**AIDAN** 00:00

It was quite extraordinary, quite extraordinarily convenient. That simply by scraping more data off the web, not necessarily clean data, like messy data it’s just web data, you're just taking in everything and there's tons of junk out there. But taking in a very noisy, messy, massive data set, and just making the model bigger, throwing some more chips at it. And what came out the other side was something that understood language, in a way I personally thought we were we were decades from.

**CRAIG** 00:55

We're talking this week to Aidan Gomez, who helped develop the transformer algorithm, which lies at the heart of generative AI and powers large language models, such as GPT-4. Aidan now leads a startup Cohere, a platform that offers users access to pre-built LLMs, as well as allowing users to create their own LLM.

But first, I want to give a shout out to our sponsor, and encourage anyone with a business to take advantage of a deal from Oracle, which is offering a full NetSuite implementation with no down payment and no interest for six months. NetSuite is a cloud-based business management software for enterprise resource planning, financial management, customer relationship management, and E commerce. To take advantage of the offer, go to netsuite.com/eyeonai.

Now let's get back to Aidan,

**AIDAN** 02:20

I’m Aidan. I am the CEO and co-founder of Cohere. I started the company with Nick [Frosst] and Ivan [Zhang] about three and a half, four years ago. Before that, I was kind of the perpetual intern at Google Brain during my undergrad and then later, my PhD. I started down in the Bay Area and Mountain View.

**AIDAN**

I was part of the team that created the transformer. And it was incredibly exciting. You know, it took the world by storm, I think, certainly, to my surprise, and I think everyone on the team was quite taken aback by its popularity. But before Google, I was an undergrad or also during Google, I was an undergrad at U of T [University of Toronto]. I grew up in rural Ontario, Canada in a maple forest. And so I'm the world's most Canadian man. Yeah, that's me.

**CRAIG** 03:25

And so you were at U of T studying with Geoff Hinton, I guess, he was probably kind of retired from teaching by then.

**AIDAN** 03:38

He was definitely not teaching, but he was still at the university. This is before the Vector Institute was created. And so yeah, he was, you know, like I, I didn't really get into deep learning until after second year. And then when I started looking into it, I became obsessed, and I was just reading papers, night and day, I would fall asleep with a research paper sitting on my bedside, I would, in between sets at the gym, you know, have a stack of papers that I was reading through. And I kept seeing this name. And his affiliation was U of T, which was where I was. And so I reached out to Geoff, this is before Google.

And I just, you know, I'd been reading his papers at that point when I was studying, you know, reused and MLPs and just the most simple piece of the AI deep learning stack. And I was like, you know, why do you have these functions that are just flat and then up? I think that they should be periodic. And so I emailed him with an idea being like, Hey, why did you make this decision? I think they should be periodic. There should be some regularity and it should be bounded so that you know, it doesn't go to infinity if we get a large input. And to my surprise, he responded, and he actually explained the decision. And so that was pretty amazing. That was my first interaction with Geoff. And then when I came back from Google, in Mountain View to Toronto, Geoff said, Hey, come work with me in, in the Toronto Brain office. And that was where I met my co founder, Nick.

**CRAIG** 05:31

just on, on so so you, you worked on the transformer algorithm with a team in Mountain View at Google. Google Brain? Was it? Yeah, Google Brain. Yeah. So can you explain that periodic versus stable, or what? Which algorithm were you talking about?

**AIDAN** 05:54

Yeah, I mean, it's not very important, because I was wrong. It doesn't, it doesn't really matter. And I think it's more just to Geoff's credit, the fact that he responded to a second year undergrad with, you know, a wacky idea, earnestly. And this guy was literally the top of his field yet toook time for me. And so I think that that particular piece that, I mean, he made me - it's interesting. So for instance, in deep learning and neural networks, we have these neurons, these neurons fire, there's some function that determines their firing, there's some generally some threshold at which they don't fire, they stay dormant. And then above that, they fire. And so when they're firing, they basically, they fire linearly proportional to the input intensity that they're getting. So if the input intensity is high, the output intensity is high when they're firing, but that leads to potentially unstable behaviors, if you have, for whatever reason, some sort of blow up or some sort of like burst of signal coming in, then you'll get a huge burst out. And it'll that'll propagate and make things more and more noisy. And that leads to instability. It makes things complicated in training. And so my proposal was, instead of just firing linearly proportional to your inputs, instead have some sort of predictable, regular periodic pattern, like a sine wave or something. So that you always know your output is bounded between some values. But that has not taken off. And we've since solved the training, instability in the blow ups and that type of thing. So it was just my first email to Geoff, I think six months into my study of deep learning.

