SDS PODCAST EPISODE 713: LLAMA 2, TOOLFORMER AND BLOOM: OPEN-SOURCE LLMS WITH META'S DR. THOMAS SCIALOM

Jon Krohn: 00:00:00 This is episode number 713 with Dr. Thomas Scialom, A.I. Research Scientist at Meta. Today's episode is brought to you by AWS Cloud Computing Services, by Grafbase, the unified data layer, and by Modelbit for deploying models in seconds.

00:00:21 Welcome to the Super Data Science 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.

00:00:52 Welcome back to the Super Data Science podcast. Today we've got the trailblazing AI researcher Dr. Thomas Scialom on the show. Thomas is an AI research scientist at Meta. He's behind some of the world's best-known generative AI projects, including Llama 2, BLOOM, Toolformer, and Galactica. He's contributing to the development of Artificial General Intelligence, AGI. He's lectured at many of the top AI labs such as Google, Stanford, and MILA in Montreal. He holds a PhD from Sorbonne University in France where he specialized in natural language generation with reinforcement learning.

00:01:25 Today's episode should be equally appealing to hands-on machine learning practitioner as well as folks who may not be hands-on, but are nevertheless keen to understand the state-of-the-art in AI from someone who is right on the cutting edge of it all. In this episode, Thomas details Llama 2, today's top open-source LLM, including what it was like behind the scenes developing it and what we can expect from the eventual Llama 3 and related open-source projects. He talks about the Toolformer LLM that learns how to use external tools, the Galactica science specific LLM, why it was brought down after just a few days, and how it might eventually reemerge in a new form. He talks about RLHF reinforcement learning from human feedback, which shifts the distribution of generative AI outputs from approximating the average of human responses to approximating excellent, often superhuman quality. He talks about how SUNY thinks AGI, artificial general intelligence, will be realized and how, and how to make the most of the generative AI boom as an entrepreneur. All right, you ready for this tremendous episode? Let's go.

00:02:35 Thomas, welcome to the Super Data Science Podcast. It blows my mind that you're here on the show, that we get to have you here. I'm so excited for this interview. Where in the world are you calling in from today?

Thomas Scialom: 00:02:47 From Paris.

Jon Krohn: 00:02:49 Nice. It's been a while since I've been to Paris, but I've never had a bad time there.

Thomas Scialom: 00:02:55 Yeah, neither.

Jon Krohn: 00:02:59 Nice. So, we know each other I'd say almost serendipitously. I did an episode a couple of weeks ago on Llama 2, so episode 702 is this, I don't know, it's like a 15-minute, maybe 20-minute episode with just me describing from my understanding all the new capabilities with Llama 2, how the model came about a little bit. And as I was opening up the technical paper, there's like I don't know how many, there's probably like 50 authors and they're in this big long list, listed vertically on the side of the technical paper page. But somehow my brain noticed that I recognized one of them. I was like, "Anthony Hartshorn. I know Anthony Hartshorn. There can't be two people named Anthony Hartshorn." And so I sent him a message and I said, "Do you want to be on my podcast? We're the most listened to podcast in the data science industry." And he suggested you as the guest instead, Thomas, which is amazing because you're the final author on the paper, which in the academic world it might sound to a normal listener like being the final author should mean that of the 50 people we have the person that made the smallest possible contribution, but in fact on academic papers, that isn't how it works.

00:04:26 So, you have very often the first author is maybe the person who actually wrote, put everything together, but then traditionally in academic work, the last author will be the head of the lab that brought in the funding and that was kind of overseeing the project. So, truly it's an honor to have you here, Thomas.

Thomas Scialom: 00:04:48 Thanks for having me.

Jon Krohn: 00:04:49 So, at the time of recording this episode, it's only been a few weeks since Meta released the open-source large language model, Llama 2. You were a science and engineering leader for this groundbreaking development. Can you explain the significance of Llama 2 in the context of other recent advancements in AI and generative models? Maybe kind of fill us in on how the Llama projects in general, that Meta was like, you know what, we're going to invest and obviously you're not going to divulge on air, but there's rumors that eight figure sums have been invested in creating Llama 2. And so it's interesting, even from the very beginning, what was it like maybe to get this kind of buy-in from the organization to be doing this open sourcing?

Thomas Scialom: 00:05:39 Yeah, I think, so no doubt large language models are a big deal. They have made some breakthrough in the research. I think also we had a ChatGPT moment at the end of last year and most of the people realize the potential of this technology. And so I think we did mainly two things with Llama 2. One, we, what we call align some model with techniques called RLHF, for instance. I can dig more in depth later if you want this, but basically the idea is you have what we call a pre-train model, which has kind of reads internet on the next token prediction. So, it tries to predict the next token, and this is what we call self supervision. It's supervision because we have a target, but it's self because text on the web are vastly accessible like that. And so just with that you have a pre-train language model, which we had with LLaMA 1 and we did again with Llama 2 and extended it a bit incrementally.

00:06:46 And that's where all the new edges learn, all the capabilities kind of emerge, but then it's hard to access. And the magic behind ChatGPT is its kind of interface as a chat, which is very natural. And to follow your instructions to say, "Oh, but talk like this person, or do these kind of things. Or make it more like a markdown or bullet point or change that or make it shorter." And it understands your instructions and does it precisely. And this happens at fine-tuning, it's kind of refining educating a pre-trained large language model, which we did also with Llama 2. And that was one of the main innovation, because I mean no one had done that at this scale and open-source a model and explaining all the research behind in a research paper as we did. So, before Llama 2 basically the only large language model online that were available, like OpenAI, Anthropic, Google with Bard, they were closed behind an API. So, I would say that's the main innovation in term of science and in terms of impact for the communities, the research communities, the business. I think you mentioned, and you're not the only one, your company now use Llama 2. This is also possible because we also change the license to something commercial, user-friendly for commercial applications.

00:08:14 This is also possible because we also change the license to something commercial, user-friendly for commercial applications.

Jon Krohn: 00:08:23 Yeah, exactly. I don't have 700 million users at my machine learning company yet. So, this commercial license that allows, as long as you don't have more than 700 million active users, it's okay to use Llama 2. So, for us, it's brilliant. Previously we had been using as our base model, so we have a number of different kinds of generative AI capabilities in our platform for our users. And so something like LLaMA 1, which was pre-trained but not fine-tuned, that would've been actually fine for us as a starting point except for the commercial use limitations. So, we never could use the original LLaMA in production because obviously there was this commercial use restriction. It was for academic purposes only. And that also meant that some of the initial fine-tuned architectures that came off the back of LLaMA, like Alpaca out of Stanford and like Vicuña that Joey Gonzalez, who was in episode number 707 of this show, developed at Berkeley.

00:09:41 And so all of those, that whole family of models, we were like, "Oh man, we're going to be left out." But then luckily some groups did come along. So, Databricks released Dolly 2.0 for example, and there was some other, and I've done episodes on these open-source alternatives that are commercially licensable. So, episode 672, I talk about different open-source options that are available where you not only have that pre-training with the self supervision that you were describing, but also the fine-tuning based on human feedback. That means that the responses are going to be deliberately helpful and more like a conversational, like a chat.

