SDS PODCAST EPISODE 733: OPENASSISTANT: THE OPEN-SOURCE CHATGPT ALTERNATIVE, WITH DR. YANNIC KILCHER

Show Notes: http://www.superdatascience.com/733

Jon Krohn: 00:00:00 This is episode number 733 with Dr. Yannic Kilcher, CTO at DeepJudge. Today's episode is brought to you by Gurobi, the decision intelligence leader, and by CloudWolf, the cloud skills platform.

00:00:18 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.

00:00:49 Welcome back to the Super Data Science podcast. Today's guest is one of those huge names in our field that it blows my mind that I get to talk to him at all, let alone record a deep and fascinating conversation with. If you're not already aware of him, Yannic Kilcher has over 230,000 subscribers on his machine learning YouTube channel. He's the CTO of DeepJudge, a Swiss startup that is revolutionizing the legal profession with AI tools. He led the development of OpenAssistant, a leading open-source alternative to ChatGPT that has over 37,000 stars on GitHub. That's crazy. And he holds a PhD in AI from the outstanding Swiss Technical University, ETH Zurich.

00:01:27 Despite being such a technical expert himself, most of today's episodes should be accessible to anyone who's interested in AI, whether you're a hands-on practitioner or not. In this episode Yannic details the behind-the-scenes stories and lasting impact of his OpenAssistant project, the technical and commercial lessons he's learned while growing his AI startup, how he stays up to date on machine learning research, the important broad implications of adversarial examples in machine learning and where the biggest opportunities are in AI in the coming years. All right, you ready for this terrific episode? Let's go.

	00:01:58	Yannic Kilcher, welcome to the Super Data Science podcast. It is unreal to have you here, a dream come true, truly. Where in the world are you calling in from today?
Yannic Kilcher:	00:02:14	In Zurich. Thanks for having me.
Jon Krohn:	00:02:17	Nice. Yeah, my pleasure. I had a couple Zurich trips earlier this year and they were both perfect. One was for skiing and had absolutely perfect skiing conditions in Klosters Davos, and that was incredible. It was like blue skies, but lots of snow, not too cold, just what you'd want. And then I was back in St. Gallen for the St. Gallen Symposium in the spring, and it was glorious warm days, you could be outside during the symposium. And they said it had been 10 years since that had happened.
Yannic Kilcher:	00:02:54	Really? Okay. Londoners always complain about their weather, I personally feel like Switzerland weather is nearly perfect, at least for me.
Jon Krohn:	00:03:07	Yeah, I thought you were going to go the other way.
Yannic Kilcher:	00:03:08	No, it's cold in winter. It's nice in summer. It's rainy in fall, which is really nice after a warm summer.
Jon Krohn:	00:03:18	Yeah, you're preaching to the choir here. I have this vague fantasy of retiring in Switzerland, so it's a tricky passport to get though.
Yannic Kilcher:	00:03:27	It is.
Jon Krohn:	00:03:30	Cool. Yeah. So let's jump right into the technical content we have for you. We have tons planned. So first I'd like to talk about your OpenAssistant. So in April, you, along with a team of multinational ML practitioners, you submitted a paper called OpenAssistant Conversations, Democratizing Large Language Model Alignment. So can you explain what OpenAssistant Conversations is? Yeah,

and what does democracy and alignment have to do with LLMs?

Yannic Kilcher: 00:04:02 OpenAssistant was a project that it was born out of a desire to replicate ChatGPT. When ChatGPT came out, there was really nothing in the open-source space that was even remotely towards that direction. People had some ideas and there were some efforts of, hey, let's just use ChatGPT to create data for something, but there was really nothing. And I think the idea of, hey, let's make something in open-source, it's fairly straightforward. I think we just grabbed the momentum and organized around that, and the main part of it is data collection. So we knew that in order to get ChatGPT to be this assistant-type model, they had to have collected some data from humans in order to do that. We were going off a paper called InstructGPT that was released a few months prior that showed considerable benefits of this collecting human data first and foremost, and then fine-tuning on that and then doing the reinforcement learning on top of that.

00:05:24 And the crucial part I think that was not available to the open-source community was, well, first of all, the base models were not available, and second of all, mainly the data sets were not available. So we knew that OpenAI has gone full business and didn't even say how they did it exactly. So our mission was to collect human data. And we built that. We built a platform, we built software where you could submit your data and people came and contributed data. And we also trained some models on top of that obviously. And those became cool chat assistants and so on.

00:06:08 I would say it was a really special time. It was a time and place where the momentum to gather such a data set was really given and people came and contributed. And I think we realized that and we wanted to make the best of it. We

wanted to make the best of that momentum, and I think we captured that. And I think what lives on from the project is the data that we collected. The models, they're fun, but other people can train models too, right? Anyone nowadays especially with super cool open-source space models and low-rank adapters and whatnot, it's super straightforward to train a model. So I think the main success of the project is the data we collected and are still collecting, actually. Although, I mean obviously interest has waned let's say and people move on.

Jon Krohn: 00:07:08 Yeah, it was. It did make a huge splash when it came out. It was something that it seemed like everyone was aware of your OpenAssistant release and you were the face of that in some ways. I know that there were a few dozen contributors, but yeah, your announcement on YouTube was the first place that I noticed it. And-

Yannic Kilcher: 00:07:29 Yeah. And to be said, I'm mostly the noisemaker. So there are people who have also contributed considerably more work than I have, so I don't want to take any credit away from all the rest of the contributors. Mostly I use serve the platform that I have to popularize it.

Jon Krohn: 00:07:50 I see. And one of those people is Lewis Tunstall, right?

Yannic Kilcher: 00:07:54 Yeah. So I mean, we've had contributors from all over, including Lewis from Hugging Face. And in fact, Hugging Face has also sponsored the project quite a bit, especially once we actually trained model and then had to or wanted to make them available to people to try them out. Hugging Face was a big sponsor of that as well. And so the people and the company were contributing, which is very cool.

Jon Krohn: 00:08:21 And so he was in episode number 695, and it's an absolutely amazing episode on using transformers for NLP, as you'd expect from him as a great author on that topic. Yeah. So how optimistic are you that as these

systems get more powerful, whether they're open-source like yours was or their proprietary, how much of an issue do you think alignment is in the coming years?

Yannic Kilcher: 00:08:52 I guess it depends a little bit on what alignment exactly means then. So I think the variant of alignment where this is an assistant and it should do what you ask it to do, I think that's just from a practical perspective, it's super valuable to not have to put huge efforts into prompting exactly the thing you want, but just like you had a real human assistant, be able to just give it a task and it'll kind of understand what you want and do it right at the right level of abstraction with the right amount of volume. Not too much work, not too little work, that type of alignment, I'm a big fan of.

00:09:43 Then there's the other type of alignment where we can say, well, can we align it such that it, I don't know, doesn't destroy us at some point or has our ethics or something like this? I don't really know how to engage with that discussion necessarily because it tends to go off into super philosophy mode almost instantly, like after half a sentence you're in, "Oh no, but the Mesa optimizer can self optimize beyond transhumanism." I don't know how to engage with that, honestly. So I mean, for sure it's probably better to have a business friendly model that complies with human rights and proper language than it is to have not. But beyond that, I really don't know.

Jon Krohn: 00:10:38 Yeah. That is something that I see you personally, and I don't know how much this is, I don't know exactly where I picked this up, but I have this impression, I see you as actually one of the leading figures in banging the drum or being lead noisemaker, as you say, of open-source, LLMs and the philosophy, we don't want to get into philosophy, but the philosophy that open-source is better than proprietary LLMs. There's obviously two big camps on that. So some people would say, well, proprietary is better

because that means that if these systems become really dangerous, then if they're proprietary, we can control who's using them. And then the opposite camp, which I believe you are in, it's like, well, we should be open-sourcing everything we can. That way, just like any other open-source software project, you can have as many eyes as possible understanding where the risks are in this system.

Yannic Kilcher: 00:11:49 Yeah. And I think history has pretty much in almost every single case been with the second camp. What strange idea is this that if it's proprietary, we'll be able to control? No, if it's proprietary, the proprietary will be able to control the rest. Like now, every big regulation that comes out is completely drafted and lobbied by the big proprietary providers of whatever is being regulated. There are people who benefit personally greatly from banging the doom drum on this. On one, it's certainly the big companies like OpenAI being like, "Oh, no one else should have it." That's their take. So we should introduce it as much regulation as we can, so we make sure that no one catches up with us essentially.