**CRAIG** 08:02

Wow. That's impressive. And from a Maple forest.

**AIDAN** 08:11

Yeah, I love that. But I go back,

**CRAIG** 08:13

Then at Google Brain, what was the project that you were working on? What was the initial idea that led the transformers?

**AIDAN** 08:30

So I was on the infrastructure side, like the original idea, I joined Google for I was working with Lukasz Kaiser, and what we wanted to do, I think, Lukasz operates half a decade to a decade ahead of his time, constantly. And so the project that I joined for was actually this paper called one model to learn them all. And the idea was, we're going to take every single dataset that machine learning researchers have compiled, and we're going to put it into one model. And that means it needs to be multimodal, because we have datasets for images, we have datasets for audio, video, you know, text, everything. And so what we wanted to do was throw all the modalities in, as well as out. So you can consume video and let's say, describe the video or you can consume audio and transcribe it. But you can also take in some text and then produce audio, you can also just describe the video that you want and video comes out the other side. So it's just like fully multimodal on both input and output side and we just train on everything like truly everything we've come across.

**AIDAN**

This now sounds kind of familiar, right? Because this is sort of the project roadmap that we're on right now with these large language models that we're throwing everything we have, the entire internet, and now we're starting to add in every modality that we can. So that was what I joined for that was a different project altogether. To support that project, we built this, we built this piece of software, this piece of infrastructure. Because that model was going to be huge. And the data pipelines were going to have to be extraordinarily complex. And so we needed something to suit that. And so, what we did was created this program called tensor tensor. It could distribute across arbitrary numbers of GPUs like 1000s and 1000s and 1000s. And it was very focused on auto regressive modeling, which is the type of modeling that the transformer is.

**AIDAN**

And so at that time, I was sitting next to Noam [Shazeer], who was fiddling with autoregressive models and in particular, attention based models. He was really interested in attention. And then we heard about a team over in translate which was being led by Jakob which was also interested in attention based autoregressive models. And so Lukasz convinced Noam and and Jakob to come over build it on our stack, build it on tensor tensor. And they did. And so over the next I think, 10 weeks, it was just a sprint to build this model. And the intensity just ramped up and ramped up because the results we were getting were extraordinary.

**AIDAN**

So I think this was like, it wasn't the first, but it was one of the very early, extremely successful scaling projects, like hyper scalable architectures, massive data, massive model sizes and massive GPU clusters just lead to extremely high performance.

**CRAIG** 11:55

And, and the first of all the tensor tensor. That's a framework or an orchestration layer.

**AIDAN** 12:04

Yeah, yeah. So it, it's, it was built on top of TensorFlow at the time. But it was basically just a library to support large distributed model training. And it had all the latest kind of tricks and hacks with learning rate schedules and initialization techniques, and it had all this stuff built in. And so it let us experiment really rapidly. I think, if I'm being honest, tensor tensor was a mess. It was crazy. It was just like all over the place that supported everything we were just throwing, every new paper that was coming out into it. It’s a little bit chaotic. And there exists far, far better systems nowadays. But back then it did the job. It did the job, we were able to move insanely fast. And so I'm quite proud of it.

**CRAIG** 13:13

And you were - attention was already something that was being talked about. A couple of questions in that process. What was your role? I mean, I'm a journalist. I imagine you guys sitting next to each other furiously coding, I mean, were you coding? Or is it more that you're in a room with a whiteboard trying to figure out the architecture or is it something else?

**AIDAN** 13:51

There's a lot of like, whiteboarding and diagrams and just conceptual structuring these building blocks and putting them together and the thinking about the architecture itself, there was a lot of that. And that was mainly done by Noam, Ashish, Nicky and Jakob. I think, for me, like I wasn't sleeping. I was working, like 14-hour days coding, building up the infrastructure, making it more robust, running experiments. And so it was very much hands on coding and no one was sleeping. Everyone was just hacking, experimenting, running little tweaks, little ablations to see if I add this in what changes if I if I remove it, if I tweak it? Every single one of us was just messing with everything and trying to figure out what was the optimal configuration. And so that's how we got to that finished product.

**CRAIG** 14:57

Yeah, and and certainly the result now is leading to auto code generation. Were you using any tools to speed up the writing of the code?