00:10:26 So, we had been using Dolly 2.0 from Databricks as our starting point for the last couple of months. When Llama 2 came out, there was something... The scale, you described this already, the unprecedented scale in terms of the number of tokens, two trillion tokens for pre-training and over a million data points for the fine-tuning, this kind of scale, its orders of magnitude more. The Dolly 2.0 for comparison had 10,000 instructions that were fine-tuned on. You're talking a hundred times more. And with these large language models, the scaling laws that we've seen come out, like the Chinchilla scaling laws, have showed that you kind of have three levers to getting a great model. So, the number of parameters, the training dataset size and training time. And it seems like with Llama 2, you and your team have tried to max out all of those things, especially with the 70 billion parameter Llama 2 model.

00:11:33 So, that's I guess something that's also worth, if people haven't listened to my Llama 2 episode already, then you may not be aware that it isn't just one model that was released here. We're talking about a model family. So, there's a 7 billion, 13 billion and a 70 billion parameter model. And those two smaller ones, they'll be able to fit on a single GPU. And so this means that you can run them relatively inexpensively. And so with applications like with my company where we have a relatively discreet number of generative tasks that we need the model to perform, we can take that 7 billion or that 13 billion and we can fine-tune it to our tasks. And so for listeners who aren't aware, you can do this yourself at home using a parameter efficient fine-tuning technique like LoRA, low-rank adaptation, which I talk about in episode number 674.

00:12:25 So, you can take the model like Llama 2, and so the 7 billion, 13 billion, you can typically vary inexpensively for tens of dollars or hundreds of dollars, you can fine-tune that to your own specific tasks. And for us, that's perfect. It means we now have this amazing large language model that it's as good as GPT-4 or better in our own tests when we start with Llama 2 and we fine-tune with our own data at this narrow range of tasks that we have. And then if you're a listener out there and you're like, "Well, I want the absolute state-of-the-art," then you can use Llama 2. And at least in terms of open-source, this is going to be the state-of-the-art. So, I've just talked a lot. But the point is that, Thomas, what you've done and what this means for us as a community to have access to something like Llama 2, it's a game changer. It was obvious that it was a game changer within minutes of starting to read the Llama 2 materials online and my data science team at my company immediately started retraining our models with Llama 2.

Thomas Scialom: 00:13:34 It's always good to hear. Thanks. Maybe worth mentioning what we realize also is, so it was extended in context length from two to 4,000, et cetera. It's on text only for now. But I think that's also the magic of open sourcing. We don't want to push for access, for as a community we will deal with that easily. And we know that extending the context length, that fine-tuning is possible. We know that connecting multi-modal inputs is straightforward. And what was magic is after the release, within a week people have done that efficiently. And so that's also one of the strengths, in my opinion, of open-sourcing this kind of models. And we see much more innovation with shorter cycles of innovation thanks to that. So, that was one of the philosophies. So, we went, as you said, all in on the scale of the things that we can do at Meta to make it as good as we can so that everyone could use it in the end to adapt it for the use cases.

Jon Krohn: 00:14:45 Amazing.

00:15:21 Are you stuck between optimizing latency and lowering your inference costs as you build your generative AI applications? Find out why more ML developers are moving toward AWS Trainium and Inferentia to build and serve their Large Language Models. You can save up to 50% on training costs with AWS Trainium chips and up to 40% on inference costs with AWS Inferentia chips.

Trainium and Inferentia will help you achieve higher performance, lower costs, and be more sustainable. Check out the links in the show notes to learn more. All right, now back to our show.

00:15:25 And another thing that you did with Llama 2 is there's extensive thought around ethics, responsible use, acceptable use. So, for example, there were red teaming exercises where you simulate internally that you have these malicious actors. And so can you dive into why this was so important? I think this was unprecedented also. So, not only was the amount of data for both the pre-training and the fine-tuning steps unprecedented, but for an open-source model, I think that the level of concern that went into the ethics and responsible use is also unprecedented.

Thomas Scialom: 00:16:14 So, yes, maybe let's give a bit of context. The strongest LLMs so far were, as we said, accessible only on an API. I think that was problematic in several aspects. It's led on research, it prevent academia to explore, industrial, to have commercial use cases. And to be honest, we will be nowhere without open sourcing. Think about BERT, Transformers and even GPT-1. That being said, the risks at present and future with respect to learn have been arguably discussed by some of the researchers. I think OpenAI and Anthropic did an extremely great and important invaluable job at tracing the bar for safety. And I'm glad they did. So, the thing is when you have an API like them, it's easy to control, you can put classifiers on top of that, you restrict the access somehow.

00:17:15 There's clearly a very hard challenge when it comes to open-source, because you release the weights and you enable everyone to fine-tune, to do whatever, to control the models. So, while I feel it is very important to do it and I think we're not yet at a stage where LLM's are so dangerous that we should not do it. It was important to do it in a responsible way to raise the bar even higher than what has been done for competitor models driven API, because the risks are bigger when you open-source it. And so we had a lot of inspiration for the works that were done at those companies at OpenAI, Anthropic, and we apply all the method we could and some new methods we discussed in the paper to make the model as safe as we could. It's not perfect. There's still some jailbreak, but maybe we can discuss that later, but I feel we had two main complaints that followed the release. And one of them was, it's too safe. And there's an example where for instance, I don't remember, can you kill the script or something like that and the model say, "No, it's not good to kill."

Jon Krohn: 00:18:29 Right, right, right.

Thomas Scialom: 00:18:30 So, I mean, well, there was a system from top of it. If you remove it, the model is actually better. But to me this was a success in that this was the first time we release an open-source, a model of scale, and so we had the responsibility to raise a bar for safetyness and responsibility. So, because it was unprecedented, I prefer to be on the side that it's too safe and progressively decrease the level of safety if needed for future release than the opposite.

Jon Krohn: 00:19:08 And so actually your discussion of that reminds me that when I was doing my research for my solo episode about Llama 2, episode number 702, with that episode, when I was digging into your technical paper, it actually talks about four models. So, three models that were released were the 7 billion, 13 billion and 70 billion parameter models. And then off the top of my head, I think that it was 34 billion was another model that you trained. But I noticed that, for whatever reason, there was a chart with some metric of safety.

Thomas Scialom: 00:19:47 Absolutely.

Jon Krohn: 00:19:48 And that model, for some reason, the 34 billion one, seemed it was more like the existing open-source LLM's in terms of safety. So, it was kind of more like Falcon or more like Dolly 2.0. And so it seems like you've held back a model, I'm guessing, and you don't need to confirm on air, but that it seems like because it didn't meet the security standards of the other three, which is an interesting thing to have happen because presumably the same process was followed for all of them.