00:12:52 And then the other group of people are just the general people who get clout because they have really out there opinions, all the doomers and all the forever criticizers and so on. And it's a pretty straightforward function. I point out flaws in LLMs, I get clicks, I get money. There are so many people who have a vested interest in being like, oh, doom, doom, doom. That I think it needs to be adjusted at the receiving end.

Jon Krohn: 00:13:36 Gurobi Optimization recently joined us to discuss how you can drive decision-making, giving you the confidence to harness provably optimal decisions. Trusted by 80% of the world's leading enterprises, Gurobi's cutting-edge optimization solver, lightweight APIs and flexible deployment, simplify the data-to-decision journey. Gurobi

offers a wealth of resources for data scientists, webinars like a recent one on using in Gurobi Databricks. They provide hands-on training, notebook examples and an extensive online course. Visit gurobi.com/sds for these resources and exclusive access to a competition illustrating optimizations value with prizes for top performers. That's gurobi.com/sds.

00:14:21 Yeah, I get you. It makes total sense to me. What do you think about the Meta open-source efforts? I think, for me personally, it's been a really cool thing to see, and it's made a huge difference in my personal perception of Meta because prior to all these open-source releases this year, things like Llama 2, and apparently they're working on something much, much bigger. So we recently had Thomas Scialom, I don't know if you know him. He's in Paris and he works at Meta and he was the final author on the Llama 2 paper. So he's like the lab head, kind of equivalent in academia. And he was clear that what they're working on next, the Llama 3 will be much, much bigger, much closer to a GPT-4 kind of size and capability.

00:15:17 And so I've been turned around, I've done a 180 on my feelings of Meta based on these open-source efforts. I was a year ago, two years ago with their stock price going down, I was like, yeah, I don't know, of all the big tech companies, I don't know, I felt like they were, I don't know, the one that was kind of doing the least positive in the world, and I wasn't cheering them on. But these open-source initiatives have been huge. It isn't open-source like your OpenAssistant, their open-source is in quotes because they're just giving you the model weights.

Yannic Kilcher: 00:16:02 Although I mean, the OpenAssistant models are also, some are based on Llama, so we have to just replicate that license, but the data is fully, I believe it's like creative commons, some appropriate. But yeah, it's a strange

world where Facebook essentially is becoming the most open company and OpenAI is becoming the most close. It's a bit like soccer teams because the players, they swap out every two or three years. So the team is entirely different. It just carries the same name, and for some reason people are like, yeah, yeah, Arsenal or something.

Jon Krohn: 00:16:44 I know. Yeah, it's so weird.

Yannic Kilcher: 00:16:48 And with the companies, I mean, I'm sure that the researchers at Meta and Yann LeCun among them and others have been pushing hard for this approach of open-sourcing this. And it's a strategy, right? Because now they can build a platform where people can contribute their apps with, I believe that's maybe what they want to do with Metaverse and things around AR and so on, where they want to say, "Hey, how about we just give these tools to people so they can build cool apps so that then they can put these onto our platform." That was always the point in making TensorFlow, making PyTorch open-source, making all of these things open-source will give these tools so people build applications that we can then serve on our platforms and things like this, and it's really cool. Yeah, it's absolutely nice to see that there's so much open stuff coming out of Meta.

Jon Krohn: 00:17:48 Yeah, it's a great example of the way that doing these kinds of open-source things can really lock you into a product is the CUDA Library.

Yannic Kilcher: 00:17:56 Yeah. Well.

Jon Krohn: 00:18:00 It's wild, the monopoly that Nvidia has on GPUs, and a big part of it is that we expect things to run on CUDA, and so it makes it hard to switch that hardware. Thankfully, it seems like PyTorch and TensorFlow, these are more agnostic to this specific application area. But I absolutely agree with you that the kinds of things, I'm

sure Meta is interested in from a proprietary perspective to be able to be getting the same kind of whatever, 30% revenue of every app, like Apple has with their App Store. I mean, this is a really lucrative thing to be forcing people to. You have an iPhone, you've got one App Store, all the apps have to go through us, and we get a pretty sizable chunk of everything. I'm sure they're salivating on that idea for the post-iPhone world, which yeah, I mean, seems reasonable to think that a device like a wearable device, an AR device, VR device, especially with the ability that these systems have, these LLM systems have, for interacting with us verbally in natural language, it seems like that's a reasonable direction for next.

Yannic Kilcher: 00:19:12 Yeah. It's just also a strategy that's kind of undermining the competitors. So OpenAI, Google, all of these places, they're throwing enormous amounts of money into developing these things. And Meta is just like, here's open-source, which just takes away so much of the market of these bigger language models, or at least it drives their pricing down. Because otherwise people will be like, "Well, for my application, actually Llama 2 is completely fine. So I'll just use that." Just as sort of a tactic, it's already working out, I think.

Jon Krohn: 00:19:56 Yeah, absolutely. We use Llama 2 in our machine learning company, and it has been great. We love it, and it's great how they have, at least at this time, released three different model sizes. So if you want a 7 billion, 13 billion parameter model, those you can fit on a single GPU. And then they've got the 70 billion parameter model, which we've never needed, we've never even tried, we've never even downloaded those model weights because for the tasks that we need, usually the 7 billion is fine. So yeah, so speaking of size and scale, in the OpenAssistant Conversations paper, you wrote about how these LLM advances that we've had in recent years follow a really straightforward formula, which is just scaling up,

so having more transformers in the architecture and then needing a larger training corpus to correspond with that. So kind of following the chinchilla scaling law idea. So yeah, do you have any ideas on what will do other than scaling? What are alternative formulas for having better LLMs?

Yannic Kilcher: 00:21:10 I have no idea, honestly. I mean, surely someone can come up with a different architecture or so, although I don't think that's going to make a giant difference. So I think a couple of works have shown that it doesn't really matter what architecture, like MLP mixer and things like this. So it seems like as long as you have something that you can actually scale without hitting gradient saturations or things like this, just something with lots of parameters that you can train with a lot of data efficiently, then that will get you to a low perplexity on language modeling. So I don't have too much, let's say, hope there. I mean, maybe we'll make some advances in long context and things like this. I think we will continue scaling until it is no longer physically possible, and then some more before we invent other things. It's like CPUs, right? People will be like, oh no, 13 nanometers is really the end, and now we're at what, three?

Jon Krohn: 00:22:24 Three. Yeah.

Yannic Kilcher: 00:22:27 And plus, once we hit that limit, then we go to, ooh can we do multi-core? Can we do blah, blah? Can we add caches, more cache? And so in general, the big models, I think it's going to scale for a while, and I hope what's going to happen is that we build additional modules, especially around memory, around things like lifelong learning and things like this, but these are going to be additional systems that are on top of these models, working with them rather than fundamental changes to them.

Jon Krohn: 00:23:11 By lifelong memory, you mean something like an episodic memory where it's like maybe writing to disc as opposed to trying to store it in weights or in context?

Yannic Kilcher: 00:23:18 Yeah, exactly. So it's like you note taking, right? You're learning something, you take notes. So that's one part, is that memory that you store explicit things in. And the other thing is the lifelong learning where you continue improving, even as you do inference. Right now we have training mode, then we have inference mode, and if we want something more, we go back to training mode. But a human just goes through life continuously, essentially doing inference and training at the same time. And I think we're yet a way for these models to do that because we have catastrophic forgetting and blah, blah. It's just not a technical, either engineering or from the machine learning perspective really feasible to do that now. But it's definitely where things should go because if you have a good assistant, what you want it to do is you want it to learn from you over time to work better with you together over its whole life.

Jon Krohn: 00:24:22 This is a little bit of a tangent, but I don't think I've talked about this film on air and I think it's just awesome. In 2021, there was a big budget film called After Yang that came out, and it stars Colin Farrell. He's the most noticeable actor in it in terms of fame. And it is about, I'm not giving you a spoiler here because this happens in the opening scene, it's this family that has an AI robot, like a physical, like a humanoid robot named Yang. And in the opening scene, the family's doing a dance competition live on TV. It's this cool idea of in the future having families dancing together in front of their TV, and it's like a competition and somehow a machine vision algorithm or whatever can judge each of the families on how well they're doing in the competition.

