/mlst

The Gap Between Humans and Machines Is ___ - Dr. Max Bartolo

March 23, 2026



Machine Learning Street Talk

Podcast Transcript

Revision: eed7e35c

Topic headings include retention indicators: • above average, • average, • below average (vs. similar YouTube videos)

Interested in licensing ReScript for your podcast? Get in touch with Tim Scarfe.

Contents

Overview 2

Model Reasoning and Consistency Verification 3

Influence Functions and Distributed Knowledge Analysis 4

AI Application Development and Model Deployment 7

AI Alignment and Human Feedback Limitations 8

Human Evaluation Challenges and Factuality Assessment 10

Cultural and Demographic Influences on Model Behavior 12

Adversarial Examples and Model Robustness 14

DynaBench and Dynamic Benchmarking Approaches 17

Benchmarking Challenges and Data-Centric Evaluation 19

Cohere Command A Development Process 21

Model Quantization and Performance Evaluation 23

Reasoning Capabilities and Training Progression 25

Context Windows and Enterprise Applications 28

References 31

Overview

Dr. Max Bartolo from Cohere discusses the gap between model capabilities and genuine robustness: why next-token prediction can produce impressive results yet still fail on slightly reformulated questions, and what that tells us about whether models truly reason.

The conversation starts with recent work by Laura Ruis using influence functions to trace which pretraining documents models rely on for factual vs. reasoning queries – finding reasoning draws on far more distributed knowledge than simple fact retrieval. Bartolo explains how Cohere's research approach centers on post-training, adversarial data collection, and iterative evaluation loops that reveal where model performance saturates.

A substantial middle section covers the limitations of human feedback: Bartolo's paper with Tom Hosking and Phil Blunsom ('Human Feedback is not Gold Standard') shows human evaluators are surprisingly inconsistent, that inter-annotator agreement is often treated as gold standard when it shouldn't be, and that crowdworker feedback systematically misses factual errors. This leads into the PRISM project exploring how demographic and cultural factors shape what people want from AI, and why single-population alignment is insufficient.

The adversarial ML portion covers Bartolo's DynaBench work: dynamically generated benchmarks where humans actively try to fool models, creating harder and harder test sets. He reflects on what worked (human-and-model-in-the-loop data collection scales adversarial difficulty) and what didn't (limited community uptake). The discussion extends to Sara Hooker's work on compute-aware evaluation and DataPerf's data-centric benchmarking approach.

The second half turns to Cohere's model development, covering Command A architecture choices, model quantization tradeoffs, and the observation that smaller quantized models sometimes outperform larger ones on specific tasks. Bartolo describes the training pipeline from base model through SFT stages and synthetic data generation, the evolution of context windows to 256k tokens, and enterprise deployment challenges including RAG search, multilingual support, and convincing enterprise customers that probabilistic AI systems can be reliable enough for production use.

Throughout, Tim pushes on whether models are really reasoning or merely doing sophisticated retrieval, and whether the field's evaluation methodology is rigorous enough to answer that question.

Model Reasoning and Consistency Verification

Max Bartolo 00:00:00

The expectation of machines is consistently their right all the time. It's a function of traditional software. If you had a calculator, you put in 1 plus 2, the calculator said 3, and then you try 2 plus 7 and the calculator said 17, you'd you'd throw away your calculator. You wouldn't you wouldn't try any other questions. We need that consistency and we need that robustness. The next step is really convince a human that the model is actually consistently doing the what is being asked to do. Do models reason? Maybe they do. I think reasoning is is a bit of an overloaded term. It's used to mean many different things. I don't even know if that's the right question. Do we really care if models are reasoning the way we might think? If humans are consistently successful at getting the model to fail on examples that, you know, are slightly different, then I think it's fair to assume that 1 isn't really reasoning. It's just, you know, figured out how to do well on a specific benchmark.[12]

Tim Scarfe 00:01:02

MLST is sponsored by 2 for AI Labs. Now they are the DeepSeq based in Switzerland. They have an amazing team. You've seen many of the folks on the team. They acquired Minds AI, of course. They did a lot of great work on ARC. They're now working on o 1 style models and reasoning and thinking and test time computation. The reason you wanna work for them is you get loads of autonomy, you get visibility, you can publish your research, and also they are hiring as well as ML engineers. They're hiring a chief scientist. They really, really want to find the best possible person for this role, and they're prepared to pay top dollar as as a joining bonus. So if you're interested in working for them as an ML engineer or their chief scientist, get in touch with Benjamin Crusier. Go to 2forlabs.ai

Max Bartolo 00:01:44

and see what happens. Welcome to MLST. It's an honor to have you here. Thanks, Tim. Thanks for having me. Introduce yourself. I'm Matt. I am a researcher at Cohere. I work on many things, primarily post training, but I'm also interested in adversarial data collection, better ways of collecting data for improving models, valuation, which kind of drives this continuous feedback loop where we, you know, target what we care about in terms of model performance and then figure out how to, improve, performance on those dimensions. And then often you

[1] Max Bartolo Website — person
[2] Cohere — company

kind of see performance saturate, relatively quickly, and then that point, you're need to figure out, well, what's the what's the next layer of of evaluation that we care about? You're kind of iterating in this in this continual process. I'm particularly interested in getting models to reason to operate more robustly and to generally be more useful. I think reasoning is is a bit of an overloaded term. It's used to mean many different things. I don't even know if that's the right question in that do we really care if models are reasoning the way we might think, you know, humans reason. And I think there's debates about whether or not we reason in the ways we think we reason. But it's clear that models trained with next token prediction can do some very impressive things. And whether or not actual reasoning is happening under the hood, I think is an interesting research question, but not really a bottleneck in terms of getting models to the things we would like them to do. So, yeah, we have this this recent work with Laura Rees. She worked worked with us at Cohere, and she investigated during pretraining what types of reasoning or what types of information

Tim Scarfe 00:03:21

models we're learning from, you know, procedural knowledge in in pretraining documents.

Influence Functions and Distributed Knowledge Analysis ●

Tim Scarfe 00:03:25

The elevator pitch that she said to me is, know, I think many of us, we've been quite skeptical about reasoning, right? You know, I'm inspired by Cholet Yeah. And we say that these things are, you know, not quite not quite hash tables, but you know, they're doing some kind of curve fitting, some kind of retrieval or something like that. And Laura said that she went into it thinking the same. Yeah. And and she was quite surprised by the results. So what what did she find? We we went into it thinking