Thomas Scialom: 00:20:23 Yeah, that's absolutely correct what you said. And that's one of the main reasons we didn't release it. One thing also, it's probably that we don't know, we didn't have the time to investigate. What people have to understand is that just the process together, starting from the pre-train model to fine-tune it to apply RLHF, reinforcement learning, to then evaluate it automatically, then evaluate it with human at details and with red teamers, which are expert at finding the failure, trying to make the model say something bad and they put the model in the hardest possible ways to make it say some stuff. All this process takes a lot of time. And so we just decided based on this bad point, which we don't know yet why we didn't have the time to investigate. Maybe it's an error in the evaluation, maybe it's a model that was not well fine-tuned. I don't know exactly yet. But we just said, "Okay, why wasting one, two, three more weeks just for that? We can already raise a smaller model. The biggest model, the more capable, let's not wait to let everyone use it."

Jon Krohn: 00:21:34 That makes perfect sense. And so it's kind of nice to have that confirmed, because that's actually what I speculated on here earlier. So, great. So, you mentioned that there were two main complaints. One of them was that it was too safe, so people were complaining that Llama 2 is too safe. So, things like somebody saying I want to kill this process leads to it saying "I can't kill, killing is bad." What was the other big complaint that people have had since the release?

Thomas Scialom: 00:22:00 Tell me if you heard the same, but from my perspective it was safety, too safe, and code. Bad code abilities.

Jon Krohn: 00:22:07 Oh yeah. So, I do say that in my episode 702 as well is that it seems like... So, when I say that Llama 2 performs at the state-of-the-art relative to any other open-source model, that's on natural language tasks where it's like natural language in and out. So, my understanding, and I haven't tested this extensively myself, but where there's code being generated or where you're asking it to do kind of mathematical problems, my understanding is that it doesn't perform as well as some other options out there.

Thomas Scialom: 00:22:44 Yeah. So, to that, we actually rushed so fast from LLaMA 1 to Llama 2 to get visibilities. We focused mainly on natural language and not code. I agree the model is not that good at code or mathematics for now, but we are working on that and, well, at the time of the podcast will be a released, I hope that some Code Llama will be also released.

Jon Krohn: 00:23:13 Oh, very cool. All right, that's awesome. That's exciting to hear. So, I mean that kind gives us a sense, it's a really tantalizing glimpse. It's possible that by the time this episode is out that will be old news. But yeah, a Code Llama, that sounds very cool. Is there anything else that you can tell us about where this research might be going? I understand, I don't want to be extracting information under duress.

Thomas Scialom: 00:23:47 I mean we are the open guys. I mean in general there's no clear secret. We'll try to improve the models in general, which means scaling them, keep training them on motor currents, increasing the abilities, maybe tackle more multimodality codes. We just discussed that reasoning. We will try also to improve the RLHF stage to capabilities. We'll go also on, one of the direction is obviously tools, teaching this model to use in zero-shot fashion, some tools maybe to access the web more easily. All those directions seems quite reasonable and expected, so there's no big secret. Now the question is more like how we will do that. Will we make it some breakthrough discovery in the way that will enable us to largely improve? Hopefully yes.

Jon Krohn: 00:24:50 Nice. Yeah. And you mentioned there being able to handle tools, which is something that you have a lot of experience with because you've also been involved with the Toolformer LLM. So, this is an LLM that came out earlier and the Toolformer is specialized to decide which API to call in a circumstance, when to call the API and what arguments to pass, and how best to incorporate the results into the next token prediction of the generative model. So, maybe this is a good time to kind of switch over and talk about this Toolformer project since it sounds like future Llama iterations might incorporate some of that kind of capability.

Thomas Scialom: 00:25:38 Yeah. The Toolformer was connecting large language models with tools was an idea I had last summer a year ago. It fell to a natural extension of all these models, Retro, Atlas, RAG, where you augment with a retriever, a language model and the intuition is very easy. So, the idea was to train together a dense retriever and a language model so that you'll augment the context. And so when you ask a question, you will search on all the training data some relevant passages. And so if the model didn't remember, memorize well, it will boost the capabilities which was very efficient as shown in all those papers. But so this is what we call a non-parametric framework, because you rely not only on the parameters, the weights of the model, but also on external source of knowledge that could possibly grow to time to, for instance, incorporate new fresh information without necessarily retraining the model.

00:26:43 But that being said, my idea was to extend this to a non-parametric general framework where you could see, and there was some work at the time that was doing that, you could see how using a calculator or Python executor or different search engine, maybe I'm using Google for some search and Google Scholar for the specific search on papers. And so the idea was to just give a set of tools to the model and much more like a human-like way teach it to use them given the context, not at each [inaudible]. But so the model now has to know when to use the tool, how to use it to benefit from this performance. And so Toolformer, Timo Schick led this work and we published it in February and I think it was also very pleasant timing. It was two months after ChatGPT and everyone was kind of, "Well, the game is over, ChatGPT, Agile is there, what's next?"

00:27:48 But ChatGPT at the time was just limited to a window like you're chatting with an agent that has no access to the world and that changed a lot the perception that you can have once you can give the LM the access to the world to some knowledge, it makes the experience for the user completely different. It extends the capabilities dramatically. And so that's what we have done with Toolformer with some self-supervised techniques. So, the model learned that basically itself when it increases the perplexity using it all. So, that was the main idea.

Jon Krohn: 00:28:26 And so this may be familiar in an analogous way and you can tell me where maybe the analogy breaks down, but having not used Toolformer myself yet, it seems to me to be similar to what later happened with ChatGPT with the plugins so that now with ChatGPT, you can turn on third party plugins. So, if you turn on the WolframAlpha plugin, then when you ask ChatGPT to do a calculus problem, it's going to bring in WolframAlpha to use that API as opposed to trying to use next token prediction to do math, which works surprisingly well in a lot of circumstances given that it's mind-boggling that this next token prediction can often do math correctly.

00:29:21 But you're basically guaranteed a correct answer, a correct differentiation, for example, if you use WolframAlpha to do it. So, ChatGPT will automatically detect, okay, this is a circumstance where I should be using WolframAlpha, let's do some math with that or it can access the web, like you said, it can do a web search or it can plug into websites like Kayak to book you a trip and to find you the car rental and book the hotel. So, is that the analogous use case Toolformer? Toolformer is obviously open-source.

Thomas Scialom: 00:30:01 Yeah, I mean I think the idea was there. I saw a lot on Twitter when one month late after Toolformer, OpenAI user plugins. So, they actually site in the plugin page Toolformer and some people say, "OpenAI reimplemented Toolformer in one month." Honestly and humbly I think the idea was in the air and we had a good timing [inaudible]. I think also the method used by OpenAI was quite different from Toolformer. So, that's quite interesting. In Toolformer the idea was, so we had access to bad... I mean at the time language model compared to GPT-3, at least. It was before Llama. And so what we did is with the self-supervised method, which works kind of well, but my conclusion also at the end of the work was we need more capable base model and fine-tune align model such that we learn to use tool with some instruction following scheme, which is also why I stepped back from Toolformer at the time and not extended the project to work on Llama 2 and making it working with instruction tuning to follow the instruction of the users.