00:25:19 And this android, this robot Yang, it has an error, a fatal error. Well, it's doing that dancing in the beginning, in the opening scene of this film. And so the film is called After Yang, and the rest of the film is Colin Farrell first trying to repair it and then investigating its episodic memory that it had accumulated. It's a beautiful, beautiful story. I cry very easily in films, but this one had me like, oh my God, because it was a really cool thing about this, it turned out, now I would kind of be spoiling. But basically it turns out that Yang has been around for a long time and he's developed a lot of highly emotional episodic memories from over these decades that it had been, I guess, in operation.

00:26:22 Yeah. Anyway, that was a complete tangent, but an interesting, a great film that I recommend to... I suspect that any listeners of the show, given your interest in machine learning and AI, it is certainly not as famous as a movie like Her, but I thought it was excellent and a really interesting perspective on where things could be going in the coming decades with the kinds of advanced AI systems that we have. And very different from Her in the sense that with Her it's a hundred percent software, whereas with this one is the hardware component as well. Anyway. Yeah. So what do you think, Yannic, in terms of making sure that these kinds of open distribution of AI models like we have with Meta, like you had with your OpenAssistant, what do you think are the best things that we can be doing to encourage that ecosystem to continue to develop? I guess contribute to open-source ourselves?

Yannic Kilcher: 00:27:24 Yeah, although I mean it's become harder, right, since giant companies and entire nation-states like Arab universities backed by their governments come into this. So there's definitely a lot of hacking still to do. And that's what I really love about the open-source community is if you see... I mean, for example, if you see things that were

done with CLIP-Guided-Diffusion. Just because OpenAI announced DALL·E and CLIP together, but they only released some weights for the CLIP model and no weights for the DALL·E model, and you see the inventiveness of the open-source community what they can do if you give them the tools. This whole CLIP-Guided-Diffusion area was completely based on the fact that OpenAI didn't release DALL·E. And people were like, "Ah, we still want to do cool pictures." So yeah, I think there's a lot of hacking and things to do for the individual people, but they're more going to be in the domain of making creative use of the stuff that is released.

00:28:47 And I think, I'm not going to say do that because people who want to do it, they're already doing it. They're naturally drawn to things like this, but just... I don't know, try to think outside the box and try to think of what is a weird way I can use these model weights, let's say, that OpenAI would never do, never think of doing, and I don't necessarily mean unethical or so, but just... CLIP-Guided-Diffusion, to me, it's one of these examples where that's just a way of using it that's not... OpenAI would not do that. Or very, very creative applications. For example, I've seen one where you give a few words and it gives you a color palette for it. And that was really early on. I think that was also CLIP-Guided-Diffusion, if I'm not... No, that was stable diffusion. But I mean it was trained to produce images, but then people were hacking it and building things on top with averaging to give you a color palette for... And you put some mood words. It's just so creative and no big company product company is going to think of those things. So I think, think of what you could do special that is not super obvious with things like this. I think there lies the beauty of the open-source community.

Jon Krohn: 00:30:32 Data science and machine learning jobs increasingly demand cloud skills, with over 30% of job postings, listing

cloud skills as a requirement today and that percentage set to continue growing. Thankfully, Kirill and Hadelin who have taught machine learning to millions of students have now launched CloudWolf to efficiently provide you with the essential cloud computing skills. With CloudWolf commit just 30 minutes a day for 30 days, and you can obtain your official AWS certification badge. Secure your career's future. Join now at cloudwolf.com/sds for a whopping 30% membership discount. Again, that's cloudwolf.com/sds to start your cloud journey today.

00:31:10 Nice. Yeah, that makes perfect sense and that's a really nice way of phrasing it to be doing weird things, that aren't illegal that OpenAI wouldn't think of. So switching gears a bit to what you're doing at your company. So you are the CTO of a legal tech startup called DeepJudge, which is based in Switzerland, and it's got tons of really high powered PhDs in AI and NLP working in the company. It seems like an awesome place to work. And you are taking these kinds of technologies, large language models, and applying it in industry specifically to legal document processing and searching. So your product called Knowledge Search is using, I assume, and you can let us know without divulging proprietary secrets, using things like open-source LLMs probably to be able to do search over legal documents in a way that nobody else can today, or none of the leading systems because big law firms have millions of documents. But I suspect that the best that I suspect that the incumbent approach is just a dumb keyword-based search.

Yannic Kilcher: 00:32:38 That's exactly right, yeah. Law firms, legal professionals and so on, they've just organized around the fact that they can't really search well and really use their old knowledge in the documents well. And yeah, I think that's poised for a change and we're doing that. We just enable better search working and then applications on top of that obviously. So the issue is really scale. If you want to

process million... It's really, it goes into billions of documents, those hundreds of millions, up to billions of documents that reside in these companies. And if you want to process those, you really need to think of scale and of plumbing, essentially, so that you have very high throughput. There's a lot of concerns of data privacy, data sovereignty. There's obviously attorney-client privilege that needs to be maintained. And then there's just it's real data. So you'll find the same document in three emails and in 10 different versions laying around somewhere and you have to handle all of this.

00:33:58 And then, yeah, it's a nice mix of machine learning challenges, but also just of practical engineering challenges that happen when you go to the real world, which I really haven't seen in academia much before because in academia you always have your nice data set and it's nicely split, it's nicely cleaned, it's nicely labeled, and all you want to do is kind of get the number up at the end. In industry, you don't even have a number, you don't even know what to evaluate. So yeah, that's really different and it's really cool to be doing that.

Jon Krohn: 00:34:42 Nice. How did this particular application area come about for you?

Yannic Kilcher: 00:34:46 It was more or less random. We were at the end of our PhDs and we did a project at university that was in the domain of legal tech, and we got into contact with some lawyers and we just realized that there's a lot of things to do and it would be a nice fit for the tech because legal documents are largely language-based, other than, let's say, financial documents, are largely based around tables with numbers in it and so on. That's not so good for language model or statistical language models. So legal documents is one of the domains where certainly I see a fit really well. Yeah, so we embarked on this. That was even before ChatGPT, and so.

Jon Krohn: 00:35:37 Yeah, I totally get it. I am on very much the same page as you because my company Nebula, what we do is a similar kind of semantic search, but over human profiles. So for ours it's talent acquisition or sales lead acquisition. So using publicly available natural language on people that we can find. And then you can put in skills that you're looking for, job title. You can use natural language to search over our database in seconds, do this kind of semantic search and pull back profiles. In your case, you're pulling back documents-

Yannic Kilcher: 00:36:12 Correct.

Jon Krohn: 00:36:13 So yeah, you're preaching to the choir. I get that there's opportunity here for sure. There's something called colinear technology that has been talked about in the context of what you guys are doing. And so I actually hadn't heard of that before, but it seems to kind of blend the semantic search that we love as data scientists that is NLU, natural language understanding, so using things like transformer architectures to create embeddings that we can search over. So that's the semantic search that we've been talking about so far since we've been talking about DeepJudge, but this colinear technology seems to allow you to have a semantic and keyword search together. Can you fill us in on that a bit more?

Yannic Kilcher: 00:37:02 Yeah, it's a term we came up with to have sort of a name for what we're doing. I guess sometimes it's called hybrid search or something like this, but what we mean by that is just that our index structures align the more frequency type information with the more neural type information because you want both. You don't want just the keyword type search or the frequency-based search, and you also don't just want the semantic search. Sometimes you just care about words and not just feelings. Sure you can say, well, we'll just get embeddings that are so good that they capture every nuance of everything. It's still a bit out of

scope, I believe. So for some applications, pure vector-based search is really good. Actually for many, especially anything around product search or anything around Amazon, product reviews and so on. All of these things, semantic search works wonderful.

00:38:14 For our applications though there are a few weaknesses and that's why we have to blend the two things together and that's where it gets into our more proprietary tech in how exactly we mix the two. And we enable retrieval with a combination, not just of one or the other. Because what most semantic searches, they either retrieve vector-based, or they retrieve actually keyword-based and then kind of re-rank with a re-ranker, things like this. So colinear technology is a name we gave to the thing that we do because we didn't know what else to call it.