**AIDAN** 15:14

At that time? Nothing existed. Truly nothing, nothing existed. It was all. You wrote it yourself. Yeah. Yeah, that came that came later. And that was powered by transformers. Yeah, they kind of.

**CRAIG** 15:35

I've read the paper and, and certainly talked to a lot of people about transformers and, and their progeny. But can you explain as in as simple terms as as you can muster what the transformer algorithm is and what it does? And I'm just curious, too, if if, if you were to send me the transformer algorithm, sort of the basic algorithm is it a million lines of code? Is it 20 lines of code? I'm just curious what it looks like.

**AIDAN** 16:21

Yeah, nowadays, it's probably closer to 20 lines of code. Extremely, extremely simple. I think a big part of the beauty of the model, the architecture was the fact that it was just so simple. Like it, it is among the simplest architectures that were going around at the time, it was just some, like the most basic layer, the layer that has existed for like, I don't know how many years now, maybe over half a century. Like the the basic layer is called, like an MLP. That's just what it's called MLP. And really, the transformer is like, it's a simplification, but it's just some MLPs stacked on top of each other, plus an attention

**CRAIG** 17:20

NLP? You're saying like natural language process? M No. Okay.

**AIDAN** 17:25

Yeah, yeah. So this is just the name doesn't matter. multi-layer perceptron, okay.

**CRAIG** 17:33

Multi- layer perceptron sounds like a neural deep net. But

**AIDAN** 17:38

totally, yeah, that's the fundamental unit. And before, before, transformers, there were these very complicated LSTM architectures with gates and all of these like confusing bits and bobs that just made it made it work. With the transformer, all of that was torn away, and the layer became MLPs plus one attention. That was it. And so that was that was super. I don't know that there was a very, it was beautiful, that you could just carve away so much stuff and just leave something so simple that performed so well, that was so scalable. So the architecture is not this hyper complex beast. It's actually just a very simple scalable compute saturating, you know,

**CRAIG** 18:38

well explain what it does. So you have the multi-layer perceptron as as the base How do you create attention?

**AIDAN** 18:53

How do you create attention? Yeah. So attention is like this idea that you want to relate parts of a sequence, to other parts, fundamental property, that there are relations, if you have a sequence of things a thing in a list in an order, there are going to be relationships between those things. Obviously, that appears on language, very, very strongly, you have adjectives, which are tied to nouns, and, you know, tons and tons of structures like this. And so since we were developing this explicitly for language, we wanted the model to be able to represent those relationships quite easily. That's what attention does. Attention says, For this word, in this sentence, I'm going to learn which other words or which other word in the sequence it's related to. And so for the sentence, the brown dog, you're going to want to learn that brown refers to dog and maybe The refers to dog. So you're gonna want to model those relationships and attention enables you to do just that. And it's not that simple. It's not just like the model is learning adjective noun relationships, it's learning far more complex stuff that we probably don't even have a language to describe. But we just do it intuitively in our heads. So that like that attention layer is the fundamental unit of learning relationships in sequences. And it turns out to be extraordinarily powerful.

**CRAIG** 20:37

And how then does that scale because I've spoken to Ilya [Sutskever] on the podcast, and he talks about seeing the paper, like the next day implementing it in, in what they were doing that that led to the GPT models. How does that scale them into the large language models that we see today.

**AIDAN** 21:12

In their earliest form, it was like a very naive scaling, it was just take it, take the model, and make it bigger. And the way that you do that, as you add more neurons to the network, you add more layers. So it becomes, you know, a much taller model much more deeply stacked. And you just take a much larger dataset than the one that we were considering and a much, much larger model than the one we were considering. And a much larger pool of compute. You plug those all together. And what comes out the other side, I think it shocked virtually everyone. It was quite extraordinary, quite extraordinarily convenient. That simply by scraping more data off the web, not necessarily clean data, or like messy data is just web data, you're just taking in everything. And there's tons of junk out there. But taking in a very noisy, messy, massive data set, and just making the model bigger, throwing some more chips at it. And what came out the other side was something that understood language, in a way I personally thought we were we were decades from. Yeah, it was it was quite a extraordinarily convenient and exciting reality.

**CRAIG** 22:42

So in that led to Bert, is that right?