00:31:16 And actually you have one paragraph at end in the discussion analysis, the paper, showing kind of emergence of tool use where you just with a prompt describe, you tell to the model basically natural language, you can use a calculator, use this format. For the API, use a search engine, use this format. Now I don't remember which one it was in the paper, but what's the difference in height between the Eiffel Tower and Empire State Building, and then naturally say step one, search height of the Empire State Building, search height of the Eiffel Tower and then calculate the difference between the two. So, you can see how from Toolformer where there's the idea of using the tools, but the method is pretty efficient but yet I would say is obsolete with a better line model we move to Llama 2 to now maybe come back to Toolformer.

Jon Krohn: 00:32:08 Right, right, right, right. Makes perfect sense.

00:32:52 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.

00:32:56 It's exciting how these different research threads diverge together and it kind of sounds like you had that vision all along that you're like, "Okay, cool. Toolformer works really well, but it could be better if the base model that was calling it was better. And so let's focus on this Llama 2 project for a while and then come back and worry about this API calling from Llama 2 later on." Very cool. Looking forward to that. And that's similarly for the kinds of things, again with my own machine learning company, that kind of ability having these really powerful models like Llama 2 with open-source API calling abilities built in, this is huge for us as well because it means that there's all kinds of cool things that we can do internally.

00:33:49 Like a lot of companies, we use APIs, these kinds of microservices to make it easy to have these different compartmentalized services within the platform. And so with something like Toolformer, we can then be able to say our users could provide natural language instructions, just have a natural language chat with our platform and all of the capabilities of the platform, the large language model behind the scenes can say, "Okay, I think that they're asking for this particular kind of data or this particular kind of task to be done and we have an API for that, so let's go use it." And then the results are returned back in exactly the kind of format like a JSON format that our platform was expecting. It can make the API call successfully, it can return the information from that call and present it to the user. It's a very cool thing to be able to do.

00:34:54 I mean it sounds like with the level of worry, the level of concern that went into making sure that Llama 2 is used ethically, something like Toolformer, maybe this kind of ties into even AGI concerns because people say, "Oh, AGI won't be that dangerous because it's not going to be connected to the world." But that's obviously not true, because with projects like Toolformer, we see that no, they will be connected to the world. In my company we're using something like Toolformer to be able to query software APIs and get information back, but there's no reason why those couldn't be connected to hardware, why these couldn't impact the real world. So, I just wonder if you have any thoughts on that and maybe we can have a bigger AGI discussion later in the episode.

Thomas Scialom: 00:35:51 Sure. But no, maybe [inaudible]. I think those are very good points and actually we take safety for the tool direction very seriously. That makes the thing quite different from a kind of closed LLM in a window with just chat in demo. There's real risks at another order of monitor measured. So, for sure there's new concerns, new research questions and problems on the way that makes it very serious.

Jon Krohn: 00:36:25 Nice. Okay, well, yeah, that's a clear answer.

Thomas Scialom: 00:36:29 There's a survey on augmented large language model we published also in February just after Toolformer. We have a section at the end of that saying like augmented language models, augmentation of notary tools where a model can now take an action in the world. This is a different story than before.

Jon Krohn: 00:36:49 Nice. Yeah, yeah, yeah, no doubt. So, in addition to Toolformer, another LLM project that you were working on before Llama 2 was Galactica. And so Galactica was a large language model that is I suppose specifically designed for handling academic research, scientific papers and these kinds of scientific questions. The Galactica model was only live for a few days I guess. So, I don't know, it seemed like a really big deal and then it was taken offline. So, maybe tell us a bit about the project and maybe the thinking behind bringing it down and maybe whether it will be back in the future.

Thomas Scialom: 00:37:34 Yeah. So, there's this website which is one of the most well known for researchers called Papers With Code, a company that was acquired by Meta. And so the project of the team was, that was kind of visionary but large language model. They wanted a large language model for science that will help us to access information for science, to help us develop creative writing for science, maybe connect different ideas for science stuff, find some papers that you will never find on Google Scholar just based on the ID. And that's what large language model are capable of and that's what Galactica was about. And actually that was one of the first open large language model that works pretty well. And it was in some aspects far ahead of its time and some aspects we made probably some mistakes on the way. It was only a pre-train model, not an instructed model. And so maybe we presented it way too much as something that can answer questions, do things, and it'll have worked so well after an instruction tuning phase.

00:38:55 The second thing also probably we did not well was to overclaim a bit on the webpage saying it can write a paper, and I can understand how for scientific person working in the science this will feel like overclaiming. That was not our purpose, but anyway, because of all the noise, and that was quite some noise at the time on Twitter and stuff, we decided to remove it. It was also a weird time because at that time there was a lot of people still criticizing large language models that were quite noisy on Twitter. And on top of that, some people from the scientific community that say large language are dangerous for science, et cetera. And it was just two weeks before ChatGPT release, so that was an interesting timing. I think for instance people don't realize how good it was at citations.

00:40:03 I fine-tuned it myself to give you an examples of following instructions and when you say, "Cite a paper about bias," it will find the papers. For instance, to give you an example of maybe more that will speak more, Chinchilla, the scaling laws, I think Chinchilla doesn't appear in the title of the paper, or scaling law doesn't appear, one or the other, I don't remember. And so just saying, "What's a citation for Chinchilla?" Which is not in the name of the title, it will find you, write it and you could just click and add it to a [inaudible] when you are writing scientific paper. So, it was kind of connecting the things like that. And from the test we did, it was outperforming some of the Scholar or Elastic search engines. And I think as search engine LLMs have not been yet well explored, but that's something big.

Jon Krohn: 00:41:02 For sure.

00:41:05 Deploying machine learning models into production doesn't need to require hours of engineering effort or complex homegrown 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.

00:41:43 And it's interesting how well Galactica was doing at being able to do citations and accurately do citations when something like ChatGPT, especially with the GPT-3.5 API running in the backend, it was famously creating citations that sound plausible but aren't real. Even creating URLs that are made up, which is what you'd kind of expect, probably what you and I would expect when the models are trained the way that they are. But for ordinary users, for laypeople, they think, "What is this? Why would this happen?" And then you even end up with lawyers presenting cases that never existed to a judge as a result of this kind of thing.

00:42:30 So, it's cool that Galactica was able to do those citations even before the ChatGPT release last year. Nice. And so speaking of the kinds of the issues with large language models, another big issue with LLMs has historically been the expense associated with all the human labor to create a curated dataset. So, you mentioned right at the beginning of the episode how there's this pre-training step where it's self-supervised, where you can just use natural language, it doesn't require any labeling and that gets us to model weights that have this rich understanding of the world, but the model isn't calibrated to be optimally answering questions from people and performing tasks based on instructions.