Jon Krohn: 00:38:57 This makes a lot of sense to me for your use case in particular, where there would be situations like a specific client name or company name or event where a semantic search might be a little bit fuzzy on those things and find somebody who's related to that specific person in some way, this second person often shows up in the same documents as the first person. And so the kind of semantic search says, well, the second person is pretty much the same as the first. But for a legal document you're like, well, no, actually the first person is the person we need. The second person doesn't matter.

Yannic Kilcher: 00:39:35 Yeah, and especially if you don't train on the data that you search over, which is very often the case because you train on a subset or you train on a different set and so on... You can't possibly have seen all the data in training. So even now if a name comes in, you don't even know that name appears with the other name, you just know it's a name. If you do semantic search, you'll just get a semantically similar thing, which is another name, which

isn't very helpful. So some of these things are just weaknesses of doing embedding-based search, let's say.

Jon Krohn: 00:40:15 Yeah, yeah, yeah, that's a great example there. So you mentioned in a recent interview that DeepJudge users are quickly learning how to craft effective LLM prompts. That's an interesting thing that I don't talk about much on the show. We don't really talk very much about prompt engineering. And we don't need to spend a huge amount of time on it, but how have you found your users... How easily have they shifted from say, the keyword-based search that they're used to for legal document search into this LLM based approach that requires prompt engineering? Yeah, so what are the prompts like? How do you maybe nudge your users in the direction of making effective prompts? Or is it intuitive? Do they figure it out on their own?

Yannic Kilcher: 00:41:00 No, they actually come... They've been using ChatGPT and things like this, so they actually come with that. When we say, "Hey, this is AI-powered search," they come with the expectation that they can start a sentence by, "You are a professional lawyer and I have a client." And I think the legal profession is definitely a profession where these technologies can make a big impact. And people have been realizing this and have been using these technologies just as they came out, right? Hey, draft me a memo, draft me something. As long as they don't put confidential data into the ChatGPT interface, they're actually totally fine using it and letting it help them draft things.

00:41:49 And so there has been absolutely no education needed on our part. There's more like a bit of anti-education to not do that too much rather than... And it's the same with keyword search as such. Google has sort of taught people how to search, because if you observe yourself typing things into Google, it's really weird. It's really weird what

people type in there. They type like half a phrase and then half a different phrase, and then they type how to. And if you look at that piece of text, it's so dumb, but we know that's kind of Google-ish, and we know that Google can give us good stuff and then Google in turn will go, we'll observe what people do and be like, can we serve these people better? And we'll adjust... So it's this dynamic where just the tech teaches people and the people teach the tech.

Jon Krohn: 00:42:50 Mathematics forms the core of data science and machine learning. And now with my Mathematical Foundations of Machine Learning course, you can get a firm grasp of that math, particularly the essential linear algebra and calculus. You can get all the lectures for free on my YouTube channel, but if you don't mind paying a typically small amount for the Udemy version, you get everything from YouTube plus fully worked solutions to exercises and an official course completion certificate. As countless guests on the show have emphasized to be the best data scientist you can be, you've got to know the underlying math. So check out the links to my Mathematical Foundations and Machine Learning course in the show notes or at jonkrohn.com/udemy. That's jonkrohn.com/udemy.

00:43:29 Yeah, for sure. With our Nebula platform, we have recently made some advances, like our most recent release at the time of recording allows our users to ask more natural language questions like you're describing, like you're a great lawyer and help me draft a document. But up until very recently, our search worked in a way where, okay, so let's say you're looking for a data scientist with experience with LLMs and PyTorch. Then a great way to query that in our system, and it's been a big lesson for me, is that just because it's easy for me to understand that, then the right thing to do in the search box is type data scientist, LLM, PyTorch and enter. And to me, that's

really obvious and you're going to get really great results back, but our users come in and they say, find me a data scientist in New York.

Yannic Kilcher: 00:44:40 Yeah.

Jon Krohn: 00:44:44 And so we've had to adapt because that is disproportionately because of, as you're saying, people have the ChatGPT experience. When you tell people that it's an AI-based search, then they assume that they can direct it in natural language. Yeah, this has been something... It's a lesson that I actually wish I'd learned earlier. I wish that I'd been figuring a way to make this work as opposed to... my solution previously was always like, well, we'll put in a demo video. Nobody watches it because you're like... I'm like, this is really easy. You just need to... The Google example that you gave there reminds me of this, which is that with Google search, people have had two decades to figure out that, okay, putting "how to" on the end works and blending two different phrases in this semi-natural language works really effectively.

00:45:35 And so my kind of... I've been pushing too hard on the product team saying, people will get it. They just need a little bit of experience, which is true. It's like with the way that our platform worked, for, say, the preceding year, where I'm expecting data scientist, LLM, PyTorch enter. With that... People start to figure that out after five searches or 10 searches kind of thing. But of course, in a demo or a sales call, you want the person to be able to type in whatever they want right away, their first time using the platform for free and get amazing results without having to think about it. So that's been a huge learning for me. And it sounds like you've already figured that out at DeepJudge in terms of having it be a ChatGPT-like natural language experience.

Yannic Kilcher: 00:46:21 Well, we haven't figured it all out, and it's certainly the case that people are still searching very different ways. We just expand what we can support of these. I think we have very much the same experience in that.

Jon Krohn: 00:46:35 Yeah. Yeah. So this is maybe a bit of a tricky question, but predicting things about the future is always hard, but given how deep you are in the ground on where open-source LLM technology is going, do you think that there might be big breakthroughs in the coming months or say the next year that would significantly impact the way that you are able to help out your clients with LLMs yet? As I said, tricky question, but I don't know if you have any thoughts.

Yannic Kilcher: 00:47:08 I think the breakthroughs are already there. So there's always this lag until things really become products. And I think we're seeing the first things now with consumer products. So people taking GPT-4, API, and building cool consumer products from it, but then... So there's already a lag because GPT-4 came out a while ago and until these things really penetrate business-to-business software and get deployed there... So I think the breakthroughs that allow, whether you consume an API or open-source models or fine-tuned models or whatnot, I think the breakthroughs are already there and there are so many products to be built just on the tech we have right now. I don't think it's super necessary that there are more breakthroughs just because there's so much to do. That being said, I have no clue whether there's a giant breakthrough. I mean, certainly one can predict there's going to be probably one or two iterations of open-source, large language models that are significantly better than the last generation before. That doesn't change the fundamental dynamics, but it just expands the things you can do with them reliably.

Jon Krohn: 00:48:37 I think you already touched on something that, something to me that's kind of obvious is something that's coming you already mentioned in the show is larger context windows. So we have things like Anthropic's Claude as a 100,000 token context window, and I can't remember the exact number for Llama 2. Is it 8,000 tokens?

Yannic Kilcher: 00:48:58 Probably.

Jon Krohn: 00:48:59 I think it's that, yeah. So that's a pretty big gap, but it's also... It's unclear how well... I mean, I haven't tested in the sense of that 100,000 token context window. I haven't tested that in any way. It is easy to say that you, it's another thing for it to actually work well.

Yannic Kilcher: 00:49:19 Yeah, I mean it's probably... From what I've seen, I think it does work well when there is a specific part of the 100,000 token context that is now relevant and that you need to use. And I think it's really good at figuring out what that is, but doing integrative information processing across the whole context, I think there, it's just a matter of, okay, the more stuff you put in there, the more noisy it gets. So I'm like you, I didn't thoroughly test that. Yeah, but it's questionable. Longer context is for sure good, but it's questionable whether there isn't something smarter one can do on top of the LLM rather than inside.

Jon Krohn: 00:50:14 Yeah. Yeah. And I think this kind of episodic memory thing that we've talked about could be better where it's like, yeah, note taking and just caching things that seem really important details, which I think is more similar to the way that if we're reading a book or you're studying a textbook, you're not trying to have context over everything. You highlight some of the few things that are the really key aspects of what you're reading in the textbook, or you write them separately in a notebook. And this is an effective way for most human beings to be able

to study for something. It isn't an effective approach to try to be like, I'm going to memorize everything in the textbook. You have to be like a savant.