Max Bartolo 00:03:52

that models basically just retrieve facts from think of it as as kind of a compressed version of all the information that they're pretrained on, and they were basically relying on this parametric knowledge, which is the the this knowledge stored in the most parameters, to to answer most, queries that look like reasoning queries. So what we did is we used something called influence func-

tions, which basically you can think of as a way to approximate the effect of a particular training example on a model's behavior. And we had roughly 40 factual questions. We had another 40 reasoning queries. And 1 of the findings, and there are quite a few interesting findings, was for the factual questions, the models were really mostly relying on documents that contained the answer to the question. And, you know, at most, that'd be a few documents that the model was relying on in order to answer a factual question. But for the reason queries, there it was a lot more complex. Right? There there was information spread across many more documents. Many times, documents that contain the types of reasoning that would be required to answer the question in the first place. And it was just much more distributed. So the the influence kind of distribution was a lot more spread out. And it definitely gave the impression that the model was just relying on procedural knowledge that it had picked up from these various different sources and was potentially combining them in in interesting ways. Yeah. Like, went into this research expecting to find the opposite, and I really, you know, quite enjoy work where the results really challenge your way thinking about the problem. And I think definitely for me and for Laura as well, it's it's altered the way we kind of perceive how these models operate. The the kind of counter to that is possibly, well, there's all these reasoning queries that are of very similar. And so maybe it's just the case that models are just relying on these different documents because of the similarity of the queries. So we had a set of control questions as well, which looked kind of lexically and structurally very similar to the actual reasoning queries but didn't require any reasoning in order to answer them. So I'll I'll give an example. Right? Say we have a question such as, you know, a line is defined by points 2 comma 2 and 3 comma 3. What is the slope of the line? You know, you need to kinda figure out what the right equation is, figure out that the slope is 1, and give the answer. And to do that, you might rely on information you've picked up from these documents. And the control query would look something like the slope of the line is 1. What is the slope of the line? Or, you know, a line is defined by the fact that it has a slope of 1. What is the slope of the line? Sounds and and kind of is structured very similarly to the the question we care about, but doesn't require any reason because the answer is just provided there. And what we saw is a big contrast in terms of the way models processed those those kind those kinds of information. So that really gave us, you know, a strong indication that the model was actually doing some some form of reasoning in order to calculate these these slopes. 1 of the challenges with influence functions is they they just naturally don't don't scale very well. In the in the very naive sense, you can calculate influence of an example by retraining a model without that example and comparing the difference. That's obviously extremely expensive. And influence functions themselves kind of require the computation of of the inverse Hessian, and that's useful because it

gives you this kind of second order information about the the loss landscape of the model. So it gives you a sense of the the curvature and the structure effects on the on the loss landscape of including this additional example that you care about, and it's it's just extremely expensive. Right? Hestine is like a big matrix. It's got dimensions in each axis, which is the number of parameters in the model. We're working here with with 7,000,000,035 parameters, so it's just extremely computationally costly. And so we use basically built on work Ocean Tropic released late last year. They scaled up kind of approximate curvature estimation using ECFAC, which was released somewhere around 2018, I think. And and so then that really allowed us to, along with some other optimizations, you know, scale this up to quite quite large model sizes. And I think on the on the note of model size, 1 aspect we found also very insightful was that it seemed like the you know, that there was very little correlation between which documents were influential for the 7 b model and which were influential for the the 34 35,000,000,000 parameter model. And and we didn't really have much time to dig into the details of that, but that's also extremely interesting. It's it's it's indication that models of different sizes are just potentially learning things in in different ways. What is your definition of reasoning, and what what what would it mean for a model to reason really well? I think it comes down to so for me, reasoning and robustness are kind of interlinked in that if you can reason, then I would expect, you know, correct reasoning to imply robust reasoning. And I'll like, take an example. Say we're say we're going back to the, you know, calculating the slopes of lines. If a model can really reason through that relatively straightforward process, right, it's it's you need to understand what what function to apply and then apply that function consistently. And if a model does that 999 times out of 1000 but fails 1 out of those thousand times, to me, it really starts to question like, bring into question, is this model actually reasoning in a way that I might expect a human to reason? And the the counterargument there is always, oh, but humans are never. They're not infallible. Right? They always kind of make mistakes and get things wrong as well. Machines have a bit of an advantage there. You know, they don't get tired. They don't have the limitations of being human, I guess, to to contend with. And so I'd I'd really expect a a reasoning requirement for a kind of specific type of reasoning or a specific definition[3][4]

Tim Scarfe 00:10:26

to really be be robust and consistent.

[3] Influence Functions in Machine Learning — paper
[4] Studying Large Language Model Generalization with Influence Functions — paper

AI Application Development and Model Deployment

Tim Scarfe 00:10:28

Yeah. I think so many people are starting to see now how valuable this technology is. I mean, I'll give an example like this this app for tracking the podcast. I wrote it in half an hour. Right? Just just using using language models. And we need to reimagine many of the apps. I I think the first strategy was rubbing AI into the existing apps, and now we're seeing a whole new generation of apps that are that are AI first. And it's like the iPhone moment, and I I think people are starting to to to really see that. But let's talk about test time training just just real quick. You know, it's been heralded as a as a new scaling law. And certainly on the ARC challenge, all of the winning, you know, results were this kind of transductive active fine tuning, which is that, you know, rather than using a global shared model, which was trained once by someone else, I'm gonna take that model and I'm gonna fine tune it in my situation. I've now got a model that works very well in this situation. Yeah. And it seems to be like a much more distributed paradigm of doing AI. But I mean, obviously, you work for Cohere, which is quite centralized at the moment. I mean, how do you see the future of having big centralized models, but also having more of a diffused kind of training methodology?

Max Bartolo 00:11:38

I think we're just going to see more and more. I think the bottleneck so far has been, you know, how many people have the kind of knowledge and skill set and the experience to train these kind of models. And as kind of more and more people get excited by AI, by the kind of promises that it it holds, The, you know, the work around, you know, sharing these models, like the Cohere's models, for example, have the way it's openly released. You can you can kind of download, you know, Cohere's Command R Plus or Command R model off of Hugging Face. You can continue fine tuning that, And and we've seen lots of people doing that. And I think I think there are some constraints. You need to figure out, you know, what the optimal way of of doing that is without potentially losing, you know, general knowledge capabilities that might exist in the base model. But it's there there's 1 interface which holds across models and across humans as well, which is language. And if you can and and we know that you can do this to to a very real extent. If you if you view it from the point of what we're doing over time is making the data better in terms of what we can learn from, then that that unlocks, you know, possibilities globally, like pretty much everyone. If you had the right way of using models to improve the data and have kind of a shared data resource,

could train a model from that data assuming they had the know how and the[5]

Tim Scarfe 00:13:19

compute. Yeah. It's it's almost a mindset thing. I mean, I've written apps now where it's it's easier to hook it up to a chatbot interface than it is to write a UI because you just have this insane amount of sophistication. I I use OpenInterpreter for example, and I've now adopted this pattern where in OpenInterpreter, I just load my my Python multi agent system into memory and then I just tell the agents to reveal their interface and then OpenInterpreter just works. Yeah. So I can manipulate my database and I say, you know, add something to the table of contents and like add this reference, delete this reference and it just works. I don't need to do anything. It's a completely different way of thinking about software, and I think this is gonna take us a long time to figure this out, but it's it's genuinely amazing. And also, you can have second order intelligence. So rather than talking directly to the interface, you can be talking to another agent, which itself is doing intelligent things. It's it's just mind boggling. It's like it's like the matrix. Possibilities are endless. Yeah. It's it's insane. On AI alignment, before we move on to this, because you were talking about some of the the the social implications as well before. I mean, of all, what do you think about AI alignment? Obviously,

AI Alignment and Human Feedback Limitations

Tim Scarfe 00:14:24

very important. Right? We want

Max Bartolo 00:14:26

AI systems that don't do bad things. That's kind of the starting point. We don't want AI systems that, you know, damage our environment and make our societies worse than impact humanity negatively. I I think the question is really what what do we want to align AI systems to? We can you know, typically, define alignment as having models whose kind of values or behaviors are aligned to what humans want or what humans expect. But if you break that down, I'd argue it's it's probably already very hard to figure out what do humans want and what do humans expect and do all humans want or expect the same things. And is that kind of static in context, or does that change depending on the context you're in? And so it's it's a very complex problem that

[5]Command A Model — company

we need to solve, and and we haven't necessarily solved that societally with for humans either. And and now we have to solve it for, you know, this this new technology that that's kind of permeating our lives. And so I I think I think we need to we need to figure all of that out, but we also need to have some basic safeguards in place to make sure that, you know, the the the technology remains safe for for general use so that we can continue to advance and,