00:43:27 And so it's this second step after the pre-training we do this fine-tuning. For that fine-tuning step, historically we've wanted high quality data sets. So, the Vicuña people for example, so Joey Gonzalez's team at Berkeley, they took the original LLaMA, which was just pre-trained and then they used hundreds of thousands of conversations that people had shared. I'm forgetting the name of it off the top of my head, but there was a browser plugin that lots of people were using to save and to share interesting conversations that they'd had with ChatGPT.

00:44:05 And so this was in the public domain and so the Vicuña people at Berkeley took that dataset and used it to fine-tune Llama and create this Vicuña LLM, which still today actually has a remarkably good performance for relatively small open-source LLM compared to other kinds of open-source LLMs, even compared to many of the proprietary options out there. So, this kind of trick can get you so far, but ultimately you might want lots more of these instruction pairs, for example, of these labeled data to be able to create a powerful fine-tuned LLM. And so my understanding is that the Unnatural Instructions project that you are a part of at Meta was designed to help alleviate this issue.

Thomas Scialom: 00:44:54 Yeah, that's interesting because at the time of a natural, actually there was not even a ChatGPT or whatever. And so at this time you had on one way OpenAI with GPT Davinci free, Davinci 1. It was a good instructed model, very capable. And on the other hand you had just kind of remarkably not that good, not that bad pre-train model and instructed the data sets very academic oriented I would say, from standard tasks like summarization, question answering. You clearly don't have the diversity of instructions that people will have asked and that Davinci instruct was good at answering. How to collect this diversity of instruction is actually extremely challenging, even for humans. Think of 10 different tasks and instruction right now, it will be hard for you to come with this level of diversity. And so at a scale of 1,000, one million diversity, that's pretty hard.

00:45:56 Somehow OpenAI managed to do that with Davinci, maybe they collected some data from their API, they had some annotators, which is well known from years ago. They had some experience. Now the question is what we found out, the question was how to get some diversity. With ChatGPT people type some instructions and you have the output of the model. Now the question is when you don't have even that, how can you generate not only the answer from the model but the instruction? And what we found out is that somehow you can ask the Davinci 3, GPT 3.5, I think or the version before, to generate those instructions. So, you can say, "Generate me instruction and output for card, for this topic, for simulation," or just without specifying any topic.

00:46:59 It will come and generate a lot of samples and examples, not only the answer, but also the instruction so that you can create an unnatural dataset that actually we found out to be more natural than some of natural data sets at the time. The reason was that natural data sets published by research talent AI using actual humans to create the data was kind of lacking of diversity and was academically oriented, while somehow the model from OpenAI managed to generate a large diversity, much more close to actual use cases. I think we could see it as kind of a distillation process of a more capable model that was fine-tuned on this data. And that was kind of a temporary solution for people that had not access to instructed models, which is also one of the reason we moved to Llama 2 and did the process from scratch to create our own data. Indeed, we paid quite a lot for that. We noted more than millions of annotation. We did the whole all RLHF stage and so now we have such capable model. At the time, no one had the models at this scale.

Jon Krohn: 00:48:15 Nice. Yeah, that's a great overview of the project. Let's dive into that. You mentioned near the beginning of the episode this RLHF, reinforcement learning from human feedback. This is a key part of the fine-tuning process and with Llama 2 you introduced a unique two-stage RLHF process, which evidently has led to even better results. So, not only having this large annotated dataset of more than a million training data points, but you use this new methodology, this two-stage RLHF. So, do you want to explain RLHF and particularly this two-stage process to us?

Thomas Scialom: 00:48:59 Yeah, so RLHF stands for reinforcement planning with human preference. And the idea is to fine-tune the model, you type a prompt, a question to the model, you sample different outputs and you ask a human to instead of writing the perfect solution and fine-tune the model on what the human will have write, you try to train the model to go towards the direction of what human prefers among its samples. And at the beginning of the project, so we knew that was kind of the backbone of some of the instructed models from Anthropic, OpenAI.

00:49:39 But if you have asked me at the beginning of the project and most of the researchers around me, the question is supervised data. When I ask annotators to write answer it's kind of gold letter. That is what is considered by the community in general. You need to take good annotators, high quality annotators, sure. But this is very expensive and comparing a model to generate, write itself the answer. But two answers, and ask a human to prefer, this takes way less time. And so you can scale it way more. And so if you ask me, I would say, okay, if I have an infinite budget, maybe I prefer supervised learning and ask the humans to do that, but it's not scalable. So, sure we will do RLHF.

00:50:27 The thing is that I realized that for some time is that there's some magic which is not well understood, I feel, by the community yet, where we already have some superhuman performance on some creative writing tasks. An example I always give is like write a haiku poem about large language models and the [inaudible]. And so then we come with something, I mean ask me, I don't know about you, but if you ask me, I will take an hour and then we'll come with nothing. And models are good at that. And the reason is the model are super capable and have seen all the distribution on internet of the humans, can think about an examples with coding. So, it knows the middle distribution of average coders, it knows the distribution of good coders, excellent coders and bad coders. And so if you ask annotators to write code, you would probably imitate this distribution.

00:51:30 And by imitation you will have the distribution of 5% of the time it's great, 50% of the time it's in the middle and sometimes there's some mistakes. And every human makes some mistakes. Now if you apply an RLHF, this is kind of different and there's where the magic is, you will shift the distribution toward excellence, toward even better than the best annotator you have. Because the thing is, even if you are the best annotator, you will write at your best capabilities and you will do some mistakes and the model will imitate you. But now if the model imitates you and you sample 10 times. On the 10 times it will sample some examples that are really good, your best examples, and sometimes the worst examples. And so you can tell him, "No, this is the best example I wanted." And sometimes it will also explore a bit beyond and do something that even you won't have done. And so because it's easier for humans to compare, I can tell you which poem I prefer, I can't write them. And so because of that, you can have some emergence of superhuman capabilities on some tasks thanks to RLHF.

Jon Krohn: 00:52:39 Yeah, yeah. And you're touching on something that blows my mind all the time about what we already have today. And this is why the release of GPT-4 in March was such a big deal for me and shifted my own perspective on the realization of artificial general intelligence, AGI, an algorithm that has all the learning capabilities of a human in our lifetime. Because already with GPT-4 because of this magic of RLHF that you're describing, the shifting of the distribution from intuitively imagining a normal distribution in my head where the outputs are going to be exactly as you described, they're going to be middling most of the time, sometimes excellent, sometimes poor. But with RLHF, we shift everything so that it's excellent all the time. And the haiku example that you gave is great, because a lot of people have the experience of using GPT-4, though probably the experience of using Llama 2 is similar.

00:53:37 And by the way, any of you listening right now, you can go to, at least at the time of recording, it's probably still the same at the time of this episode's release, you can go to Hugging Face chat and the default model now for Hugging Face chat is the 70 billion parameter Llama 2 chat fine-tuned model. So, you can experience yourself.