00:50:59 So yeah, so speaking of volumes, getting very large things like big context windows and research is moving really quickly. Machine learning research has been growing really rapidly. So since 2015, it's been growing at attack of about 20% per year in terms of volume of papers published in machine learning research. And so from the time that you uploaded your first YouTube video in 2017 to today, roughly the volume of ML papers being published has tripled. So you run, as far as I can tell, the biggest YouTube channel on ML research. How do you keep up personally with that huge volume?

Yannic Kilcher: 00:51:45 Yeah, it's gotten harder. So there was a time at the beginning of my PhD, 2016 or so, where I, not even for YouTube, before I started YouTube... I at least looked at every single paper on Arxiv. So in the morning I had a script that downloaded all the ML, so [inaudible 00:52:09] ML and all the CS or the ones I was interested in, but there were a lot. Like the lists on Arxiv, the new publications, and then I just flicked through them because I had an hour train ride. So I just flicked through them for 45 minutes and occasionally read one that I found interesting. So that was possible at the time. I don't think that's even possible anymore to do that, or you really have to be dedicated. And luckily, there are people who are dedicated.

00:52:39 So I think the way most people keep up nowadays is to do a bit of your own scouring on new Arxiv releases, plus to have a network of social media/blogs/lists/automated things that just deliver a stream where you can guesstimate. But there's absolutely research nowadays that most people miss just because it didn't manage to grab the attention of enough people. That would be very

valuable, but there's just, no one cares about because no one else cared about it. And I also, I have the additional luck that people also post on our Discord, people post interesting papers and talk about them. And there are almost daily paper discussions going on and things like this. So I have, by now, a really good support network, I would say, helping me in all of this, making sense of the space. Yeah, that's super helpful.

Jon Krohn: 00:53:48 Nice, yeah, that makes a lot of sense. Do you have any particular resources, publicly available resources, that you recommend? Like these kinds of blogs or communities that any of our listeners could subscribe to or be a part of?

Yannic Kilcher: 00:54:00 Yeah, I would say the best thing to do is find your own personalized mix. Because also what, if we all start doing and following the same thing it becomes, we just increase our blind spots. So I think if everyone does their own personalized mix of sources, the likelihood that all the research somewhere is covered and can be amplified is better. Now, obviously, everyone should subscribe to the two of us, that's out of question. But other than that, do your own personalized mix.

Jon Krohn: 00:54:37 Nice. Yeah, great answer. Thank you for the plug. We didn't pay him to say that.

Yannic Kilcher: 00:54:46 Yet.

Jon Krohn: 00:54:47 Oh yeah. Hey, actually that's not a terrible idea. We actually, we recently did, we started doing, there's a podcast that I love and I've co-hosted twice now, at least at the time of recording. By the time this episode is published, it's potentially I'll have co-hosted more of the show. It's a show called Last Week in AI, and it's a podcast. It's an audio-only podcast where they wrap up the week's AI news. And so this is a different kind of

podcast of this show where I have a guest and we go deep into topics, like with you with OpenAssistant. And so they're not a competitor to ours. But I absolutely love the show, it's the only podcast that I listen to. I make sure I never miss a second of it because it allows me to, kind of like you're saying, having a survey of systems.

00:55:37 The episodes are often two hours long and it isn't a deep technical dive. It isn't intended for data scientists or machine learning engineers, like your content is or my content is. You could be a manager and follow along with all the AI stories over the last week. But it's been great for me because at least then I've, kind of like you skimming through all the Arxiv papers in 2016 on your commute, it allows me every week to at least have heard every one of the big stories. Because there's this kind of weird, for many years, there's this weird disconnect where, yes, I understand, I can write out for you how gradient descent works in calculus formulas, or I can code up a neural network in PyTorch. And so this makes me a data scientist.

00:56:29 But there was this weird expectation on me, or I used to think it was a weird expectation on me, that somebody who isn't technical but who follows AI blogs or something, they'd say like, oh, did you see this paper? And I had this kind of default reaction of, well, I can't keep up on everything. But now with the Last Week in AI podcast, I can. But anyway, all of that is to say, that we actually, we started recently actually paying them to just mention the Super Data Science podcast on air. And yeah, maybe the Yannic Kilcher YouTube channel is a great venue for us to be paying you to mention us.

Yannic Kilcher: 00:57:13 I have no, my main blocker for sponsorships is just my capacity to make the sponsor slots or the shoutouts, because I want to do a good job because they pay me, but

then I just don't have the time. So it's a peculiar situation.

Jon Krohn: 00:57:37 Yeah, there's a channel that I love, a YouTube channel, it's like History Oversimplified or something like this. It's definitely Oversimplified is in the title.

Yannic Kilcher: 00:57:46 Yeah.

Jon Krohn: 00:57:47 And they mostly cover history. You know that channel?

Yannic Kilcher: 00:57:49 Yeah, yeah, yeah.

Jon Krohn: 00:57:50 I love it, it's so funny. But I don't actually know who the guy is that makes that channel. But it's always the same guy, it's always the same voice, and I'm pretty sure he does the animations too because they're terrible. And that's kind of the point. But it's a really fun, if you kind of want to have an overview of like, particularly historical wars, and this stretches back many millennia. Very cool stuff in there. But the point of all this is to say that he does a great job and a very funny job of integrating his sponsor message into that specific YouTube video. So he'll be like, maybe we could have stopped Hitler if we had Nord VPN. That wasn't a great example because they do, it's better than even that. But just to give you a sense of how he goes seamlessly from whatever that specific video is about into his sponsor's message. So yeah, I hear what you're saying there on trying to make it a good experience for everyone, your sponsor as well as your listener. Nice. So yeah, have you ever considered fine-tuning an LLM to be able to curate everything that's on Arxiv or to help you with your YouTube [inaudible 00:59:09]?

Yannic Kilcher: 00:59:08 It could be an idea. Yeah, for sure, for sure, that would be fun. I don't know. It would be interesting to see what comes out. I have not.

Jon Krohn:	00:59:18	Yeah, it would be, yeah, there'd be some engineering challenges. If you cracked it though, I bet a lot of people would be interested in it, right? So if you somehow had, on a daily basis or a weekly basis, you were taking all Arxiv papers and, actually, I guess doing something similar to DeepJudge. Where you are having all of these documents stored in model weights or having embeddings created for all of the Arxiv papers. And then that would allow people to ask questions. But that's actually different, that is completely different.
Yannic Kilcher:	00:59:49	Wasn't like Arxiv Sanity, just an attempt at, I think that was the main attempt at that. So that could be pimped a little bit with newer language models, yeah.
Jon Krohn:	01:00:01	Yeah. That's Andre Karpathi's Arxiv-sanity-preserver, right?
Yannic Kilcher:	01:00:04	Yeah.
Jon Krohn:	01:00:06	Yeah, yeah. And it's interesting that he would've made that around 2016. So I wonder, I don't know to what extent he's updated it since to try to be able to handle the volume we have today. I don't know. But yeah, you could, as you say, pimp it out with a large language model potentially. Nice. So then when you're deciding on what to cover in a YouTube video, how do you, so it's like this, so you're basically crowdsourcing. You have your set of blogs, say, newsletters, that you follow.
Yannic Kilcher:	01:00:41	It's just whatever, whatever feels interesting to me. I don't assume any authority on saying what is and isn't important and so on, other than it's interesting to me.
Jon Krohn:	01:00:52	Nice. And then this way that you've ended up in this situation where you have over 200,000 subscribers on YouTube, was there any kind of intention or goal when you got started with this in 2017? Or was it just kind of

something that you enjoyed doing and whatever happens, happens?

Yannic Kilcher: 01:01:12 Yeah, I mean, I had no expectations or anything like this. I had no knowledge of, that anyone would want to listen to or anyone that doesn't need to, would want to listen to 45-minute ramblings about papers or anything like this. So it's been very special, but I have no aspirations of growth or anything like, ooh, it'd be cool to reach whatever, a million subscribers or so, just because it's a big number. But I never actively do anything like this. I don't use, like I do some YouTuber stuff, I try to get a thumbnail that sort of communicates well and is inviting.

Jon Krohn: 01:02:04 Yeah, you wouldn't try to deliberately sabotage yourself with the worst thumbnails and titles?