Tim Scarfe 00:16:05

yeah, and develop and make sure we we make the best use of its potential. Let's go over to human feedback. So, you you wrote this, paper with with Tom and Phil called human feedback is not a gold standard. Can you can you tell us about that? The human feedback is not gold standard,[6]

Max Bartolo 00:16:20

paper was motivated by, you know, all the interest and excitement around using human feedback pretty much throughout the model training and evaluation pipeline. And so, you know, a couple years back, RHF was a big thing. It was the the kind of new thing. Everyone was excited about it. And the the general way of doing RHF was, well, we give humans a prompt, and they see 2 completions. Right? So say, what color is the sky? Completion a is blue, and completion b is yellow. And a human annotator looks at those 2 completions and says, I prefer completion a. Hopefully, they have good reasons for preferring a completion over another. And and there was massive massive progress we saw in many of the benchmarks from from this idea of of using human feedback to optimize and kind of fine tune fine tune these models. And we quickly started to see diminishing returns from this. And the effect was really, at least initially, that the style of the output that humans preferred was providing better results. And in many cases, the better results were also measured in some way by human preference. So you either had, you know, a human evaluation task where humans would look at output from the model and say which they they prefer in order to kind of rank models against each other, or you train a reward model and use that kind of as a proxy for for what a human would prefer. And this idea of a single preference score that captured everything about a generation just seemed quite quite limited. And what we started out looking at was what if we make that feedback signal more granular? What if we try to understand what about specific model generations do humans prefer over others? And so the first experiment we ran in that paper was we categorized these different error types. We identified examples in kind of established data sets which to which models generated completions which contain these errors.

[6]Human Feedback is not Gold Standard — paper

And then we asked 1 group of humans to rate these completions by whether or not they had certain errors present. And we asked a different and independent group of humans to annotate a rating score of those completions from from 1 to 5. And then Tom ran some very detailed analysis and tried to figure out, well, what types of errors contribute to human preference, judgments of quality, of overall quality. And what we found was quite interesting. We found that humans really, really dislike if the model refuses to answer a question. That thinking about it seems extremely intuitive. Human preference judgment is strongly informed by formatting and by style. And we see that in some of the the early work as well where, for example, a lot of the early summarization work seem to suggest that humans preferred longer summaries, which which is kind of also quite intuitive because longer longer summary contains more information and even just by by that fact is more likely to contain more relevant information from the summary. And what we found interesting was that certain features when, you know, say, we at Cohere, we talk to enterprise customers, and they care that the models are correct. They care that the models aren't repetitive. They care about certain properties that might be different from what an annotator looking at 2 different completions might care about. And attributes or criteria like factuality were were quite

Human Evaluation Challenges and Factuality Assessment

Max Bartolo 00:20:17

lowly ranked in terms of their contribution to overall quality. And so we thought, well, that's that's potentially problematic. Right? We, to some extent, optimizing these models to give more, you know, elaborate, more interesting completions at the expense of them potentially being less correct. The the follow-up to that was then we wondered, well, could it be that there are also certain confounders or other effects that we're not we're not noticing? And we had this this hypothesis that maybe assertiveness and complexity of model outputs might impact how humans perceive the quality of a completion. And so we ran a very similar experiment. What we did is we prompted these models a very simple 1 line prompt, something along the lines of, you know, give the response, but, you know, sound very confident. What we found was that, you know, there was a massive effect in terms of the the generations that were designed to sound more confident were were consistently rated as higher quality and as having fewer errors. And there's there's a particular plot in the figure, which I think shows this quite nicely, where you have this almost almost like an elbow curve where you see the error rates on

the y axis, and on the x axis, have kind of the the assertiveness value between 1 and 5. And and as assertiveness reaches a certain point, the error rates practically tend to tend to negligible. And we we know that this is also kind of a a trait of of how humans interact with other humans. Right? If if if someone sounds very confident, they're more likely to be perceived as, say, an expert in their field. There's quite a bit of work in this space. We didn't necessarily think that the effect would be so obvious when humans are interacting with AI systems. And in many cases, these are trained annotators. They've been given very clear instructions about what and how to infer the quality of these different completions. 1 1 other striking point from that work for me was this effect of assertiveness on perceived correctness. The the more assertive generations were generally perceived by humans to be more correct. So we we kind of look through a sample of these these examples judged, you know, as as expert annotators and inverted commas, mostly Tom kind of going through the data and and rating, you know, whether there were any factuality errors. And what we found actually was that when the model was prompted to be more assertive, it tended to be less factual. And even though it was less factual, because it was more assertive, annotators, you know, got the impression that it was actually more correct. And so that that to us was striking because we have this very clear example of something counter to the behavior we want to optimize or the behavior we want to incentivize in the model, but also doing so in a way that is barely recognized. The annotators doing this didn't realize that they were being influenced or biased in these ways. And that's, I think, representative of just this very large scale effort of optimizing towards human preference with a very kind of specific and to many extent underspecified

Tim Scarfe 00:23:56

definition of human preference. And isn't it funny because we were saying earlier that humans are rational. Right? But you've just given us many many examples of how irrational we are. Right? We have this style bias. We see something that's assertive and and we think it's correct and and so on. And you also gave an example of, well, you know, what if what if we tried to get annotators to compare 2 entire books? Well, that's clearly not only beyond our cognitive horizon, but just the the ambiguity just goes up exponentially. It seems like it wouldn't be possible for us to do that in in any reliable way. So what can we what can we do here? You know, you you said let's have more attributes. But how far do you go down that road? Because won't you end up with thousands of attributes? Yeah. For sure. And and I have a a similar problem. Right? I'm I'm,

Max Bartolo 00:24:44

Mediterranean culturally. I just like answers to be very direct and to the point. Right? I'm the kind of person who if you ask me, what is the color of the sky? And I had blue and the color of the sky is blue, I would prefer blue. And that's that's not what most humans would would prefer. And you can you can definitely go, you know, very deep down the rabbit hole of, well, what, you know, what are the criteria? What are the dimensions? What are the things we care about? And I don't I don't necessarily think that that's a problem. I think 1 1 area I'm very interested in is this idea of modifying model behavior to suit kind of personal requirements on the fly, right, without additional fine tuning. So if you have all of this data, imagine I could build some kind of, you know, portrait of Tim's preference in data. Right? There's these 100 data points that reflect how Tim wants to interact with the model. And and maybe those can change, and they probably will change over time. It's customized. It's specific to you. It's kind of that captures all of the information of what you personally are looking for in your interaction or the behavior of the model you're working with. And you could do something relatively straightforward if you wanted to, like, use that as part of the prompt or part effectively doing context learning. Right? Provide the model with that at inference time. Say, this is temp. These are his preferences. Generate according to those preferences. And models we've seen are relatively good at conditioning on, you know, even complex information in input space. Like, generally, this idea of models built for specific people for specific reasons without having to go through the overhead and the cost of training a model for every individual on the planet. Right? If we had to train 7,000,000,000 GPT-four's and serve those models, that would be very prohibitive from a resourcing perspective. But if we could just train 1 model, which was extremely good and, again, extremely robust and really conditioned well on kind of the requirements for an individual, then we could do that relatively cheaply in kind of relative terms. Yeah. So this kind of brings me to Prism,

