Col 30 just convo

Fri, 06/21 11:47AM · 35mins

AI Notes

Transcript

Jon Gallagher 00:01
Okay, welcome back. So we are on the second episode of three that we promised talking about generative AI. The first one talked about the recent history and all history of generative AI is pretty recent of what's been released, where it's going.

Jon Gallagher 00:17
And we kind of left off at the end of that talking about the fact that we're really feeling like it's commoditizing. Now a certain period of time has passed since that episode and therefore things have changed.

Jon Gallagher 00:28
The acceleration here is just incredible. So we're going to start off talking about what's changed, whether our thesis of commoditization is still valid or invalid, and then talk a little bit more about the challenges of implementing generative AI and whether it's right for you or not and where the guardrails are.

Jon Gallagher 00:48
So hopefully you're listening to this in terms of guidance and in terms of taking advantage of this technology. So as usual, I'm John Gallagher.

Logan Gallagher 00:56
And I'm Logan Gallagher.

Jon Gallagher 00:57
And let's get started by Logan tell us what's changed and about the generative AI landscape and where we are with that idea of commoditization.

Logan Gallagher 01:07
Certainly. So since we last recorded and released that previous episode, there have been a few more models that have come onto the scene. Meta, the parent company of Facebook, released their Llama 3 model, which is even larger and more performant than their Llama 2.

Logan Gallagher 01:25
And also a startup that has received a lot of funding from Google and Amazon and others called Anthropic has released their Claude 3 family of models comprised of Obis, Sonnet, and Haiku. Three different models of three different sizes in that family.

Logan Gallagher 01:44
But I think that our thesis of commoditization of models is certainly continuing to prove true. We're seeing that across these models, the Google models like Gemini, the OpenAI models like GPT -4, and these other models like the Claude family of models and Llama 3 are all have very similar levels of performance.

Logan Gallagher 02:09
You can pick one benchmark over another and see some differences. But if we're taking a broader view of all of the models that are out there in the ecosystem right now, they all have relatively similar levels of performance.

Logan Gallagher 02:24
And I think we are really approaching the idea of commoditization of these models that they could be swappable. You could drop in GPT -4 into your application and swap it out with Gemini or with one of the Claude models and get similar to equivalent performance with these different models.

Logan Gallagher 02:49
And that opens up some interesting questions because these organizations are investing an enormous amount of money into these models to train an individual large language model takes tens of millions, sometimes even hundreds of millions of dollars to get to a point where they are all relatively replaceable with one another.

Logan Gallagher 03:10
And that does put organizations in an interesting position where organizations will have the ability to choose and maybe even have some negotiating power with the organizations that are building these models.

Jon Gallagher 03:24
Interestingly, when we look at commoditization, you're kind of implying, and I don't want to put words in your mouth, that the answers that you're getting out of these will be in the same basket. So if you're looking for something, and this is something I harp on all the time.

Jon Gallagher 03:43
If you are looking for something unique to target a specific need in your company, let's say you're looking for a chat bot that will replace or enhance your customer service representatives. So what we're saying is out of the box, none of these models will do other than by happenstance.

Jon Gallagher 04:04
None of these models will be something that totally fits your needs. We're going to circle back to that later. But when we talk about commoditization, we're talking about the fact that in general, the training data is what guides the knowledge, the insight of the model, and everybody's training essentially on the same data, which is the internet.

Jon Gallagher 04:25
Therefore, you should have other considerations for this, correct?

Logan Gallagher 04:28
Yeah, absolutely. And I do think that there are some other trends that are contributing to the potential commoditization of the models. Some of the frameworks that are emerging that can help you write software to interact with a large language model.

Logan Gallagher 04:44
Some of those frameworks are becoming open source. Open source is something we want to talk about this episode, but these open source frameworks could potentially make it even easier for you to swap out models in the backend and continue to use the same code that you have written.

Logan Gallagher 05:02
So that puts organizations that are wanting to use a large language model in an interesting position. They have myriad options. None of them are going to be a perfect fit, but they do potentially have some negotiating power and some ability to play these models off of one another, not necessarily have to get locked into one model or one ecosystem.

Jon Gallagher 05:28
And this reminds me of a point we talked about in the previous episode of the history of relational databases, relational databases, the process of storing, retrieving, manipulating data got to be pretty cut dried really quickly.