Yannic Kilcher: 01:02:08 Yeah, yeah. But I don't know. I don't really go into my analytics and be like, oh, this X many subscribers and so on. I try to improve, but not for the sake of pushing the numbers.

Jon Krohn: 01:02:24 So you're not deliberately chasing the algorithm or hopping onto trends for the purpose of building clickbaity videos that lots of people click on? It's more about having great quality content in the video and catering to the person who's going to listen to a lot of that video and just enjoys getting deep into the content. And so however they happen to chance upon you in the first instance, they're going to stay because, yeah, they love this kind of deep analysis of the papers.

Yannic Kilcher: 01:02:55 I hope so. I don't have time to do big retention tactics or anything. I just don't have the capacity for that, so yeah.

Jon Krohn: 01:03:04 Nice. Yeah. Well, I think we're aligned, just kind of similar to the way that our startups are aligned, I think our philosophy on content creation is the same. This is

definitely the same. We try to have a YouTube thumbnail that's relevant, and I try to have a clear title that is basically just the subject. I'm like, if this was a chapter of a textbook, what would I name the chapter title? And, yeah, hopefully, yeah, maybe we can get to 200,000 subscribers on YouTube someday too. We'll see what happens. But, yeah, I'm not, my personal value isn't tied to my number of subscribers. I hope people are, just trying to create the best possible content every time and hope that whoever's listening to it out there is loving it. So we'll see how that continues to go. So a common topic on your channel that comes up is adversarial examples. So I think there's over a hundred videos on how [inaudible 01:04:10].

Yannic Kilcher: 01:04:09 Oh really? Okay.

Jon Krohn: 01:04:11 That's what our researcher Serg Masís said, that figure.

Yannic Kilcher: 01:04:16 Okay.

Jon Krohn: 01:04:17 So you've also written papers on adversarial examples and adversarial training, including at some of the biggest venues like NeurIPS. And so for our listeners who are unfamiliar with this topic, what are adversarial examples, what is adversarial training?

Yannic Kilcher: 01:04:36 Yeah, adversarial examples are, well, let's see. It's still a bit of an unsolved question what exactly they are. So the phenomenon is that I can take a neural network that's been trained on something, for example, classifying images, and I can make a tiny perturbation to an image that is invisible, provably invisible. Or not provably invisible, but lower than eight bit precision, for example. So a pixel can't change more than a single color value, which is not perceptible to humans. So I make these tiny imperceptible changes to the pixels of the image, and the

human would still see exactly the same image, but the neural network thinks it's something completely different.

01:05:36 And what that does is, obviously these perturbations aren't random. These perturbations are very targeted, such that it kind of abuses the dynamics of the neural network to just push its decision as far away as possible from what it originally predicted. It's kind of a fluke, I would say. And it's certainly an out-of-distribution example. By doing these exact perturbations, you are going into a direction in the input space that was never covered with the training data. And because you're going in exactly the right direction, in exactly the direction of maximal discrepancy, you don't need to go very far to make a big change. That's actually how you, by construction, that's how you build these examples. So you make this super targeted change, and what you're targeting is the decision of the model, and that means you end up with the change that is the least amount of change for the biggest amount of change in decision. And that's how you craft these things.

01:06:52 And people have actually done very cool stuff. They've done these in real life, so they have taken street sign classifiers and they just slapped, at the correct place they slapped like a sticker on the stop sign so that then the neural network is like, that's not a stop sign, that's a street lamp. And even though it's clearly a stop sign, right? But just because you, and the idea here is you are able to look inside the neural network at all the weights, at the gradients and so on, and that's how you can craft these attacks. It is a bit, so it happens to humans, for example, optical illusions are very much adversarial examples.

01:07:34 They are, they don't... Optical illusions, coincidentally, they wouldn't look really special or different to a machine. But to a human because it abuses the particularities of

your visual system it just looks like something odd. Or, the fact that people see faces everywhere, that's an adversarial, that you can think of as an adversarial example. And yeah, that's that. And there's a lot of cool work around it, including the whole series of work on GANs, on generative adversarial networks. It's essentially just an extension of this idea of adversarial examples. So there's a lot of stuff to be done, and we've written some papers on it about how, one explanation of why these things happen and how you can potentially defend against them, approaches and so on, so yeah.

Jon Krohn: 01:08:30 That's cool. I had never thought of this analogy to human vision with the optical illusions, but this makes perfect sense. So in the same way that our particular system, our rods and cones and the way that they combine together in our neural cortex, there's this particular way that evolved, is going to be helpful if we're a monkey, and we are. If we're a monkey in the trees looking for fruit amongst leaves, this leads to particular kinds of, or a system works in a particular way. That then when somebody on a piece of paper draws lines in a specific way that is unlike anything you would ever encounter in the wild, you can end up messing with the way that our visual systems work. So this is a really great analogy.

01:09:18 Where it's like, you're describing with the deep neural network if we look on an individual neural basis, so artificial neurons, we can figure out, okay, putting this sticker on the stop sign is going to make it think that it's a streetlight instead of a stop sign. Similarly, it's our understanding of the way that human perception works that allows these optical illusions to work. So that's a great analogy, and I've never come across that before. It makes a lot of sense. On your channel, you reviewed a paper by Professor Alexandre Madry's group at MIT. This paper was called Adversarial Examples Are Not Bugs, They Are Features. And this was really interesting to us

as we were doing research for your episode. How is it that this could be a feature and not a bug? It seems like such a problem.

Yannic Kilcher: 01:10:10 I mean, this is one of the, certainly one of the landmark papers in the field that goes a long way into explaining. I mean, it's a, you can look at these things as phenomenon from different angles. And one of the angles is very well explained by this paper. Which is essentially where do the adversarial examples even come from? What causes them? And what this group has found is that these things are, they are given by the data. So adversarial examples or adversarial attacks are made possible by essentially using real patterns in the data set that are just too imperceptible for humans to see. So, for example, if you're classifying animals, right? You have the shape of the animal and so on, where it is in the landscape, and things like, big features like [inaudible 01:11:24], and features we humans usually go by. And then you have other features like the structure of their skin, of their fur and so on, and these are predictive too.

01:11:33 So now any machine learning model, it can essentially, it's not bound by human limitations. So as far as it can pick up on these tiny features, it may use them or it may use the big features, let's say the shape features. And just by the nature of, a good model is obviously going to use both. But the difference is any model, whether good or, it will include these more imperceptible features. And all the paper says is that when we craft adversarial attacks, we essentially operate in that space of imperceptible features. Which is, I'm going to take a picture of an airplane, I'm going to slap the fur of a fox on it, but not the fur like boom, but just the very high frequency features of the fur that the neural network would pick up on. So to a human it, because these are so high frequency, it doesn't look like too much of a difference, yet to the neural networks

like, wait a minute, I know that structure of pixel differences. That's a fox, right?

01:12:48 And the paper essentially says, this is not a fluke. This is not a bug in the sense of, oh, it's something that is a result of how we train things and so on. No, these are actually predictive, true predictive features that are absolutely valid for the neural network to learn from the data. It's just that they're not the features we would like it to learn, but they're valid. I think that was the core point of the paper, and they had a series of quite smart experiments to demonstrate that that is actually what's going on.

Jon Krohn: 01:13:25 Nice. That's very cool. I hadn't thought of any of those kinds of ideas before, and crystal clear, amazing explanation of this. The high frequency fox features on top of an airplane, it's so easy to understand that way. And it's also so easy to see how these are features. So despite all of that, despite them being features, are there ways that we can mitigate adversarial examples without affecting model accuracy?

Yannic Kilcher: 01:13:49 Yeah, well, no. The short answer is no. Because we have to define what we mean by model accuracy, obviously. Obviously, because adversarial features exist, these models don't generalize in a sense. Because if they, by generalization, one can define generalization as working everywhere a human visual system would work. If I think of a vision classifier and I ask myself, what does it truly mean for the thing to generalize, my response would be, well, at any particular thing I look at, I want it to give the same response as if I were to give that response. So clearly by the definition of that, they don't generalize, right?