Cultural and Demographic Influences on Model Behavior

Max Bartolo 00:27:15

which is this project led by Hannah Kirk. Just won a best paper award at NeurIPS. And the idea there was, well, let's figure out how at scale these different attributes, right, demographics, cultural kind of influences, you know, geographic effects, linguistic effects impact what it is that humans care about and how humans want models to to behave. And, you know, it's super interesting work. There's there's

a lot in there. There's over a 100 pages of content, so I I highly recommend that everyone, you know, reads that paper at least twice. It's it's extremely rich in terms of the insights it provides into, you know, this whole space of of understanding what human preference means and what it means for different people. There there's there's some extremely interesting findings, 1 of which is just representation. So even even something as simple as what do people converse with models about is just highly influenced by, you know, their backgrounds or by who they are as people. And you might argue, well, that's potentially not such a big deal. But you you you always have to see model development in this kind of whole, you know, cyclic feedback loop effect where the conversations that people are having with these models will drive the improvements to the models. And even if your usage is not well represented in kind of standard post training data, then the models are just gonna be worse at the things that you care about. Then it will be, you know, something that more people care about relatively. So, you know, there were there were examples like this this kind of work was run over something like a year and a half. So it was, you know, quite quite a huge project. And there were examples along the lines of, you know there there was a substantial amount of conversation around the Israel Palestine conflict, and most of that was happening from, people based in The Middle East. And if most of the data going into these models does not represent even something simple, we're not even talking about human preference here in terms of completions. We're just saying, what is it that we're interacting with these models about? Then you're just going to have all these kind of biases and problems down the line where the model doesn't serve everyone. It serves the people who who have built it to some extent or the people who have contributed most to using it. Then yeah. So this work again is very, very detailed. There there's a lot of analysis about what about completions or, like, model generations impacts human preference. So there's quite a few findings similar to the human feedback as not called standard work. The the same kind of general sense that, refusal and formatting and style play strong effects. Like, generally have the sense of averaged out a rank of models that we consider the best and then kind of the second best and then the third best. And there's a set of, I guess, established benchmarks or established methods of evaluating models like like, you know, the childbirth arena where, again, it comes back to to what is human preference really capturing. Right? We have this this almost canonical ranking of models, but it's heavily influenced by who the people providing the judgments are, what they're asking about, and what their personal preferences are. And so what Prism does is it really breaks down what, like how ranks or preferred models change depending on who's asking the questions or who's interacting with the model. And yeah, you see big, big changes. And even in terms of which models are better, you see models kind of moving 5 5 positions up the kind of ranking or 5 positions down the ranking or more. And there's 21 21 models in in the work, so that's quite quite a big change. And that

that can be very insightful for model developers as well if you, you know, fundamentally believe that you're building technology to serve all of humanity. Our coherent models, for example, command, this is, like, a very early, like, first generation of command models, Performs, you know, very well across most cases. And then for some reason, for users in Asia, performance just seems to drop relative to the other models. And that kind of insight can be extremely informative for trying to figure out what is it that we can do to help serve, say, people in that location better. And and, you know, based on insights like that, we've taken many actions to improve how we

Tim Scarfe 00:32:41

go about building these models.

Adversarial Examples and Model Robustness

Tim Scarfe 00:32:43

Yeah. I mean, Andrew had this really good paper out called adversarial bug sorry, adversarial examples are not bugs. They are features. And he was saying that they learn these non robust features, which are basically these like, you know, weird blue pixels mean Mercedes and stuff like that. And they they happen to generalize really well. But from a from a reasoning point of view, they're they're clearly doing it for the wrong reason. So it just happens to be the case that that they work now. But when there's a distribution shift, they'll they'll stop working tomorrow. Exactly. So we almost wanna have a principled way of kind of saying that's a good feature and that's not a good feature. Yeah. Which,[7]

Max Bartolo 00:33:25

you know, in these high dimensional spaces is just very hard to do. But you can you can control the input and the output spaces, and I think that's yeah. That that's where so so, you know, there's there's the general sense of adversarial examples where the the adversarial noise is generated by a function. Think of something like, you know, injecting Gaussian noise, which is undetectable into an image or switching a a word for a synonym in in language. And you'd argue that, well, the model's output shouldn't change because nothing has changed in terms of the the semantics of what the the query requires or means. And it's it's often the case, I think, that you see the model's response

[7] Adversarial Examples Are Not Bugs, They Are Features — paper

to that is to basically learn to cope with the function that generated the adversarial noise. And so it's not really learning to deal with real world noise. It's just you have a function attempting to simulate real world noise, which tends to be simpler and easier to to learn and counteract. So that's why I'm very interested in human informed adversarial examples. So this this general idea of have a human and the model interact, have the human, you know, probe the model, try to identify weaknesses in the model, failure modes, things that the model either consistently or not so consistently gets wrong and use use that as, you know, seed or like inspiration for training data to then train the model on it and make make the model more robust to, let's call it a more hopefully complex and representative kind of shape of this this noise distribution. And so that's what that's what adversarial QA tried to do. So this work from 2019 in this paper called BDAI, which was my kind of first PhD paper. And the idea wasn't wasn't necessarily novel from the perspective of getting models and and humans to interact. It's there had been quite quite a bit of work previously, so drop again, this dataset we mentioned earlier, had kind of a small component of the way it was constructed, be adversarial in the sense. There was work on on the quiz bowl task. So quite a few people have tried have tried this in the past. And what we really wanted to look at was what are the different effects that doing data collection in this way and in this way only have on, you know, what models learned, how rich the representations they learned, how robust those representations are. And, yeah, we found we found some extremely interesting findings. We found, for example, that the diversity of of the questions that you collect when having a model in the loop are just a lot more diverse, a lot more complex, generally a lot more interesting. You tend to see the the quality of the data you collect go down as well just because you, start to see more ambiguity. You start to see, you know, more, implicit information, that kind of a human would easily understand, but which is may may may be kind of slightly under defined. And it's just in reality a lot more representative of real world interaction where, you know, not every question is is perfectly defined or or perfectly specified. And yeah. So so we we had kind of 3 models which were state of the art back then. We had BERT and ROBERTA, and we also had BIDAF, which is a kind of earlier question answering model, which was state of the art back in 20 17, 18 probably. And what we found was that even even adversarial examples collected against relatively weaker models were very beneficial for training stronger models. And so it almost felt like there was this shift in distribution where you had the non adversarial setting where think of it as there's a paragraph and an annotator needs to find an answer in that paragraph and ask a question for which the answer is is kind of that span extracted from the paragraph. And in that setting, you you get very let's call them bland questions. Right? Relatively easy things. Like, say, you know, say the paragraph is talking about the sky. Would you would you use

kind of a consistent Yeah. Consistent text throughout. And so this paragraph is talking about the sky and how it's blue and all of this. And the question is, what color is the sky? And you have this almost direct mapping, even in terms of lexical overlap, to the paragraph to where the answer is located in the paragraph. And to annotate the answer will be blue. So then the setting is, now you've got a model in the loop. Now the model is going to attempt to answer the question. So you're going to ask the model, what color is the sky? And the model is going to say blue. And so you haven't located an adversarial example. You found an example that's very easy for the model. So what you're really trying to do is maximize the signal of the data you're collecting. And now you have to tweak your question. And you have to ask something like, what is the color of the thing in which the clouds are? And maybe it's a bit more complex, a bit more involved, and maybe now the model finds another color in the paragraph and says green, and it gets it wrong. And so you've managed to to trick the model. And through this process, you just get a lot more interesting data, both from a training data perspective. So we saw, you know, pretty massive improvements in terms of when you train on this data, how just robust the performance of the models is. And I think that to me personally, at least, you know, at the time, there were quite a few startups in the in the question answering space. I I knew quite a few of them personally, And they would tell me, we've added your data set to our models training, and it was just way better in real world application. And they'd see improvements on the benchmarks as well, but they were particularly impressed by just how how big the effect was in in real world settings. And and I think that's what robustness gives you. It makes it makes these systems a lot more applicable to to, you know, settings we care about. You know, from from from some perspective, it's not a very complex task. Right? You have 1 paragraph. It's from Wikipedia. It's relatively simple language. It's very easy to understand. And then you have these kind of more complex questions that come about from, you know, humans interacting with the model and being creative in terms of how they probe it. And and here we have models that represent, you know, state of the art in in 2019, so 5 years ago. And and AdversarQA was recently used to probe and test the adversary robustness of of the Lama 3 family family of models. So in in that tech report, it's kind of referred to as DynaBench QA, and we'll get into kind of DynaBench later. DynaBench was kind of the follow-up to adversary QA. And, you know, while while LLMs today massively outperform models we had back then on this specific task, they're still quite far from human performance, which suggests that there's still, you know, there's still a way to go in terms of general robustness capabilities.