**AIDAN** 22:49

That that in particular, like Bert predated, or maybe I have them in the wrong order. There's some order there's, there's GPT one, which was the first of these scale up large language model papers. I think Bert predated GPT one, I think. But Bert is a different thing. Bert is kind of like a different beast. Instead of learning to generate language it learns to represent and that's a subtle distinction. Now, like, we're all paying attention to the Generate side, because it's so it's visceral, right? It's like, you can talk to these things they can write back to you. It feels there's a very visceral human reaction to something that can speak to you.

**AIDAN**

But there's another side to this whole thing, which is representing language in a numerical form. And that's extremely important. It's hard to overstate how significant that is. And that was like the first killer application of transformers. It was integrated into Google Search and Google themselves describe it as the most significant advance in search quality in I think it was two decades 20 years like basically Google's entire lifespan. So that was, that was amazing. We got we got something we got a model, we got a program that was capable of representing language to be used downstream for applications like search and classification, etc. Extremely, extremely faithfully, like in a very, very high utility way, in a way that just boosted performance in a way we really didn't expect across pretty much any tasks you throw at it. And anytime you want it to use language for some downstream thing, putting a Bert model there and taking the representations from that and running with those representations, you beat state of the art, you outperformed everyone else. So maybe, maybe Bert was like the first seed of this idea. We can take a transformer, we can set it against a very simple task on a very diverse set of data. And what comes out is something that seems to get language, but it seems to just get it. If I'm right, that predated GPT-1, I'm not sure that's true.

**CRAIG** 25:36

You'll forgive me, I want to get to Cohere. But I, I'm a layman. My audience is somewhere in between me and you. I mean, they're, they're fairly sophisticated. But so you've got 20 lines of code. You feed it some data, let's say a sentence. How is it and it's it's relating within the neurons of the or the perceptrons of the multi layer perceptron? It's relating one piece of data, one word, to another word, how is it doing that? Does it is it? Is it by feeding huge volumes of data that it begins to see patterns? Or within that 20 lines of code, Something incredible is happening? Is it possible to explain that?

**AIDAN** 26:46

I think it's not, it's maybe one line of code that leads to that behavior. The other 19 are support. I would say the one line is is the objective. It's like what you're asking the model to do with the data. You're feeding through this, like hypercomplex pool of data. And what does it mean to feed it through? Well, what you're actually doing is you're saying, in the generative case, this is like the GPT style case, you're asking it to given all the words up to a point in a sentence, predict the next one. And that sounds simple. It sounds like stuff we've had for a while, which is like autocomplete tab autocomplete or no, it's like that that objective is horrendously complex. Because if I give you on the internet, there's examples of translation, right? Like these forums online where people teach each other how to speak different languages, and someone asks, Hey, how do I say, the brown dog in Spanish? And then stop, and then the person responds, oh, you say it by? I don't know how to speak Spanish, but whatever it is, right? And so if you ask your model to model this, the only way for it to accurately model this, it has to know how to speak Spanish, because it's seeing the English part saying hey, how do I translate the brown dog into Spanish stop. And now I need to produce the Spanish translation. And so you can see like, just organically, by learning to generate sequences in order, you're forced to learn extremely complex behaviors like translation, like classification, like writing code, you know, at the top of a piece of code, you'll have a function signature, you'll have a comment a docstring, saying, this function does XY and Z, it takes these inputs of this structure and outputs the following. And then if you're going to model that code, you have to learn to program because you're just given a function signature, and then a doc string that humans wrote for other humans to read. And so I think one of the most beautiful things that falls out of this is using this very, very simple structure, which is just hears a ton of data, learn to generate it, learn to predict the next, the next token, you're you think you're asking the model to do something quite simple and minimal. The reality is, you're asking it to do an extraordinarily complex task set of tasks. You're asking it to understand our culture, our language, the interactions between us your app, you're asking it to understand that data at the deepest level and so what you get out the other side is a model that, you know, roughly does understand and does have the capacity to do all that stuff does understand our culture. I think That's another one of these like, beautiful Simplicity's. Such a simple object. Such a simple object Pick, pick the next word. And what falls out of that what you're actually asking you to do. It's so extraordinary.

**CRAIG** 30:15

And when you're - so there's what five? Have you working side by side? How many people were working on the project? I think weren't there five or six names on the paper? I think there were eight or eight? Yeah. But in any case, you're it? Was there a moment? Or did you know, going in just from whiteboarding that, wow, this this could work? Or was there a moment when you were, you know, running tests that you began to see these extraordinary results and knew you were on to something amazing?