Jon Gallagher 05:41
And therefore, the breakout product in the in this is in the early 80s, early to mid 80s, the breakout product became Oracle, because of the infrastructure, the environment, the tools that were developed to enhance business processes.

Jon Gallagher 05:57
As a result, Sybase, Informix, all had to go different paths because they didn't really choose to build the infrastructure or weren't as successful to build the infrastructure that implemented it in business.

Logan Gallagher 06:10
Yeah, I think that relational databases can be really instructive for the trends we might see with generative AI. I mean, generative AI is not a database strictly, but you query it with a prompt to ask it some questions and it returns some

information back to you.

Logan Gallagher 06:28
That information is not held as records and tables, like in a database. It's held as weights and parameters inside of this model resource, but the workflow of interacting with a model is not too different from a database.

Logan Gallagher 06:42
So an evaluate offers is not too different from a database. So looking back to the relational database wars could be pretty helpful, I think.

Jon Gallagher 06:51
And the ultimate reward for companies is to structure their systems so they're not tool -dependent. So at Mechanic, the personal choice for Mechanic might be a snap -on tool versus any other tool. But there isn't anything about a crescent wrench or a box wrench that is going to be different.

Jon Gallagher 07:14
It's the quality of the system. And using snap -on as an example, the infrastructure around it that the snap -on representative is going to be there to provide your professional tools on site. These are the sorts of things that other businesses have had to build or use to differentiate in the marketplace.

Jon Gallagher 07:31
And I think technology is going to be absolutely no different.

Logan Gallagher 07:34
Yeah, and one of the ways that I think we're already seeing that is organizations are realizing that a large language model out of the box is not necessarily going to meet their use case. This large language model was trained on oftentimes public data or other large data sets that may not be useful enough for a specific organization.

Logan Gallagher 07:59
A specific organization is going to want the generative AI model to be able to use its own data, the proprietary data owned by the organization, data about their customers, data about their internal inventory, data about their own business processes.

Logan Gallagher 08:16
And companies are realizing that while they don't have the resources to invest tens of millions or hundreds of millions of dollars into building their own generative AI model, and they'll need to rely on the Googles and open AIs for that, for that base model, that base model in and of itself is not necessarily going to be useful for them.

Logan Gallagher 08:37
And they're going to need to take some steps to make it more useful. We have seen two patterns really emerge and become quite prominent in the last several months. The first one being model fine tuning, where you make a generative AI model more performant at your specific tasks.

Logan Gallagher 08:57
Let's say that you are wanting to have the model be able to label help tickets as they're coming into your company. And you want those help tickets to be routed to the correct department in your organization.

Logan Gallagher 09:10
This help ticket should go to IT, this one should go to some other department, the model out of the box is not going to have knowledge of your particular departments. So you will need to do something like provide your own model tuning data set where you show examples of here's a help ticket.

Logan Gallagher 09:28
And this one is labeled as IT. Here's another help ticket. And this one is labeled as account services. And this one is labeled for getting help with your entry badge or whatever it may be. And you provide that tuning data set to the model and run a fine tuning job in order to create a custom model or some custom layers of a model that will be better at performing your task.

Logan Gallagher 09:53

A model is kind of a Swiss Army knife, it's very good at many things, but may not be particularly good at your task. So you will have to take the steps of tuning to make it better at doing your task.

Jon Gallagher 10:06

That's gonna be a challenge because in many ways that's not how these large language models, generative AI are being sold. That they are being presented to the marketplace as answers in and of themselves.

Jon Gallagher 10:20

And as you've said before, these are Swiss Army knives. They are, Swiss Army knives, I carry one. I love Swiss Army knives, but I'm not gonna build a house out of that because it's not suited to do that.

Jon Gallagher 10:32

But if I'm needing to whittle up some shavings to build a fire, it's perfect for it. So non -scalable stuff, things that aren't intimate to our business process, it's not gonna be able to do without us providing data about our operations.

Jon Gallagher 10:47

Now, here's the other thing is that if you're effectively going to use generative AI, you have to know your business. This is, and it's trite to say it, but IT systems garbage in, garbage out. If there is no data, then nothing comes out.