DynaBench and Dynamic Benchmarking Approaches

Tim Scarfe 00:41:54

Isn't it weird how we keep talking about this mixture of pattern matching and reasoning? I mean, the the example you gave was beautiful. You know, what is the color of the thing where the clouds are? I mean, yeah, you could reason your way through that, but there are so many other examples in English where we just make it up as we go along, you know. It's just like this it's like the language game. Yeah. We just make stuff up. And it's just it fascinates me that it it's it's a combination of of what you're saying. So we we just robustify, and we just we just kind of incorporate all of these different patterns. But we also wanna have systems that reason as well. You know, I I I'm sure

Max Bartolo 00:42:28

I could continue to come up with more complex and more intricate ways of asking the question such that the answer would remain unchanged in ways that aren't reflected in the model's training data. And sure, in this case, I'm I'm potentially intentionally trying to confuse you. But I think the question is, well, I shouldn't be able to if your representations are robust. Tell me about DynaBench. So DynaBench was kind of a follow-up project from adversary QA and adversarial NLI, which Yixin Ni and Dawakila and Adina Williams did. So Dawakila kind of, you know, was the mastermind of of this project. He kind of, you know, did all the heavy lifting to to get the funding in place, to kind of get the the support for building out this project. It's it was originally started at Facebook AI Research. So I interned there working on DynaBench in 2020, I want to say, early 20 20, I think. And then that that kind of carried on for roughly a year. A a kind of global pandemic happened in between as well, so it was an exciting time. So DynaBench is is is a platform. It's a research platform for kind of testing and probing the span of model in the loop adversarial data collection and particularly doing that in a in a dynamic fashion. So this idea of you start out with some dataset. Right? You train a model on it, and then you interact with that model, and you want to find where its weaknesses are. You want to probe it. You want to figure out what its failure modes are, and you collect data in doing that, which you can use as training data. You get the training data, and you retrain your model typically on kind of the original data plus your new adversarial data, makes the model a lot more robust. And what you often get as a as a byproduct is a model which for most of the types of, you know, complex adversarial prompts you were trying previously, now would would generally start getting those right. It would still typically fail on some set of those prompts which were generally more complex. And then it would challenge you as an adversarial annotator to now, you know, think beyond

the limits of this new model and start trying to probe it and break it in different ways. And you can just keep doing that over time. And what you're doing is effectively just building a more and more robust model to the point where it just becomes incredibly hard hard to fool. And so DynaBench has has kind of powered, I'd say, many many research projects in the space. There were projects like hate speech detection, sentiment analysis, Prism itself was run through through the DynaBench platform. There's work ongoing with with CommonCrawl to try and figure out how we can kind of improve the the quality of the crawls for for pretraining data. And Dynabench has kind of since and now sits with the ML Commons community. And so there's there's this working group, the the data centric machine learning research working group, which is responsible for kind of maintaining DynaBench and making sure that, you know, people in the space who want to contribute, who want to explore ways of making models more robust, ways of, you know, building better both training sets and evaluation sets, have have a space and have the the the resources and tools to do that. And the the other aspect of DynaBench, which is kind of the benchmarking piece, which is where it gets its name from, is this idea that it's not just the interaction with models that's dynamic, but also also our benchmarks. For quite a long time and a lot you probably still today, the the general way we've operated as a community has been, you know, people create benchmarks because there's some interesting phenomenon they want to test. And that benchmark is created. Once it gets created, you effectively have good heart's law. Right? You have you have a clear measure of what it is you want to optimize for, and then you've got incredibly smart, incredibly talented people in the community optimizing for that specific problem that the benchmark reflects. And, typically, you see progress, which just happens really, really rapidly on that specific benchmark. So so DALA has this plot in in the Dynabench paper where you just kind of see saturation of different benchmarks over time. And if I recall correctly, it was, I think, somewhere around 8 to 10 years for MNIST, for example, and then somewhere around, like, 1 and a half to 2 years for SQUAD, and it just gets shorter and shorter with each benchmark, primarily as a function of of the field growing and maturing. Right? There's just just more people working on the problems, which means progress happens generally faster. So 1 of the core kind of philosophies behind DynaBench was this idea that we want to ideally also benchmark dynamically and that our benchmarks should evolve and should should change over time so that we you know, we're really measuring the the current set of capabilities of models that we care about. And and I think that's very true for for humans as well. Right? You wouldn't give someone with a a PhD a grade school math exam and use that to figure out, well, you know, should this person perform surgeries or or something relatively important. And to some extent, we do some version of that with with LLMs. I've seen so many cases of people who have a very clear task in mind, a very clear understanding of how they want the model to behave in that

setting. They have, you know, the domain knowledge, the expertise to to validate performance on that task really well, and then that's like which model they want to use because it has a higher MMU score or because it ranks higher in chat bot arena. And I don't know that that's the best way to ensure we're we're really evaluating the the full breadth of what we care about in terms of model capability. And I think 1 of the best ways to do that is to build benchmarks that reflect, you know, the the task, the application that you care about. And ideally, if you can do that dynamically such that, you know, today's model is at 70% as technology continues to advance, it gets to a 100%. That probably doesn't suggest that the model can solve every possible kind of situation or setting that you're putting it in, but rather suggests that your benchmark was limited, which, you know, it will always be because it's conditioned on some understanding of existing model capabilities. And so you just want to make sure that your eval also[8]

Tim Scarfe 00:50:01

update themselves

Benchmarking Challenges and Data-Centric Evaluation

Tim Scarfe 00:50:02

over time. Yeah. Benchmarking is an absolute nightmare. I mean, the great thing about benchmarks is they're standardized, you know, so we can compare apples to apples. I was speaking with Sarah Hooker in in the summer and she was advocating, you know, instead of having an absolute flops limit, for example, which could get good hearted straight away. She was suggesting having a basket or index or dynamic benchmark or or something like that. How how can we design new types of benchmarks that are somewhat impervious to Goodhart's law? I think we[9]

Max Bartolo 00:50:33

are almost caught in a local benchmarking optimum where we've we've just done things in a certain way for many years, and it almost seems like the the naturally right way to do. And and and I think we we potentially as as a community, right, we should probably think about taking a step back and thinking about given where the technology is today, given what language models can do today,