**AIDAN** 31:01

Yeah, there are definitely moments where like, someone would come running over from their desk and be like, Yo, come, look. And they had just run the eval. And it was like, it was state of the art beat everything that came before. And then we would all be like, next, okay, let's, let's keep pushing. And the funny thing is, it came together so quickly, it was really like over the span of three months. This wasn't like a year long effort or anything like that. It was just like super fast iteration pace. I don't know if there was a moment, I really don't think anyone fully grasped the significance. And that's mostly because the significance wasn't there at the time. The significance came from the fact that people adopted it, they could have adopted something else, they could have leaned into something entirely different. They chose a transformer for whatever sort of mimetic effects led to that. But they chose a transformer, they started investing, the community started investing tons of time in building infrastructure and support all the way down to the hardware level, for this particular architecture. And they enabled us to us being the entire, like aI community, to consolidate, consolidate on one architecture. And so I've said this before, and I, I feel quite confident almost everyone on the paper would agree. The transformer could have, it could have been another model. Frankly, it could have been another model, the transformer was just this bliss, they had the best support, and then the community reinforced that. And the community made some sort of decision to consolidate on this architecture and really invest in it, and they made it a success. It could quite easily have been another architecture that similarly scaled up, well saturated compute. Well,

**CRAIG** 33:20

you think there are other architectures out there that could that just haven't been discovered or explored? That could lead to such dramatic results?

**AIDAN** 33:34

Absolutely, like, unequivocally, I think, definitely. They exist, they're out there. And with enough work and effort, maybe we could flip to another architecture, but we've already done half a decade of infrastructure development and software support and you know, writing highly optimized kernels for the the hardware for transformers. And so there's a there's like this resistance to move, and it would take a lot of community will willpower to move away from the transformer. And the only thing that would motivate that is like some new substantial breakthrough at the architecture level. Yeah, so I don't see that happening. But I also don't make the claim that like, the transformer architecture is something like divine. Yeah, clearly, you need pieces.

**CRAIG** 34:34

I mean, right. But presumably, these large language models themselves could at some point suggest other architectures.

**AIDAN** 34:48

Yeah, people have wanted to use models in that sort of like feedback loop. Yeah, yeah. I think that's definitely we're already starting The chip architectures being decided by by models. No one's heard, right. Yeah. And so the chips train the model and the model change, you know, decides the next generation of the chip. And there's this feedback loop that

**CRAIG** 35:20

who's doing that?

**AIDAN** 35:23

Google mostly there V four or five TPU. chips were model placed designed. Yeah. So I think that's, that's exciting. That happens on a super slow timescale, because it just takes so long to actually fabricate chips, push them out, verify them. So that happens, too slow a timescale.

The stuff that you're describing, like the architecture search projects, I would say those have actually surprisingly been quite low yield. And that's probably because humans have spent so much time on neural net architectures, they've explored that space so thoroughly, and done a pretty like pretty compelling job at it. And so when we threw models at it, like, the gains were marginal always. Or, or they like rediscovered stuff that we had discovered previously, and kind of missed. And they just brought it to light, they surfaced it again. So people have kind of tried that. But it seems like in architecture space, it's actually it seems to have been saturated. Or perhaps the methods used, this was also a Google. Perhaps the methods used weren't the right ones. It's hard to say. But there was an effort to try to get models to produce new model architectures and have this self improving feedback loop. And I would, I would say that it largely fell flat.

**CRAIG** 37:05

So you, you went then from Google? Web, tell me about how you started Cohere?

**AIDAN** 37:15

Yeah, so I spent the better part of three years bouncing around. So I was in Mountain View for the transformer. And then I went to Toronto, and Geoff said, Hey, come come and hang out at Google and in Toronto. And then I graduated from undergrad, I went to Oxford for my PhD, Jakob from the transformer paper, he had actually decided to leave Mountain View and go back home to Berlin. And he was like, Yo, I'm going to set up a brain office in Berlin. And so I was like, Hey, that's pretty close, like a 40 minute flight from London. Let's work together. And so then I was on a plane every two weeks to Berlin to see Jakob and work there. And eventually eventually, I just realized, like there was a revolution kind of promised. Back when I was in Mountain View, just after we had released the transformer paper publicly Noam immediately started working on language modeling, and scaling the models up and he was like, actually deeply involved in the GPT one paper, he was helping OpenAI with it. And then I went back to Toronto, and I got an email from Lukasz. And he's like, Hey, have a look at this. And in that email, there was a Wikipedia article. And the title was the transformer and then I saw I was like, Oh, hey, this Wikipedia article on this I kept reading down and then with a Japanese punk band and consistently these members and this member had left and I was just like, What the fuck like Lukasz What is this? He was like But transformer wrote this I just put in the transformer as the title everything else. And that was just like, You're kidding. Like, it was like surreal. It was just like, you know, you went to bed one night and models could barely spell and then you woke up the next morning and they were writing as fluently as a heat like such a plausible story about a Japanese punk band called The transformer and I I think that was like the moment that I was like, Okay, this unlocks in product space this unlocks something categorically different like it just something extraordinary.