01:14:36 So if we say accuracy is in terms of generalization capability, then the way to go about it is to just align

them more with the human visual system. So as long as these things have access to features, to perceptions, that the humans don't have access to or are biased to downgrade or so on, they're going to learn them. And they're going to learn, then you can, if there is a misalignment, you can always abuse that misalignment to craft these adversarial examples. So the more you align, even the perception, like the input domain of these things, the less adversarial examples you're going to have. On top of that, there are some techniques that you can do to sort of mitigate the immediate phenomena, and it's a bit like cat and mouse. So people come up with a new way of defending against adversarial examples in classifiers, and then other people will come up with a way of attacking those, and so on.

Jon Krohn: 01:15:42 Yeah, yeah, yeah. All of this makes perfect sense. So this isn't exactly an adversarial question now. I mean, we were just talking about adversarial examples. But something that seems kind of related in the sense that we're talking about flaws or abuses possible with these modern AI systems, particularly with deep neural networks, large language models, something that has been getting more and more prominent in recent years. This is an obvious thing for probably all of our listeners, is that Deep Fakes are getting better and better quality. And so if people are watching a YouTube version of this episode, they'll notice that you've been wearing sunglasses this whole time and that actually you're always wearing sunglasses on YouTube. So this is related to people not being able to deep fake you, right?

Yannic Kilcher: 01:16:43 It was originally, but now one picture is enough to deepfake someone. So it's become more of a branding thing, honestly. There's really not too much more of a reason for it other than people kind of know the glasses by now, so yeah.

Jon Krohn: 01:17:06 Well, yeah, it works for sure. I was blown away when Natalie on our team who does, she helps me find great photos for social media of people. I was kind of blown away when she found one of you without glasses. I was like, oh yeah, there we go. Yeah, so that's what we actually used. We're going to get to audience questions in a bit, but I used that photo to post that you'd be coming up on the show, and we had quite a few audience questions for you. But before we get to those audience questions, I have a few questions for you about where you see this space going, where you see the future of AI going. So obviously we've talked about some of this kind of stuff in the episode already. We've talked about context windows, we've talked about episodic memory, but yeah, what else? You're on top of the latest machine learning research, other than the topics we've already talked about, what do you see? And it doesn't necessarily, because also those advances that we talked about were LLM specific, so maybe beyond LLMs or with AI in general, what do you think is likely or promising over the next few years or few decades? What does the future look like to you in this increasingly AI-driven world?

Yannic Kilcher: 01:18:24 Yeah. Who knows, right? Who knows? I have no idea. Absolutely. Absolutely no idea. Yeah, I mean, the question is where do you get, if you gain more and more the ability to statistically model really complicated distributions, for example, like language? Where do you go or where can you get to? Some people say you can get all the way to AGI with that by just being able... Obviously if you had a perfect predictor for what's the next token in every given situation, you'd be pretty close to human intelligence, even beyond maybe for some tasks.

01:19:11 I have no idea and even if we say you will be beyond LLMs or so. I think the hardware domain needs to catch up, especially the domains of robotics and so on and anywhere where the software components interact with

the real world. LLMs can now write novels with actually intricate plot points and so on, but if you try to make a gripper grip something, it's still a super huge challenge and it's going err and then aim and then very slowly. I have no idea whether it's possible or going to happen, but I am hopeful that in the future, that domain will significantly improve. I definitely see a lot of contributions, a lot of breakthroughs to be made there.

Jon Krohn: 01:20:11 Yeah. It's an interesting thing, isn't it, that we, I think a decade ago, the assumption was that it was blue-collar tasks that were more automateable, and it was because we made some strides. You can have the robot arm in the factory, like you're describing the gripper, doing tasks. Whereas 10 years ago, it was inconceivable that you could have a machine do a good job of helping you write an essay or just write the essay or write the novel. I think it surprised a lot of people in the last year. The scaling has been extremely effective at automating white-collar work. Yeah, this was an interesting shift, but it does mean that there is still now lots more potential in automating blue collar work and it isn't as obvious how we can be making the strides because with these white-collar tasks, like creating a social media post or drafting an email, those are...

01:21:17 We talked about how scaling early in the episode has been working well so far, and it probably will, just like your example on the CPU is going from 13 nanometers to three nanometers. We still have a ways to go with scaling for several more years that is going to create, I anticipate, with GPT-5, GPT-6 kind of things, they're still going to be making big strides and we're going to be very surprised at the emergent capabilities that come out of that. Whereas with robotics, it's physical, so it's a lot harder to scale. So yeah, there's just this real world constraint. It's way more expensive when you're dealing with some robot that has

to be moving cars around and all of the energy and gas it takes.

Yannic Kilcher: 01:22:03 Yeah, for sure. For sure. That's why we put more work into software because it's just easier and the fruits of the labor, let's say, come faster and cheaper. But yeah, I have no idea. But yeah, it's interesting, the first jobs that go away are Instagram models. Who thought the first jobs to be replaced by AI is girls taking pictures and people sending them money for it?

Jon Krohn: 01:22:32 Yeah, yeah.

Yannic Kilcher: 01:22:32 Yeah, it's a crazy, crazy world.

Jon Krohn: 01:22:37 Yeah, it's wild. So as you say, who knows, is probably the best answer to my question about where you're going. One last thing before I get into the audience questions, which is a question that I used to actually ask on the show all the time, but I haven't been as much lately. And I think that's partly because Serg Masís, our researcher, does such amazing deep research that I have all these incredible... I mean, the topic areas and questions that he had prepared for, it was like many times, every week, there's many times more questions and suggested topics than we have time to cover. And then, of course, we end up getting organically into more topics as well just through conversation and through me having ideas on the fly.

01:23:24 So a question that I used to ask frequently of guests, but I don't as much anymore, but I think would be interesting in your case, is I'm sure there are people listening who when you were talking about DeepJudge, your company, and the work that you're making applying LLMs to the legal space, I'm sure there were people out there that were thinking, "Wow, that sounds really cool, and it would be awesome to be working with Yannic Kilcher and the

amazing people that they have over there." So are you doing hiring at DeepJudge? And whether you are or not, what are the things that you look for in the kinds of roles that our listeners would be interested in?

Yannic Kilcher: 01:24:00 We are always hiring, of course, and the roles that we look for are quite diverse, but mainly we're looking for generally skilled engineers that have more or less experience with machine learning. But for us, people have to be multi-talented, let's say, because in a startup you have to do many things, and it's not always a new model that has to be trained. That happens often, but sometimes it's something else, sometimes it's plumbing, sometimes it's this and that. So I think what we're mainly looking for are generalists who have a broad knowledge of technology, obviously geared towards machine learning, but also general technology, and who are happy with a diverse set of tasks rather than being super duper specialized on a particular thing.

Jon Krohn: 01:25:03 Nice. Makes perfect sense for an early stage startup. And yeah, like a lot of the things we talked about in this episode, I couldn't agree more with what I'm looking for. Nice. All right, so let's dig into some audience questions here. All right, so our first question here comes from Mike Nash, and Mike is in the UK, I believe. Yes, he's in England. He has a question for you. Actually, I picked this as the first one because we just were talking about DeepJudge and what you look for in people you hire. Mike was interested what the most complex challenges you've had with your startup or key lessons you've learned in getting DeepJudge off the ground, getting an AI startup off the ground.

Yannic Kilcher: 01:25:44 I don't think it's any or too different than other startups, just referring to AI startups. Probably there's a bit more education involved with potential users and so on of what really AI is and what in your particular product it can do

and it cannot do because especially in the area of ChatGPT, they'll just expect sort of a magic thing that can do everything and anything, which is... I mean, you can get in some part there with the newer tech, but still, I think the only difference to a non-AI startup, whatever that is today, is that there's probably a bit more education involved on your end towards customers or users. There's not a biggest challenge. A startup is just a string of problems. So everything's always burning. Yeah, that's the life. It'd be nice if there was just the one problem or the one big problem that you have to overcome, but it's more like that's every day. Yeah, you have to be mainly a problem solver if you're interested in startups, especially in founding.

Jon Krohn: 01:27:08 Yep, great answer. Another one here comes from Dr. Mark Moyou. He's a senior data scientist at Nvidia, and he had a few questions, but I think one of the ones that I liked the most was, how do you balance leveraging frameworks out of the box versus implementing from scratch to solidify learning? So I think the idea here, it's related to, I guess, just learning ML techniques, but maybe particularly in the context of your YouTube videos, there's these trade-offs between implementing things from scratch and really understanding it well versus just using the framework and having it work magically.