[8] DynaBench: Rethinking Benchmarking in NLP — paper
[9] Sara Hooker on Compute Limitations — paper

what is the right way to think about evaluating them? And I think examination in in people, for example, in in our education systems has evolved over many hundreds of years and is arguably imperfect. But I think the fact that we target specific capabilities almost in this, you know, hierarchical taxonomy where you've got sets of skills and you've got some concept of how those skills combine together to, like, you know, generate more complex skills. I'll give a very simple example. If you train to be a surgeon, you kind of go to medical school and you start off at some level with the basic set of skills that all humans require and then you kind of need to specialize and you go into probably chemistry and biology, and you need to understand the core concepts there, understand, you know, how to apply certain functions or reasoning patterns. And and and the the further you progress, the more complex and specialized your your exams become. And potentially then, the way to evaluate models designed to do everything, assuming we want to build general models designed to do everything, is to come up with a set of examinations which are intentionally designed to test models. And they might be inspired by how we test humans, they might not because models and humans have different characteristics. And I think we're starting to see some some very early work in this space. And and 1 thing I think we should also be thinking about in that kind of framework is is just ensuring again that we have a way to guarantee that this process is dynamic because the technology will keep improving and the moment that the benchmarks don't keep up is really the point at which we kind of need to either rethink everything from scratch or maybe, you know, we we just reached the point that the models are better than an expert human across all domains.

Tim Scarfe 00:53:19

Can you tell us about data perf? Data perf is

Max Bartolo 00:53:23

a set of challenges also run through the DynaBench platform targeted at being very, you know, data centric. So this this idea of data centricity focuses on the fact that a lot of what we do depends on the data that we're training these models on. And there's definitely kind of been waves of, I guess, community perception around, you know, what contributes most. Is it is it the data? Is it the models? Is it the architecture? Is it the the the resources? Is it the hardware? And I I think data perf came came around at a time where it definitely felt like there was a lot of focus on on the algorithms and less focus on the data. And, yeah, that's the main main motivation is is ensuring that, you know, the data is still central to everything we do and that the improvements to the data remain a core part of kind of this journey we're all on of building better,[10]

Tim Scarfe 00:54:33

more capable general models. So you're working at Cohere. Tell us about what you do there. I work on post training.

Max Bartolo 00:54:41

So I joined Cohere roughly 2 years ago at a point where the kind of language modeling landscape was a lot more raw. I'd say a lot less established. So this was kind of a few weeks before ChatGPT launched. And, obviously, since then, there's been a massive influx of kind of interest and excitement and investment as well. So at Cohera, I I built out the post training team. So when I joined, obviously, we had already very,

Cohere Command A Development Process

Max Bartolo 00:55:15

very solid base models. These are kind of pre trained models. They're they're trained on, you know, lots and lots of text, many trillions of tokens, and trained to generate what the next word is. And and back then, it was very clear that these models were very capable but didn't really follow instructions very well. And 1 1 kind of step towards making these models a lot more useful is getting them follow instructions from a human and getting them to do what the human wants them to do. And so, like, 1 1 example of this is if you asked a a base model a question, something like, what color is the sky? A base model would reply with something like, what color is the grass? Yeah. What color is the sun? Mostly because,

[10] DataPerf: Benchmarks for Data-Centric AI — paper

you know, the the context in which it's seen questions, in many cases, was was just lists of lists of questions. Like, even if it's pages of FAQs kind of with the the collapsible sections where those might not have been parsed in the right way. And and that's interesting behavior, sure, but it's not what you expect or definitely not what we've come to expect from these type of models. And so so 1 1 thing I did very early along with Asir who kind of leads pretraining is we we build a very kind of rudimentary interface for collecting some instruction following data, and we ask people internally, listen. Can you please provide some questions and type out what the response to those those questions should be? So this idea of having a prompt and and a completion. You know, that that's still today a big part of how data for these models in the in the supervised fine tuning stage is created. You have these these groups of annotators who are either writing their own prompts or having those prompts kind of seeded or synthetically generated or, you know, prompt sampled in different ways. And then they have to write out a completion, which has to be extremely high quality and kind of to the the standards of your style guide and to the standard of how you want your model to to behave. And so we we we did that. We ran kind of an internal competition. The the kind of interest and and, like, participation within the company was was insane. We we we thought we were being extremely ambitious targeting 10,000 examples in, I think, under 2 weeks, something like that. And and, yeah, we we obviously exceeded those expectations, and we trained kind of the first, you know, first generation of command models that that, you know, really followed instructions well. And and then kind of thing things grew from there. So we we then moved on to training a new model every week. So that initially was kind of the the instruction following supervised fine tuning stage where we we then started working with our own internal annotators with external data vendors, getting all sources of interesting, yeah, interesting instruction following data. We started synthetically generating data, and we did that for pretty much a year. So every week for something like 52 weeks, we delivered a new model. Back then, they were the common nightlies. I don't know if you're following back I with them. But, yeah. Received a lot of interest and excitement in the developer community in particular because people are always, you know, waiting to see what's what's the new model. And, you know, there were a few weeks like, most of the weeks we just, you know, had a model which was superior to the previous model as is is kind of natural with these processes. There were, I think, maybe 4 or 5 weeks where, you know, the the model was the same or slightly worse than previous models based on our internal metrics. And and so we wouldn't, release that model. And then you get people asking, oh, I didn't know. Based on what my interactions with the model, it doesn't seem like the model has changed. Are you still planning to release the model? And so it was just a very inspiring time for us. And then around you know, towards the end of of last year, we kind of organized a bigger kind of structured push towards a

new generation of models, which became command r and r plus, which at the time of release, so in April, were kind of, you know, ranked extremely highly on pretty much all all the metrics. I think we were the fourth biggest like, the fourth ranked provider on on Chatbot Arena. So I think there was Anthropic, OpenAI, and and Google, and then Cohere. And and now, of course, we're working on the next generation of models.

Model Quantization and Performance Evaluation

Max Bartolo 01:00:26

What do you think about quantization?

Tim Scarfe 01:00:28

You know, people kind of like ripping your model to shreds and reducing the precision and hacking and stuff like that. It's very useful

Max Bartolo 01:00:37

as a short term solution from the perspective of efficiency gains. I think it's not always clear with quantization. The problem always comes back to evals. So the objective for most quantization efforts is make the model a lot more efficient and maintain performance on the evals from the perspective of, you know, maybe you accept some, you know, some small drop in performance for massive efficiency gains. And it's it's it's very effective from the perspective of, you know, if I have a 100 b model or a 35 b model and I want to quantize the 100 model, the 100 b model to kind of the same level of just like per token cost as the 35 b model, I'm probably going to get a much better performing model at the same the same price range, but then I can also quantize the 35 b model and reduce the cost there. I think I think 1 of the the trade offs is is is these blind spots where you might not be measuring everything. Like, we've kind of discussed, our evals are incomplete. They don't measure things like, in many cases, advanced complex reasoning. In many cases, they're quite limited in terms of their ability to test, say, you know, very long range interdependencies and long context performance and and these kind of things. And and so there's always a risk that you're potentially degrading performance in aspects that people care about that you might not have been able to measure

Tim Scarfe 01:02:21

for for whatever reason. Is reasoning the the synthesis or the execution? And what I mean by that is I can teach you how to do multiplication or long division or something like that, and you're just going through the mechanics of executing the rules. We make lots of mistakes, as you say, but we have the ability to synthesize the rules and kind of describe them algebraically. I would expect that the synthesis is a requirement,