**AIDAN** And I thought it was gonna happen. And I waited and I waited and I you know, I was In my PhD, and I was putting out new research and proving fundamental methods. And after three years there, nothing had changed, the world was the same. And Nick and Ivan, my co founders, like, I think we all felt the same disappointment. Nothing had changed. We saw something magical three years ago, and nothing had changed. No one's talking about it. And so eventually, that disappointment turned into resolve to do it ourselves. And so we decided, okay, let's leave. And let's go build cohere to bring this to the world. This is before GPT-3, just after GPT-2, in in 2019. And back then the mission was really just a, this is the most amazing technology that humans have ever created. Let's model the web, let's build a model of the entire Internet. And be, let's put it into the hands of every single developer on Earth. And Let's inject it into every single product and just create a new generation of magical product experiences. So that was really the seed. Yeah.

**CRAIG** 41:29

And then, so Cohere is, at its core, a large language model, or a suite of models. For different vertical tasks are what describe what what it is, and how people use it.

**AIDAN** 41:53

So at its core, yeah, it's like a, we're an intelligence factory, building these big models, making them as usable, as usable as useful as possible. There are like a suite of models, we have both sides of that coin that I was describing before, where there's the generative and then representation, so both styles representation, and GPT styles, the generative side. So we have both of those, and we build them in house.

**AIDAN**

The way that we bring them to the world is that we partner with enterprises, and we solve really, what what are some of the today's largest blockers for adoption, which are privacy, privacy blockers, data compliance blockers, if you're really gonna put these large language models into useful applications at the forefront of your product, they're going to be touching data that's the most sensitive, like user data, right, like people's private data. And so that very, very high security bar. So for us, one of the benefits of being independent, our competitors mostly are sort of bound to one cloud provider. There's exclusivity there. For us being independent means we can play with everyone. And with the enterprises that use us, they don't get vendor lock in. So they're not trapped into one cloud provider. They can bounce between, and we can deploy wherever they go.

**AIDAN**

So for cohere, one of our core efforts right now is making it so that these models can be deployed on any cloud provider, in situations where the data is the most sensitive, because that enables the most interesting and impactful applications. Otherwise, you you kind of get what I've been seeing a lot of recently, which is superficial deployments of these models, not real, not product changing, not like fundamental shifts in infrastructure, but more like, here's my product, and I'm just tacking it on to the side. Here's like a delivery experience. I think that makes a lot of sense, given the fact that this year everyone just kind of like woke up. And so it's gonna take a while to actually replace this with the the thing that we want. So it makes sense.

**AIDAN**

But really, the piece that's blocking this is the fact that there's not a lot of trust in some of our competitors due to the fact that in the past, they've trained on their user data and they disintermediated people. And so for us, we want to regain that trust and be the trusted partner for enterprise to actually bring large language models into like a truly transformative way. So I think there's like right now. There's a product transformation that's kind of similar laying under the water, because the whole world just woke up, every single company now is trying to figure out what does this mean? What does this technology mean for my product? My experience? What am I users? My the consumers? What are they going to expect from me? How do I not get left in the dust by my competitors who are going to reinvent their product on the back of this technology? So they're starting to do the work.

**AIDAN**

in 18 months, product space is gonna look completely different, because right now, everything is shifting behind the scenes. And so for cohere, we really want to power that transformation, and be a trusted partner to the largest enterprises and the best developers on Earth.