And the queries that I've done in that Hugging Face chat have been comparable to what I'd expect with GPT-4. But either way, with one of these state-of-the-art open-source LLMs, it's capable of doing so many more things than I could as an individual. Obviously you expect to be able to come to this interface and ask a question about anything in the world, and it knows the answer and it can articulate it well, and it can dive deeper and it can explain why it did it a certain way.

00:54:32 And when you argue with it, when you disagree and you say, "No, no, I thought it was this other way," it often knows, "Oh yes, that's a common misconception." And so it's interesting that we're like, "Oh, how far away is artificial general intelligence in this thing that's capable of learning everything that we can learn?" And already today what we have, while maybe it isn't as good as humans on some tasks, it is so much better than an individual human at so many things, that in some ways we've already attained this really crazy superpower here on this planet.

00:55:17 So, I don't know. It kind of just gone off on a tangent. There wasn't really a question, but yeah, our researcher, Serg Masís, he often digs up the most incredible things on people, on our guests. And one of the things that he dug up on you was that five years ago in 2018, and I don't even know he might've translated this because you were saying this to French children, you said that there's evidence that we are not at all close to achieving general intelligence and that it's a fantasy. But I mean my perception has shift. An example that I've given I think on air before is that a year ago I was giving a TEDx talk in Philadelphia, and my whole point of the TEDx talk was that because of AI, technological progress is moving so rapidly that we can't predict accurately even a few years from now what kinds of capabilities we'll have.

00:56:20 And if somebody had asked me at the time of the talk a year ago whether we would have an algorithm that could do the things that GPT-4 can do or that Llama 2 can do, I would've said, "I don't know if we'll have that in our lifetime." And now a year later we have it. And people like you are making it so that anybody can access it open-source. It's wild. That shift is unreal. And it has me now, I went from being a skeptic about what can happen with AI in our lifetimes, to believing that, yeah, some really crazy things are probably going to happen in our lifetime. So, I don't know if you have any more thoughts on that. I know that you've been interested in AGI for a long time and what are your thoughts on when we might realize AGI or artificial super intelligence beyond it?

Thomas Scialom: 00:57:14 I mean let me show you my faults at the moment, but preliminary, let me say that it probably depends on the mood I am. I often change my mind. Five years ago I will have said yes, and I would say no, and I was always balanced. But also I'm bad at predictions there. I think the only thing that I'm sure that the unexpected is unexpected. I think actually five years before I kind of started my PhD, it was 2017, Transformer was there, GPT-1 was there. I was kind of doing working on summarization with [inaudible]. And I remember some slides where three words meaningless was kind of the summary I could obtain. So, again, if you have asked me the same question, will we be there now? I would've say clearly no. Actually I was even late to the party and all the scaling things, I realized very late how big it can be.

00:58:27 And now related to AGI, I think there's one question which is, do we have already all we need to get AGI? Is it just a question of compute FLOPS and scale? And will we get there in the decade with more investments, which we'll have, or not? And I don't have a strong conviction there, but I can tell you that, well, first I was bad at predicting the impact of scaling. Then I just watched a talk from Carl Schulman on YouTube where he clearly explains how for him scaling has a very important foundation and then even on the brain and human condition, and that could be it. And then there's a very important question I always ask when doing deep learning, is it just statistical correlation or is it more? And I'm always balanced on that. Sometimes it seems so good at reasoning and making it, and sometimes it's like mistakes are so silly.

00:59:37 And so actually there's a paper that I tend to be on the side that we could get AGI [inaudible] with scaling only. There's a paper from Harvard EML called Emergent World Representations exploring a second model train on synthetic task, published this year. And so this paper was notably like using some stuff on Othello-GPT where the idea you can feel about AlphaGo and all these things, but the idea here is not to get the state of the result. It's to train a model to predict the next token, which is the next move from human players. And that's it, just as a language model. And then the question is at the end of that, did he learn the distribution of the move as a stochastic part or did he learn more an understanding of the world? The world is the game. And they clearly found that the model, the transformer, train on that dataset kind of learn the world, the rules, the game, what it is, how it interacts beyond just a sequence of actions. And that is a clear signal that there's more profound understanding and that maybe from scale could emerge this intelligence.

Jon Krohn: 01:00:53 Yeah. Yeah. That is fascinating. So, I guess that's the answer I'd expect. It was quite a balanced answer. Maybe we will, maybe we won't.

Thomas Scialom: 01:01:05 But we're working on that.

Jon Krohn: 01:01:06 Yeah. Yeah. And I could probably guess that you are probably on the side that we should be trying to open-source these. If we can have AGI, I expect based on what you're doing with open-sourcing Llama 2, Toolformer and Galactica, that you would like AGI to be open-source as well.

Thomas Scialom: 01:01:35 Yeah, I mean I'm pro open-source. I'm pro not having in somewhere controlled by few people a very capable model. But at the same time, it doesn't mean we should rush open-sourcing such a big technology and some efforts on the other labs to put the bar very high and think forward about this and what it means and how we could prevent is very important and we should learn from that. And eventually we'll have some regulations and governance and we will have an open AGI. It's better than a closed AGI historically speaking. It always has been and always will be, but it doesn't mean we need to make it a responsibility.

Jon Krohn: 01:02:24 Nice. And that kind of responsible development and huge development of large language models is something that it goes back for you. We've talked in this episode about the stuff you've been working on recently at Meta, like Llama 2, Toolformer, Galactica and Unnatural Instructions. But this is something that goes back a while for you. So, you worked on BLOOM several years ago, which was at the time of GPT-2, GPT-3 era, BLOOM was the leading, I think, open-source kind of analog to those kinds of things. And your whole PhD was based on this kind of... Well, so I mean the title of your thesis was Natural Language Generation with Reinforcement Learning, and you developed a method called QuestEval. Is there any relationship between that QuestEval and the RLHF that you were talking about earlier, or is the reinforcement learning that you were focused on in your PhD different from RLHF?

Thomas Scialom: 01:03:29 So, somehow it has the same foundation in the sense that you want to maximize the reward. And so the question at the time we were maximizing some, the RLHF on natural language generation was based on some automatic matrix called BLEU or ROUGE. And people that know those metrics know how bad they are. So, basically the thing was you improve the score, but you [inaudible] the quality of the output. So, how can you develop new metrics that will actually capture more what we want? So maybe we can apply reinforcement learning on that, which was working pretty well. I developed enforcement learning techniques on one side, I developed metrics like QuestEval on another one. There's a paper that did reinforcement learning with QuestEval from IBM one or two years ago, and they reduced hallucination by 30%. So, it was working. Now maybe the algorithmic and definition with respect to RLHF are very close in terms of architectures implementation, math, but this philosophy of RLHF, which I discussed before about improving beyond the max of the human annotator is something that is quite different.