Max Bartolo 01:02:49

but that you can really test the execution. And there's various mechanistic interpretability efforts trying to look into, you know, how are models actually thinking what's going on under the hood. The the recent kind of work in terms of scaling test time compute where models are are generating these reasoning chains and then providing a final answer. And in many cases, giving better performance, I think that's that's still heavily underexplored. We don't necessarily know exactly what the effects are that are resulting in the better performance. And I think there's there's various early indication. We we have some work going on as well that there's no strict requirement for models to to reason in natural language or to have these these kind of internal thinking chains which are explicitly generated. And I think if you want to test whether a system reasons, I think the adversarial setting is 1 which allows you to really probe for, you know, specific capability, specific type of reasoning. We saw this in a lot of the early adversarial QA, BDAI style work, where limitations of a particular benchmark are the benchmark itself. And so you've curated a benchmark. You've gotten crowd workers to ask lots of interesting complex questions, and then you you kind of check how well the model performs. And typically, performance saturates very quickly, and, you know, models do very well. Drop, for example, is a a pretty classic question answering benchmark. It was very popular. It's still very popular for evaluating language models. DROP, if I recall correctly, stands for discrete reasoning over paragraphs. And it's really focused on numerical reasoning. The limitation there is that if the model, you know, achieves human level performance on the benchmark, can you claim that the model can reason? And I think it requires 1 step further, which is if you then get that model, put it in front of humans, ask humans, listen, probe for this capability, probe for this type of reasoning, see if you can get the model to fail. If if humans are, you know, consistently successful at getting the model to fail on examples that, you know, are slightly different from, what has been trained on, what has been tested on, then I think it's fair to assume that 1 isn't really reasoning. It's just, you know, figured out how to do it on a specific

benchmark.[11]

Reasoning Capabilities and Training Progression

Max Bartolo 01:05:18

How puritanical should we be? You know, because you you could say as you just did that reasoning is a binary. Like, you you either got it or you didn't.

Tim Scarfe 01:05:26

Or you could take the stance that if it performs robustly a lot of the time, even with a few edge cases, then we could still say it's it's reasoning.

Max Bartolo 01:05:35

I think we have to hold to a very high standard. And the reason is that I don't expect the interaction between humans and machines to be the same as the interaction between humans and humans. You know, we we build up trust with each other. We engage in conversation, many cases involving complex reasoning in specific domains, and and we really understand what our own limitations are. And I think the the expectation of machines is consistently they're right all the time. It's it's a function of traditional software engineering and how how that's approached. If you had a calculator and you put in 1 plus 2 and the calculator said 3 and then you try 2 plus 7 and the calculator said 17, you'd you'd throw away your calculator. You wouldn't you wouldn't try any other questions. And, you know, LLMs in the early days, even these very early, you know, very simple mathematical questions, they they would they would get wrong. Yeah. So I think there there is a sense of the standard is actually higher. We need that consistency and we need that robustness. I can definitely imagine that as the complexity of the task increases, there might be a bit more room for error. But definitely on more simple tasks, the tasks we're using today to evaluate these models are relatively simple. 1 of the very common benchmarks is is GSM 8 k, which is grade school math, which is extremely easy for for humans or at least humans with a certain level of proficiency. I mean, current models do do do very well. And I'd probably argue, right, you're typically looking at like 97, 98% level levels of performance. You're probably there looking at noise in the test set rather than anything about the model's capability specifically. But, yeah, going back to the earlier point, the next step is really, you know, put put that model in front of a human, get them to probe grade school math levels and types of reasoning, and, you know, convince

[11] DROP: A Reading Comprehension Benchmark — paper

a human that the model is actually consistently doing the[12]

Tim Scarfe 01:07:44

what is being asked to do. So you know like with these deep learning models, they have characteristics of reasoning and they have characteristics of kind of statistical matching. And they seem to do both in 1, which is brilliant because we can align them with reasoning algorithms, but we can also get them to make good guesses in other situations. But, you sketched out this example that sometimes it would just be like a hash table when it would just go and retrieve the fact from the document and sometimes it would do reasoning. But the boundary is kind of clear, isn't it? So sometimes it's directed, and sometimes it's, you know, it's quite spread out and it's quite diffused. How does that process work, and how epistemically aligned is it? I suspect analogous to how humans learn, where we kind of learn

Max Bartolo 01:08:29

what I guess I'd call relatively simple tasks first or learn, you know, to understand and apply relatively simple functions first, and then over time assemble those into my kind of more complex composite functions where you might apply a specific type of reasoning or a specific function at at a for a particular problem or might apply a different approach for a different problem, I suspect that's that's similar to what what we are starting to see and what we probably see more of in these deep learning models. It's inefficient to reason if you don't need to. Right? If you if a model is asked, what is 1 plus 1? And it has seen that, you know, hundreds or thousands of times in pre training, it doesn't need to reason from first principles and figure out, you know, the complexities of math and how to do basic addition. But that doesn't take anything away from solving that problem as long as the solution is correct. What do you think about the ARC challenge? I think it's another step in the right direction. Right? We we need to keep coming up with challenges that really push the limits of current models. We'll we'll we'll solve that, and then we'll move on to the next 1. And hopefully, some point, I I think it becomes a lot more relevant to ground, these kind of challenges in real world application and in terms of what's useful for us. What by design do we want AI systems to do? How do we want them to operate in society? How do we want them to interact with humans? And I think those questions are becoming more important. And and hopefully, at a point in the not too distant future, you know, these would be the questions we care about, which is how can I get the existing technology and apply it to,[13]

[12] GSM8k — paper
[13] ARC-AGI Challenge — paper

Tim Scarfe 01:10:22

you know, to the benefit of humanity? Will connectionism get us all the way? You know, do do you think we can come up with the right types of neural networks that will do, you know, reliable, robust reasoning? Or do you think we need to have some kind of hybrid architecture?

Max Bartolo 01:10:38

I don't really care in the sense that I see no immediate reason to expect that there are, you know, severe architectural bottlenecks to what we're doing. I mean, there there are some, but they're they're easily overcome. Right? So so 1 example is is tokenization, and we had this this paper recently at the MLLP with with Sander. It won an outstanding paper award. It got retweeted by Andre Carpathi, which made my socials explode, and I've never seen a spike like that in terms of kind of the usage stats. But this this work focuses on on this concept of glitch tokens, which is this idea that certain tokens cause weird behaviors in, deep learning models. You know, there's various reasons for this. The main 1 is you have this, mismatch in terms of the the data that a tokenizer is fit to and, the data that a model is then subsequently trained on. What ends up happening is you have tokens which are just very rarely seen or very or pretty much, like, not seen at all during training. In many cases, you you have weight decay, which means that the the embeddings of these tokens kind of tends to 0 as model training goes on. And even if it doesn't tend to 0, it just tends to a very kind of small magnitude where it becomes very hard to kind of separate like token specific information and it becomes very easy for a model to get confused about, you know, a token's identity. You know, this kind of effect highlights an immediate limitation in terms of not necessarily the the architecture, but the the current way of doing things. But I don't think that's that's a question of, you know, connectionism versus the more, I guess, traditional symbolic structured way of way of approaching things. More more just pointing out a a limitation of the way we currently do things. I don't see indication today that, you know, deeper, larger, bigger models won't be able to do more impressive things. And once we reach the point that they can't, I think then we'll then we'll know the answer, and then we can kind of take it from there. But in the meantime, you know, I think I think we've made incredible progress over the past few years, both from a data perspective, but also you know, massively driven by just the efficiency and the the parallelism that transformer provides. And I'm I'm sure we'll see more innovations in the future. I'm sure that, you know, further efficiency gains will will accelerate things further. It it it remains pretty pretty clear that an LLM's data efficiency is not quite the same as a human's. LLMC, you know, many lifelines of human data during training that that we do, but they are fortune. Right? If there's a compute and there's there's the resources, we are time constrained and

Tim Scarfe 01:13:46

AI systems aren't.