**CRAIG** 45:49

And and enterprises span the gamut of industrial verticals, or are you focused on one industry,

**AIDAN** 46:02

it's totally, totally horizontal. So it impacts everything. Like, I think you're going to be doing your banking with a conversational agent, you're going to be doing your shopping with a conversational agent, I think it's really hard to think of a particular vertical or industry that's, that doesn't need to be changed by this, because consumer expectations are going to be, there's going to be this interest, when I show up to this new product, there's going to be this interface that I expect, which is language. So with these interface level changes, and in the same way that if you're a product or a service, you have to have a mobile app, because everyone's on their phones. And that's, you know, how they want to interact with products and services, in that in that same way that that mobile transition led to everyone having to support this interface that the consumer expected, everyone is going to have to support conversation and dialogue with an intelligent agent, as an interface onto your products and services. So there's like this resurfacing of product space that is literally happening right now.

**CRAIG** 47:11

Is there an example without naming names that you can give that you think is gonna blow everybody away?

**AIDAN** 47:21

I mean, it's no secret. It's no secret that we're starting to see some very compelling assistant like offerings. There were the promises with Siri and Google Assistant and Alexa that came 10 years ago, or whatever it was. And those fell flat, and I think the technology truly just was not there to support it. There is now the possibility of a truly general assistant. Like we actually have the technological bedrock to support that. it's emerged recently, it's a fairly recent development that that has been unlocked as a thing you could possibly build. Yeah.

**CRAIG** 48:19

You know, I talked to Ilya about RLH F reinforcement learning with human feedback, his way of kind of guiding the model toward more grounded responses. But I've talked to other people who say that's still speculative and and takes a lot of time and they're, they're using vector databases and loading vector databases with authoritative data. And then the language model in effect is just the the mouthpiece it's it's not it's it's not calling up the answers from it's accumulated knowledge it's referring to this vector database How do you guys deal with hallucinations

**AIDAN** 49:25

Yeah, I there's like there's someone Sarah hooker echo here. She said this before and I really I really like it. You have to distinguish between the hallucinations that you want which are like creativity, and the hallucinations that you don't want it like it's great when it hallucinates a story or a new joke or you know, you want that and so you don't want to like beat that ability, that capability out of the model. At the same time you need ways to control it. So for instance, if you're doing knowledge gathering or research, you definitely don't want anything made up. There's like almost zero tolerance for hallucination.

**AIDAN**

And so you kind of want a gradient, or a parameter that you want to set, which might be the creativity parameter. And I think that's becoming increasingly possible. Another another really good way to get models to be more truthful, is to actually force them to cite their work. So there's Patrick Lewis, he was the first author of meta on creating rag. It's called retrieval augmented generation. And so that's this idea that you have a model. And you have an external knowledge base, or maybe multiple external knowledge bases, maybe one's Google ones, your private emails, ones, your blah, blah, blah. And what the model can do is it can go out, and it can query these sources. So it can say, hey, the user just asked me about this, I think I should query Google. And then it gets back from Google some documents or gets back from your email, whatever emails you're looking for. And then now that it can read those, it can generate a response, and it can cite back to them it can say, hey, you asked me this. I think this is the answer, because of this sentence inside of this document or this webpage. By forcing the model to learn to cite it sources, you get two things. One is the fact that you can actually check its you can verify it right? You can check that it's telling the truth, you click into that link, you can read the thing, and you can say it lied. Or you can say, oh, no, it's right. Yeah, you know, that checks out. So one is you get it two sided sources. The other thing is that you force the behavior, you reinforce the behavior into the model of not making claims without grounds to those claims. And so it starts to learn the scenarios where, you know, when I'm writing stories, I don't really need to cite sources, I just need to write and the user is happy and content and you know, I get a good reward. And in the scenario of ham doing research on a topic, can you tell me about X, it starts to learn, okay, shit, in this case, I need to have a very rigorous bibliography, I need to be able to tie that back. And if I mess up, if the user clicks through and sees a, an error or a hallucination, I'm wanting to get a super strong negative feedback. And so it learns to differentiate between these scenarios. So I really do believe retrieval augmentation is going to be one of the key pieces of along with human feedback, it's gonna be one of the key pieces of making these models more reliable, more grounded.

**CRAIG** 53:07

That's fascinating. I'm coming up to an hour. Can I ask a few more questions? Yeah. Yeah. I've got to ask this, you know, this has set off sort of the public release of chat GPT. has set off this debate about how dangerous these models can be. To everyone's surprise, Geoff has gone public saying some really dire things. Which, you know, I don't know, like you do, but I've known him for a while and it's It surprises me. I've never heard him speak that darkly about something. Do you have a view on that? That's one question. And then the other is this debate about sentience or self awareness? I mean, you've you had your fingers in the, in the brain of of these things. Do you? Do you think that sentience or self awareness could really emerge? Or do you think that you know, these are bits of code and it's all an illusion?