Jon Krohn: 01:04:45 Very cool. And prior to your PhD, you were involved in quantitative trading, so you were at SocGen, Societe Generale, which is something, I mean I wasn't at SocGen but something that you and I have in common is that I was also, before becoming a data scientist. So, in my case between my PhD and becoming a data scientist I worked for a few years as a quantitative trader working on algorithmic trading and I don't know, I don't know how interesting it is to go into financial applications or algorithmic trading applications with AI and LLMs. You're welcome to talk about that if you want to, but I think something that might be more interesting for our listeners is that you advise on and you invest in early stage companies that are focused on generative AI and LLM. So, we probably have a lot of listeners out there who would like to have a startup or scale it up. So, what kinds of advice do you have for people that are looking to start up or scale up a generative AI company? What kinds of problems should they be solving or what should they do?

Thomas Scialom: 01:05:58 That's a tough question. I mean I'm very good at advising them on the side of the research is like what is a trend? Where we'll be in one, two years? Is this technology far ahead or very major? And so that helps transition from research labs to applications quickly. I feel I have some ability to help them in this regard. Now it's especially difficult to predict where to invest right now in generative AI. There's kind of a paradox with this technology because we discuss the scale and use the velocity of the technology. You said that a few minutes ago and think about when I started data science deep learning, it was like data is a unit. And so then you have companies like Grammarly that annotate a lot of data, create with a deep learning model, train on these corporate data, some very strong models to correct grammatical error.

01:07:10 And this is a very strong technological barrier, because to beat them, to outperform them with deep learning, you need to annotate the same volume and the same quality. So, they're leaders. And now with the same kind of background technology, deep learning, what, one to three years later you have a model, a plug and play, ChatGPT, that you can just create a website in one minute to plug in on a Google Chrome that is even better than Grammarly to correct and much more general. And so all the technological barrier vanish in one second. And so the products with this technology that everything that we are saying now could vanish in one year. What I said before, it's expecting that the unexpected will happen. And so I guess the main question for entrepreneurs is what can you build that will be robust in this condition?

Jon Krohn: 01:08:10 Yeah. What can you build that will expect the unexpected?

Thomas Scialom: 01:08:15 That would be reinforced if there's some expectation.

Jon Krohn: 01:08:19 Nice. So, I guess that's the kind of thing people need to be thinking about with their moats. What is it? Is there some kind of data or some kind of market access that is unique? That means that even if much better generative AI models are open-sourced or could eat your lunch kind of thing, that you still have this opportunity. So, if you can get some kind of edge somewhere, then when these kinds of unexpected new things come out, these due AI capabilities, you can be integrating them into your tech as opposed to being eaten by them.

Thomas Scialom: 01:09:01 Yeah. And again, I don't want to get entrepreneurs worried, this is the risky and challenging environment, but at the same time it's one of the greatest moments for entrepreneurs to make some products. That's where comes the paradox. It's one of the best time to create, but also very risky.

Jon Krohn: 01:09:19 Nice. Very well said. All right, awesome. So, that is the end of my questions for you, Thomas, and the end of Serg's questions for you. So, let's turn to audience questions. I made a post a week before recording on social media on both LinkedIn and Twitter and the LinkedIn post in particular got a crazy amount of reactions, 250 reactions, over 70,000 impressions just at the time of recording here, which is definitely at the top end of the distribution for posts that I make. And we had a really cool one from Alice Desthuilliers who used to work with me at Nebula. She was an amazing product manager responsible for our data products and AI products. But Alice, I think your questions on unnatural instructions have already been answered earlier in the episode by Thomas. So, hopefully that answer was to your satisfaction. So, let's move on to a question from Adithyan.

01:10:22 So, Adithyan is interested in generally rough rules of thumb for how you choose what kind of open-source LLM to start with and how to fine-tune it. So, if he's building a startup for a niche use case using a large language model, some of his questions are around things like how do you decide what size to go with? So, I think I already actually answered this question earlier in the episode. So, with Llama 2 for example, the released model sizes are seven billion, 13 billion and 70 billion. And I talked already earlier how the seven and 13 billion, this can often fit on a single GPU. And so a small model with that could be good enough for a niche task. You'd only need the 70 billion if you wanted the model to be able to do a very broad range of possible answers, a very broad range of possible tasks.

01:11:23 So yeah, so I think in your case, Adithyan, with a niche use case, probably seven billion is probably going to be fine. You can start there. If it doesn't do the trick, try 13 billion. But the question then for you Thomas, is how many data points do you think that he needs to collect or somehow synthesize in order to be able to make use of fine-tuning? So, the implication here is that there's some niche use case that he would like the model to be able to handle. How many data points does he need to have in order to make use of a parameter-efficient fine-tuning approach on top of Llama 2 and excel in that task?

Thomas Scialom: 01:12:02 Right. It's an interesting question. I will start by saying maybe you can start even without fine-tuning just off the shelf with zero shots, but also with few shots, one, two, three, five examples you created yourself. I think it's not like a few shot pre-train model like it used to be before. It's a chat model. So, maybe you need to do a bit of prompt engineering in the sense that create a dialogue, like example one with your input, you make the model kind of generate your gold output and then when you will ask your question, the model is kind of biased toward the format, the kind of template you want it to be answered. That will the first thing I would try. If it's not enough, I would say that generally, and it's very hard to answer systematically because it depends on each use cases, task or difficulties, et cetera. But in general what I have seen is that with very few example sometime a hundred, a thousand at max, you can have dramatic improvements on some tasks.

Jon Krohn: 01:13:11 Very nice. Yeah, that's a really great answer. Very practical answer, Thomas, thank you very much. All right, our next question is from Svetlana Hanson, she's a senior software engineer. I believe she works on a lot of outer space projects with folks like NASA folks, that kind of thing. So, Svetlana has been following the Super Data Science podcast I think for as long as I've been hosting it, so several years now. And she's had some great guest suggestions in the past and she had a series of great questions for Thomas. One that I really liked was about the lessons that you've learned, Thomas, from developing and managing these large scale AI projects. So, being involved with BLOOM years ago, Galactica, Toolformer, Llama 2, these have huge team sizes and huge models, very long compute times and you kind of gave us a bit of an insight into this.

01:14:11 There's this pressure, this race, especially in open-source to get out there before other people. And so, for example, you made the decision when the 34 billion parameter model wasn't meeting the same safety standards as the seven, the 13 and the 70 billion parameter Llama 2 models, you said, let's just go ahead and publish what we have because we've got the state-of-the-art of the 70 billion, we've got the smaller models you can fit on a single GPU. So, we've had some kind of insight into your thinking on these kinds of large scale projects. But yeah, I don't know. First Svetlana's question here, what other key lessons have you learned about developing and managing large scale AI projects?

Thomas Scialom: 01:14:47 Yeah, it's a very interesting question. Let's try to sound smart on that one, but maybe the main difference with these big projects with respect to when I was in academia with some small papers with very few people, because of the size, it also means a lot of people are impacted. There's a lot of budget around and you have a potential to reach out also much more people, the project is at another scale of impact. And because of all those ingredients, well, it was a case for probably BLOOM and even more Galactica where I was even more involved in the training and the project were way fewer, but you have a lot of GPUs veterans, you have to make some decisions. And the thing is there's a main difference with let's say in a perfect world for researchers as I am, you want to understand everything, all the phenomenon.