Context Windows and Enterprise Applications

Tim Scarfe 01:13:48

Yeah. I I suppose there is a similarity there in the sense that the kind of information we process and build on is also multi generational. Like our language and our knowledge just seems to continuously grow even though we die every 70 years or so. I think that's very true and I think there's

Max Bartolo 01:14:06

there's definitely some compression of knowledge that happens which makes passing information onto the next generation more efficient. Software has the advantage. The the the kind of space of continual pre training is is just starting to become relatively explored. But currently, in most cases, and mostly due to architectural constraints, right, if you want to increase the size of your model, it becomes nontrivial to figure out what's the right initialization based on the knowledge I currently have. But if you view what LLMs are doing as an efficient compression of the data, then starting from an existing efficient compression seems a lot more efficient than starting from scratch every time. And that's a lot more convenient to do with LLMs than it is with humans. You can, you know, copy the weights of 1 model, pass them onto another model. You have an instantiation of exactly the same kind of knowledge and information and capability

Tim Scarfe 01:15:02

in a way that you can't you can't do with humans. How important is having a big context window for you folks? We're starting to see some models with really big context now. Is that something that's quite easy to add on or is is that fraught with problems? It's generally quite challenging

Max Bartolo 01:15:20

to to maintain high performance across the kind of span of the context window. And I think it's you know, going back around 12 months, there was definitely what felt like this this race, right, to go larger and larger in terms of context window. Our our kind of first generation of models had had 4 k context windows. Our current models have a 128 k context windows. And, you know, you saw pretty much all of the LLM providers really start to to push this idea of longer

context and and market that. Google in particular, I think, made some very interesting innovations in that space, you know, going towards context windows that were previously considered quite challenging to achieve and still maintaining good performance. And I think it's a bit of a trade off in that the vast majority of user queries are not long context, but long context is useful for many kind of things you want to build on top of language models. So things like retrieval augmented generation. The you know, that's sensitive to how good your your retrieval is. So Cohere has extremely competitive kind of embedding models, which are very helpful there, and re rank models as well. And then there's a question of, well, how much of what I've retrieved and ranked can I put into the model as context in a way that the model can use that information and, you know, aggregate information across documents in in ways that are are beneficial? Then once you get to things like tool use, once you get to things like, you know, incredibly long conversations, right, there's these ideas of effectively infinite conversations with the model where the the conversation to some extent is the state. Right? We were talking about personalization earlier. If your entire conversation with the model is representative of, you know, the way you want to interact with the model including feedback, then potentially that's that's kind of an area which is is is ripe for exploration. When it comes to working with code, there's massive advantages to having long context capabilities where you want to do something like give a model your entire code base and have it process information, extract information, potentially rewrite things. There's a lot of value you can unlock there. And I think it's it's it's a bit of a chicken and egg problem. Right? Models traditionally weren't you know, had relatively short contexts. I mean, going back, I think BERT had 5 12 token context, and most people thought that was plenty. And now we're, you know, in the in the hundreds of thousands, if not millions. But but I guess there's a question of at what point does it make sense to continue to feed things into into the context? It it becomes more inefficient. Right? You can do some clever things with caching and you can you can speed things up, but there's just generally more processing that needs to happen for generating the same answer. And if that answer is better than what it otherwise would have been, then I think that's that's fair game. If you get to the point where, you know, kind of adding information to the context for no good reason, then I I'm not I I don't really hold any judgment. I'm just not sure there's much sense in doing that. So I I remember a few months back there was this idea that you don't need to do retrieval augmented generation with infinite context models because you can just put in as input all of the data that exists, which is very true. You could. But do you want to? Like, what what do you gain from that? And it almost feels intuitively a bit wasteful. It's like, again, if I asked you something like, what is the color of the sky? But I told you, also, you need to read the entire Internet to be able to answer that question. In many cases, you don't. What do you think about reasoning models? I mean, of course, it's definitely

something of of interest. Right? We we we generally want models or or work towards developing models that are kind of best in class across all dimensions. And and we know that general reasoning capabilities make make models a lot more both, you know, they perform a lot better on reasoning style benchmarks, which in itself has value, but they also are more useful and more valuable for people because you can do more complex things with them. I quite like the idea of being able to trade off test time compute with performance. So that's that's 1 thing that my team is currently working on where where but we're we're looking at a bit a bit more broadly. It's not just it's not just allowing the model to generate more tokens, you know, to to optimize some some performance number. It's more around thinking about how are these models interacting with users. What is the signal and information that you can share with users, maybe even as you're going through this reasoning process? And how, again, do we make that controllable and customizable? You might have a, relatively simple noncritical task where, say, if models were had had well calibrated confidence scores, you could tell the model something like, if you're at least 60 confident, give me the answer. And as long as that, again, you had that kind of core requirement of of of reasonable calibration, then which basically means that if you ran the thing a 100 times and the model should get the answer right 60% of the time if it's 60% confident. And that, you know, might be fine in say, creative writing settings or it might be maybe what you want in particular settings. Whereas if you had really, you know, really complex, really critical task, you might be able to say something like, I don't afford for you to get this wrong. Just like think as much as you absolutely need to. If you're in any doubt that the answer is wrong, tell me. But if you're going to output an answer, make sure it's right. Obviously, we're far from any model that can do that. But that I think that kind of behavior, kind of thinking about the problem from almost a user interaction perspective, I think, yeah, it's going to be extremely interesting over the coming years.

Tim Scarfe 01:22:47

I agree. And, you know, the the big challenge is just building the apps, imagining what can be done with this technology. Just the basic stuff like doing enterprise search really well. We've got a long way to go, but it's gonna be an exciting journey. For sure. Max, thank you so much for joining us today. It's been amazing. Thank you. Likewise. Thanks for having me.

References

[1] **Max Bartolo Website**
https://www.maxbartolo.com/
person

[2] **Cohere**
https://cohere.com/command
company

[3] **Procedural Knowledge in Pretraining Drives Reasoning in LLMs**
https://cohere.com/research/papers/procedural-knowledge-in-pretraining-drives-rea
paper

[4] **Influence Functions in Machine Learning**
https://arxiv.org/abs/1703.04730
paper

[5] **Studying Large Language Model Generalization with Influence Functions**
https://arxiv.org/abs/2308.03296
paper

[6] **Command A Model**
https://huggingface.co/CohereForAI/c4ai-command-a-03-2025
company

[7] **Human Feedback is not Gold Standard**
https://arxiv.org/abs/2309.16349
paper

[8] **The PRISM Alignment Dataset**
https://arxiv.org/abs/2404.16019
paper

[9] **Adversarial Examples Are Not Bugs, They Are Features**
https://arxiv.org/abs/1905.02175
paper

[10] **DynaBench: Rethinking Benchmarking in NLP**
https://aclanthology.org/2021.naacl-main.324.pdf
paper

[11] **Sara Hooker on Compute Limitations**
https://arxiv.org/html/2407.05694v1
paper

[12] **DataPerf: Benchmarks for Data-Centric AI**
https://arxiv.org/abs/2207.10062

paper

[13] **DROP: A Reading Comprehension Benchmark**
https://arxiv.org/abs/1903.00161
paper

[14] **GSM8k**
https://paperswithcode.com/sota/arithmetic-reasoning-on-gsm8k
paper

[15] **ARC-AGI Challenge**
https://github.com/fchollet/ARC-AGI
paper