**AIDAN** 54:40

There's a lot to say that we need we need another hour or two together to properly represent my beliefs around that question, I think the first part for Geoff Geoff is like Geoff went through the same thing. I think many of us in the field went through where our timelines got pulled forward massively. And so it you know, we thought we'd have models that could write compelling English and a few decades and then suddenly it shows up a year later. And so that throws you into this state of shock and uncertainty. And you're quite caught off guard and he's spoken about this I think publicly. The scent sentiment of surprise, progress and rate of change. I remember having conversations with him myself, where both of us were kind of like these people who talked about AGI you know, what, nonsense, haha, this was back when models could barely spell. But then you kind of get surprised and shocked and your your uncertainty blows up. And sometimes that can have the effect that okay, anything's possible. Oh, my God, like I was so far off on that. But now I am shooting up my uncertainty across all anything could be possible super intelligent god. Okay, maybe that's even. So I think like, a lot of folks. We're all reckoning with that and recalibrating. And, you know, adjusting our, our own timelines and understandings of progress and pace.

**AIDAN**

Geoff is extraordinarily thoughtful. And he's been thinking about this, since at least the beginning of cohere. So at least the last three and a half years, he's been thinking about this very, very deeply. So I think people should take him very seriously. I think there will be a lot of sensationalism. And a lot of extrapolation from what he's saying. But if you actually listen to what, what he's actually saying, it's quite a measured, he's like, I'm highly uncertain about what can happen. And that means we should take this stuff seriously. Because we just don't have certain bounds. We don't have certainty around the future. And so we should be taking all the different possibilities, quite seriously. Not saying that they're likely to happen. But just saying that we they can't be ruled out yet. And so let's take them very seriously. I think there's a lot of journalistic texts and headlines and clickbait and nonsense. But if you actually listen to Geoff, I think his take is quite measured and reasonable.

**CRAIG** 57:47

And, and actually, I'd love to have you back on to talk at length about these things, but on the idea of sentience or the illusion of sentience. I mean, you've, you know, more than almost anybody having built these models, and both what they're capable of and, and what's behind their expressions. Do you think that? I mean, it's a philosophical question about what what sentience or consciousness is, whether it's, you know, whether our consciousness is just an emergent property from the neural activities of our brain and, and, and it's largely illusion. I mean, just what would you say to all of that?

**AIDAN** 58:54

Yeah, I would say, I don't place like a divinity on on humanity. I think that consciousness is in the brain. And it is like a physical process. And it's maybe like, maybe consciousness is what computing feels like, like, what processing feels like? And if that's the case, it's really hard to argue that that same phenomena couldn't be present in silicon. I think you'd really have to, I think there has to be a leap right to say that the circuits in our brain because they're human, or because they're biological have some sort of fundamental distinction. I think you really have to take a leap of faith there. And so if I'm just saying being pragmatic and reductive. Again, we need two hours to discuss this, I think more completely, but I think it'd be really, just as a scientist, I think it'd be really hard for me to say, There's no way these machines could become sentient. I just I can't construct an argument that and that.

**CRAIG** 1:00:27

Yeah. Yeah. Well, let's leave it there. But Can Can I get a promise that you'll come back? You know, in a few months, and we can go deep on that subject. Yeah, I'd love

**AIDAN** 1:00:43

to. Yeah.

**CRAIG** 1:00:46

Okay. Yeah. Aidan. This has been really fascinating. I'm delighted. And I'm sure you heard at the MIT Tech Review Conference. Somebody asked Geoff, he was on virtually from the UK, but somebody asked him whether he would divest himself cohere. And he said, No, no, he's, he's gonna stay invested. So yeah. Yeah, that's a that's a funny question. Yeah. Yeah. Okay, great. Well, I really appreciate your time and, and we'll talk again.

**CRAIG**

That's it for this episode. I want to thank Aiden for his time. I also want to remind you to check out NetSuite Oracle's business management software for enterprise resource planning, financial management, customer relationship management and E commerce, among other things. Go to netsuite.com/eyeonai to take advantage of this offer.

**CRAIG**

And remember, the singularity may not be near. But AI is about to change your world. So pay attention