Jon Gallagher 11:04

If you have bad data, then bad things come out. So there is this first step that you have to understand your operations and how you want your operations to run. And you have to have data that reflects a correctly running operation and hopefully some data that reflects incorrectly so that the model can base its decisions on data from your operations.

Jon Gallagher 11:25

So there's a lot of legwork. There is a lot of stuff, a lot of expert subject matter expertise that has to go into the front end of this before you can even start thinking about, wow, I think Claude is, Claude's sonnet is better than Gemini chatbot.

Jon Gallagher 11:41

Unless you've got a pile of data to show me about how your operations run, that's just pure late night bull sessions. Yep, certainly.

Logan Gallagher 11:51

The other trend that we are seeing become increasingly popular is maybe you're not trying to train the model to be better at doing something in your organization. But you would like the model to have knowledge of some data that is proprietary in your organization or some knowledge base or data set that the model was not initially trained on.

Logan Gallagher 12:17

And you can do this, you can provide these supplemental knowledge bases to the model in a pattern known as grounding, also known as retrieval augmented generation or RAG. But with this pattern, whereas model fine tuning helps a model get better at doing something, grounding helps a model know more about something.

Logan Gallagher 12:41

You can specify some database or some collection of documents that you have internally in your organization as a grounding source. And you're telling the model, when I ask you a question about something pertaining to all of the marketing material PDFs that we have provided as a grounding source, I want you to go into that knowledge base and use that knowledge base to inform your answer to me when you provide your response.

Logan Gallagher 13:10

So grounding has become another pattern that is becoming increasingly popular to supplement the knowledge of these models.

Jon Gallagher 13:20

This is our perspective with commoditization. You should be collecting the data about your operations so that you know what a known good and a not known and a bad known bad situation are so that you can feed these systems, either training them or providing them the grounding data.

Jon Gallagher 13:37

But you're not going to get away from collecting solid data. And that is an enormous task for most operators.

Logan Gallagher 13:44

Yeah, so the trends that we're seeing is potentially a commoditization of models where they may be swappable. But after that, a real focus on tuning or providing access to data to these models because while they may be swappable, none of them are production ready out of the box for your specific app and your specific organization.

Jon Gallagher 14:08

leads to some of our concerns about using these things. So there's more effort involved than the mark in your sales folks will tell you and that's pretty universal across any product but for doing this in your in your system you're going to have to be knowledgeable about your operations and your processes enough to be able to tell whether the data you're providing is correct and also when the outputs occur to detect whether you are getting the right kind of data.

Jon Gallagher 14:40

So let's talk about that. Let's talk about some of the tendencies that these large language models have to be vulnerable and the first one that we're kind of leading up to is the idea of hallucinations.

Logan Gallagher 14:52

We've brought up hallucinations before, but it remains a major concern because the fact of the matter is these models, when you ask it a question, you pass the prompt, it is not thinking per se, it has no consciousness, it's giving you an answer based on statistical probability.

Logan Gallagher 15:15

And that answer, based on statistical probability, can really be thought of as a best guess. It's a guess that is informed by an enormous amount of data, but at the end of the day it's still a best guess, and that guess may be factually incorrect.

Logan Gallagher 15:33

I know I personally use, I use one of the models when I'm writing code. I have a plugin in my code editor that's using a model called Copilot. Under the hood it's using GPT -4 from OpenAI, and sometimes when it gives me a suggestion for a line of code, it will give me a suggestion that looks plausible, but if I look at it a little more closely, I realize that one of the libraries it references, or one of the data types that it references, doesn't exist.

Logan Gallagher 16:08

It's made up. It looks real because the model was guessing what's a likely package, a likely library, that would be used in a particular use case of this line of code, but it's fictional. It's a fictional package.

Logan Gallagher 16:24

Fortunately, I know what I'm doing when I'm writing code, but I could easily see a novice developer or someone who's just learning really get wrapped around the axle trying to write code with a generative AI model assistant.

Jon Gallagher 16:40

And that's something is, as you look to implement large language, generative AI to solve business problems is something you're going to have to be aware of that the answers need, you will definitely need to check the first batch of answers and do it on a continuing basis, but you should probably also have random answers out of there to try and try and understand how many hallucinations you're getting and then deal with them.

Jon Gallagher 17:05

Revisit the training data, revisit grounding, but you're going to have to have enough statistical information for the model to come up with correct answers.