Yannic Kilcher: 01:27:51 Yeah, I guess it depends on your goals. I think I had to trade off much more during the PhD. Because now it's kind of clear in the startup, just whatever is best for the use case. So it's take a framework, take the library, and when you reach the point where those things break or aren't enough anymore or don't scale well enough or anything like this, then that's the point where you do your own. I would say on the YouTube side, it's the opposite. It's like obviously I want to produce educational content, so it would be kind of dumb for me to just be like, "How to..." I've not done too many coding videos, but

it would be weird to be like, "Let's code transformers," and I just do hugging face dot inference and be like, "Okay." In academia, it was probably the most balanced where obviously you need to get to a paper quickly, which means that you need to somehow produce results that leads you into the direction of really just using other people's code without much consideration. But also obviously you deeply need to understand something. I used to implement quite a bit of stuff myself in order to just learn all the things that are in there. I mean, it's hard to say. It's mostly what do you want to achieve? If you want to achieve learning, then sure, implement yourself, but if you just want results, then I'm not in your way to just take what works and run it. That's fine.

Jon Krohn: 01:29:32 Yeah, that makes sense. We've got one here, Indu Sambandam, hopefully I'm not butchering that last name too much. She's based in Toronto, and so she's interested in how you can make your models robust to unexpected consequences. Yeah, I don't know if this is something that you deal with. I don't know if she's asking this question specifically based on content you published before or whether this is just kind of a general question, but when you are, say, building machine learning models for DeepJudge for production, how do you anticipate and mitigate against long tail events?

Yannic Kilcher: 01:30:15 Yeah, you really can't. I mean, you can build some, let's say, support systems to detect when it happens. And really, mostly against these systems are usually called guardrails or things like this. You can also fiddle with the fine-tuning a little bit. But if you want to fine-tune these things away, you really need to be able to explicitly enumerate them. You need to be able to say, "Well, sometimes it does the thing where it does that," right? You really need to be able to explicitly name the thing and then you can fine-tune it away. And then you go one by one enumerable. A combination of that will get you a long

way. And then obviously expectation management from the end users where you say, "Hey, look, here is how it works under the hood, here is how it's trained, here is what it does, and that means it's a statistical model, and here are the consequences of it." And when you use any of these products, ChatGPT, Copilot, whatnot, you're perfectly fine with that, right? You just click regenerate once it happens. So I think that's okay.

Jon Krohn: 01:31:37 Great answer. The last one here from the audience, so this gentleman, Sam Dixon, he's a product test engineer in Austin, Texas. And yeah, this is kind of just a personal one in the sense of, it's not really a technical question, but with all the things that you do with your wildly successful YouTube channel, being an entrepreneur, how do you do more with less? How are you able to achieve all of this and not go crazy?

Yannic Kilcher: 01:32:09 That's a good question. I have no idea. I mean, it's getting harder, I have to say. Once you have the responsibility for other people's literally jobs, obviously we employ a lot of young people and they're super skilled and we're in Switzerland, so no one's going to go hungry if we fail. But still, it's like, it's their jobs. So as you load more responsibility, it becomes harder to keep other things going, but it's just you find some time here and there and you try to be effective. And I am certainly not the best at that. I'm certainly very procrastinate-y. But yeah, I don't know. I have no good answer. I'm really not good at time management. Really bad. I'm a really bad procrastinator. I could probably do five times as much if I had all of that under control. But in the end, you know when there's a deadline tomorrow and you really have to hand in that project report? It somehow gets done. Now, it would be nice to get these things done without lack of sleep, but it always somehow gets done on time what needs to get done. So that kind of tells you that time is kind of bendable.

Jon Krohn: 01:33:43 Yeah. Well, it is over here.

Yannic Kilcher: 01:33:44 So I have no better answer.

Jon Krohn: 01:33:44 Yeah, yeah, yeah. No, I guess mean if I was to try to extract a morsel of self-help tip from that, it would be that setting deadlines helps. If there isn't a real deadline, then it's very easy to just let that procrastination lead to lack of productivity.

Yannic Kilcher: 01:34:08 And I miss so many things. It's not like I get all the things done. I miss so many things, and I'm quite disorganized. Often if I'm overwhelmed, I stop responding to email. There are some sponsors that I literally haven't had the capacity to send an invoice for more than half a year now. It's like, what is this problem to have? There are people who want to give me money, but they need [inaudible 01:34:41]. I'm like, "Ah." So don't take time management tips from me by any means.

Jon Krohn: 01:34:48 Yeah, I understand. I think the invoice thing, for me, that happens as well, and it's because, well, I don't want to disappoint people. And I know with that one, with invoicing them and them paying me, typically people, it's not something that's going to disappoint them in a way that having a software release out on time matters in a startup. Yeah, I totally get that. Nice. So yeah, so that covers the topics that I wanted to cover in this episode that I thought were the most interesting ones to get through with you, as well as the audience questions. So before I let my guests go, Yannic, I always ask for a book recommendation.

Yannic Kilcher: 01:35:33 Yes. Well, it's a tricky, tricky question. Obviously The Little Book of Deep Learning is really good by François Fleure. I have it here somewhere, but I've looked for it before. It's in the other room. No, apart from technical books, I just enjoy mostly nonfiction, so I like two. Well, I

		just said not technical books, but I guess I just read other technical books. None in particular though. I really liked Zero to One for some reason.
Jon Krohn:	01:36:23	Me too.
Yannic Kilcher:	01:36:24	And I can't really even say what it is about it. I'm usually not super much into AI startup books or so on, but for some reason, that book, I listen to it usually. I just listen to it over and over. I don't know, maybe it's just the reader who has a soothing voice. I don't even know who speaks it honestly. But yeah, just good.
Jon Krohn:	01:36:51	Yeah, yeah, yeah, I like that book as well. Awesome. All right. So if people want to be following you after this episode, if they weren't already, obviously subscribing to your YouTube channel is a great way to keep up with your content. Are there any other ways that you recommend?
Yannic Kilcher:	01:37:08	That's the main one. Yeah, we have a Discord community around the YouTube channel, mainly for interested people in machine learning, discussing research and so on. That's also a big thing.
Jon Krohn:	01:37:25	Nice. All right. Well, we'll be sure to include links to those in the show notes, which yeah, people can get from the superdatascience.com website, specifically superdatascience.com/733. So yeah, look out for those links and everything else, of course, discussed on the show will be there as well. Yannic, it's been a pleasure, an honor to meet you. Thank you so much for coming on the show. Yeah, hopefully we'll have the chance again maybe some years in the future to see how your journey is coming along. Thank you so much.
Yannic Kilcher:	01:38:01	Thank you very much for having me.

Jon Krohn: 01:38:08 What an experience to meet a legend like Yannic. I hope you enjoyed our conversation. In today's episode, Yannic filled us in on how the data collected for his OpenAssistant project has proved to have the most long-term utility to the open-source community, how blending semantic and keyword-based approaches together has proved critical to building a tool that works well at surging over swaths of legal documents, how when he got started on his PhD in 2016, it was possible to flick through all of the ML research papers that were published. But now he uses a collection of aggregators like newsletters and social media feeds to stay abreast of the biggest developments. He talked about how adversarial examples where a neural network thinks an image is wildly different despite appearing the same to a human can actually be features, not bugs. He talked about how there's tons of room for growth in hardware capabilities, particularly robotics, in the coming years.

01:38:56 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 Yannic's social media profiles, as well as my own at superdatascience.com/733. 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 terrific episode for us today. For enabling that super team to create this free podcast for you, we're deeply grateful to our sponsors. You can support this show by checking out our sponsors' links, which are in the show notes. And if you yourself are interested in sponsoring an episode, you can get the details on how by making your way to jonkrohn.com/podcast.

01:39:41 Otherwise, please share, review, subscribe, and all that good stuff. It's your word to your friends, your colleagues

that helps us grow this show, so we really appreciate it. Thank you for any efforts you put in on our behalf, ensuring that we can continue to make amazing episodes for you for years and years to come. But more importantly than anything, we, of course, hope you'll just keep on tuning in. 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.