, thumbs down, R-L-A-I-F, where AI models evaluate performance and RL eef where we have verifiable rewards from code execution or game scores. Julian gave his prediction that we'll achieve super intelligence within five to 10 years, but that it will be messy and spiky revolutionary in domains like mathematics while still requiring real world experiments for things like biology and medicine. And he talked about why current silicon-based computing is likely sufficient to bootstrap a GI, which will then help us to scale new computing

paradigms like photonic and quantum computing technologies.

01:10:44 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 Julian's social media profiles, as well as my at superdatascience.com slash 9 1 3. Alright, thanks to everyone on the SuperDataScience podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, our partnerships team, which is Nathan Daly and Natalie Ziajski, our researcher, Serg Masís writer, Dr. Zara Karschay, and our founder Kirill Eremenko. Thanks to all of them for producing another excellent episode for us today for enabling that super team to create this free podcast for you. We are so grateful to our sponsors. You listener can support this show by checking out our sponsors links, which are in the show notes. And if you're ever interested in sponsoring an episode yourself, you can find out how to do that at john crone.com/podcast. Otherwise, you can support us by sharing the show with people who would enjoy the episode, reviewing the episode on your favorite podcasting platform. Subscribing obviously if you're not already a subscriber, but most importantly, I just hope you'll listen to us, you'll keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the Super Data Science Podcast with you very soon.