01:15:50 And so you want to do all the ablations, you want to do all the experiments to see what's the impact of this factor and this one and what if we have done that. The thing is there's so many possibilities and every experiment costs so much and takes so much resources that you cannot do that anymore. And so one of the main challenges, you're responsible to make some decisions as I was in Llama 2 of like, okay, we need to choose between that and that. The thing is even more because no one is publishing anymore, [inaudible] maybe just we did. You're like, okay, I don't know what's my intuition? How can we very quickly verify and change if needed? And you are playing with actually a lot of resources like millions of dollars, some mentioned for the annotation of a lot of thousands of GPUs, many [inaudible] that are involved in the project. And time is also a constraint on resources and you cannot spend one year to explore. And so how to deal with this challenging environment is what I thought was the main challenge. And when you are at night and before sleeping, you took a decision and is it the correct one or not? And you don't know. And this uncertainty for researcher is something hard to deal with.

Jon Krohn: 01:17:15 Nice. So, I guess your answer, your key lesson is that there's trade-offs and you don't know whether you're making the right answer and maybe these decisions on how quickly do we rush this or spend some more time on it. Well, it seems like with Llama 2 you certainly got it right, it made an enormous splash and a huge impact. So, you seem to be getting it right. We've got a comment here from Laurens van der Maaten who was recently on the show, episode 709, a colleague of yours at Meta and he doesn't have a question, but I just wanted to highlight that he said, "Thomas is awesome. I'm looking forward to hearing your conversation with him."

Thomas Scialom: 01:17:59 Thanks Laurens.

Jon Krohn: 01:18:00 Yeah, so Laurens, I hope that you enjoyed this conversation as much as you were hoping to. And then last question here is from SM. So, SM has asked questions on the show before, but SM has a, I assume, very deliberately sparse LinkedIn profile, which is unusual. Most people on LinkedIn, it's like real names and that kind of thing, but SM seemingly the account exists solely to ask questions on the Super Data Science podcast, because there are no other connections. So, I appreciate that compliment. So, SM's question is, it's a long question, but I think it's basically getting at this idea of LLMs can be wrong, they can make mistakes, they can give unhelpful answers, but nevertheless they are often very useful and they're becoming more and more useful all the time. So, I guess this question is, I think we touched on this a little bit earlier in the episode as well when you were talking about the research at Harvard and the ability for transformers to seemingly understand... Understand is such a bad word, but-

Thomas Scialom: 01:19:18 I understand. I would agree.

Jon Krohn: 01:19:19 Yeah. So, I think you have a good sense of the question of where this is going, so you can answer it.

Thomas Scialom: 01:19:25 Yeah, I think the question if I understood it correctly is about can we one day in the future, and how is it not yet the case, rely on these models? Isn't that humans on some very simple tasks obtain a 100% score, while models will sometimes do so impressive things and when it's not expected will fail on silly things. And so that's very weird. My understanding, and I'm not saying I get it right, but just my intuition understanding at the moment is that as we discussed before with scale, we might have an emergence of much general reasoning and understanding. And my understanding is that those algorithms learn the compression of the data.

01:20:20 Maybe let me give you an example to understand, one or two examples. I can print you an infinite number of tokens to train the model of numbers and calculus, one plus two equal three and so on. Now, if I give you that, there's two ways to predict the next token after equal, you can memorize everything, but if it gets to an infinite vocabulary, you will need a lot of weights to memorize it. Or you can compress the information so that you internalize the algorithm beyond that and so you can predict accurately the next token whatever, and that requires much less weights to learn calculus than to memorize an infinite number. Now, at the current time, it seems that large language model are very good at doing calculations for one, two, three digits, and where it goes beyond it fails more and more.

01:21:22 My understanding is that they cannot internalize generalization in term of calculus for one digits, but somehow in the large vector space they kind of see it as different objects calculus for two digits than for four, five digits, maybe because it appeared less. And so they don't have yet this generalization of, "Oh, this is one, two, three, five, six, seven, eight is calculus, and so nine and 10 that I didn't see in the training is calculus as well." And so there's one dimension I didn't generalize, but there's some of others they already generalized. And I feel like true AGI, if we get there with scaling or in any other ways called from this generalization of compression at another scale when the generalization will be complete somehow if we get there with scaling.

Jon Krohn: 01:22:14 That was an amazing answer. That made it so crystal clear and really built nicely upon what you said earlier in the episode around representing these complex concepts. Very, very cool. All right, Thomas, it has been an amazing episode. I've learned so much. Truly, it's been an honor to have you on the show. Before I let you go, do you have a book recommendation for us?

Thomas Scialom: 01:22:39 Maybe Black Swan from Nassim Nicholas Taleb.

Jon Krohn: 01:22:43 Nice one, yeah. Great choice. And how should people follow you? After this episode if people want to keep up with the latest on your work or your thoughts, how should they do that?

Thomas Scialom: 01:22:57 Sure, they can follow me on LinkedIn at Thomas Scialom or on Twitter as well. I'm really easy to find there.

Jon Krohn: 01:23:03 Nice. All right. We'll be sure to include those links in the show notes. Thomas, thanks again and best of luck. We can't wait to see what you release next. Some stuff probably it sounds like even before this episode is live. And truly on behalf of my listeners and tons of other early stage startups like mine, we are so grateful to have people like you and Meta being willing to open-source these incredible technologies. It's making such a huge impact

		commercially and also big social impact, so thank you very much.
Thomas Scialom:	01:23:40	Thank you, Jon, for having me and for all the kind words. It was my pleasure.
Jon Krohn:	01:23:49	Thomas is already a legend, but it seems he's only just hitting his stride and his biggest most mind-blowing potentially AGI summoning projects are yet to come. In today's episode, Thomas filled us in on how pre-training and fine-tuning an LLM on an as yet unprecedented scale for an open-source LLM led to the big Llama 2 splash. He talked about how handling code, tools, web search, and even better performance are up next for the Llama project, how Toolformer calls an appropriate API and it incorporates the output into its next token predictions, how RLHF shifts the distribution of a pre-trained LLM model's outputs from a normal distribution of human generated quality, to outstanding, often superhuman quality, and how with AI developments the unexpected is expected. And so AGI may be just around the corner.
	01:24:38	As always, you can get all the show notes including the transcript for this episode, the video recording and materials mentioned on the show, the URLs for Thomas's social media profiles, as well as my own at superdatascience.com/713. Thanks to my colleagues at Nebula for supporting me while I create content like this Super Data Science episode for you. And thanks of course to Ivana, Mario, Natalie, Serg, Sylvia, Zara, and Kirill on the Super Data Science team for producing another tremendous episode for us today. You can support this show in so many ways. You could check out our sponsor's links, you could share it with a friend or colleague, you could review an episode, you could subscribe, but most of all, just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until

next time, my friend, 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.