Logan Gallagher 17:17

Yep, if you're using a generative AI model in some application or other use case where factual accuracy and correctness is paramount, then you are going to need to build additional steps and additional layers in your process to do fact checking, to make sure that the responses that the model is generating are correct.

Logan Gallagher 17:36

If you are using this model to make any decisions that can impact human health or human safety or human financial well -

being, you're going to want to fact check those answers.

Jon Gallagher 17:47
and all you high schoolers out there using GPT -4 to generate your papers, check them. you

Logan Gallagher 17:52
Yeah, yeah, yeah. At least scan through him first before you turn him in. I think that Cory Doctorow, who we have used as really a guiding light when talking about this topic, because he's been thinking seriously about this topic and writing seriously about it for quite some time.

Logan Gallagher 18:09
I think that he provides a really perfect quote in a recent piece he wrote about hallucinating AIs, which I'm just going to go ahead and read because I don't think I can say better than Cory, where he's saying, hallucinating AI is a terrible copilot.

Logan Gallagher 18:26
It's just good enough to get the job done much of the time, but it also sneakily inserts booby traps that are statistically guaranteed to look as plausible as the good code. And that's what a next word guessing program does, guesses the statistically most likely next word.

Logan Gallagher 18:45
And that's been my experience. And it's funny that he uses the word copilot, because that is literally the product that I use to help me write code. And that really highlights the danger potentially of relying on these tools too much.

Logan Gallagher 18:58
I still do use that plugin to help me write code because it can help me write boilerplate code more quickly can help me save a step of going to a web page and grabbing the syntax of some object or some function.

Logan Gallagher 19:17
But beyond that, I know that I have to be vigilant while I'm writing code using that tool.

Jon Gallagher 19:22
As the member of this company who still uses VI and make, sometimes it's good to be out there banging on rocks and generating code.

Logan Gallagher 19:29
into my VIM, I'd say.

Jon Gallagher 19:33
but yeah. So hallucinations, we talked about them last episode and those of you who are over it, sorry to bore you through that. Let's talk a little bit more about security. And again, we touched on security in the previous episode but we wanna take this in two separate directions.

Jon Gallagher 19:50
One is the security of the engine itself, which is the things we're worried about, data leakage and definitely inappropriate data leakage and possible attack vectors within the engine.

Logan Gallagher 20:05
We were talking earlier about models being similar to databases. It's not a perfect analogy, but when you think of the workflow of using a model, you're asking it a question, and it's returning a response from the data it was trained on.

Logan Gallagher 20:19
When you ask it a question, it can provide you answers from any of the data it was trained on. And that presents a real potential security vulnerability. If that model was trained on any sensitive data, think about personally identifiable information like our names and addresses and phone numbers and personal health information or financial information that we don't want that information being leaked to some anonymous user who could leverage it towards bad ends.

Logan Gallagher 20:57
You have to be very careful about the data that the model has been provided. And you're going to have to take steps to make sure that any data that this model has been trained on or it's being grounded on has been de -identified.

Logan Gallagher 21:13

Any sensitive data has been scrubbed from the data set. And that takes a lot of pre -processing and initial work before you ever get the model up and running.

Jon Gallagher 21:25

I worry also, so data leakage obviously, but I worry about the things that you can do with correlation within the model itself. Let's say that you are looking to market a product to people who are suffering from end -stage renal disease, who are currently undergoing processes to replace their kidney functions.

Jon Gallagher 21:47

You could marry disparate databases to reveal this information. So where you take a database of people who live near one of these facilities or database of people who make journeys by these facilities and make purchases nearby.

Jon Gallagher 22:04

So there are things that we need to worry about as a society. We as a society want our health information to be private. We don't want our health information to be back -engineered into, well, you have end -stage renal disease and we are not going to provide you this product or we are not going to do this other service for you.

Jon Gallagher 22:24

So we need to be aware of the fact that these engines, again, they are not necessarily databases themselves, but have access to all this data and have the ability to make linkages across this data at a scale that human beings may not be able to.

Logan Gallagher 22:37

Yeah, and to continue using databases as a comparative tool. With a database, you set up users. And this is some identity with credentials to access the database. And users can have different permission levels.

Logan Gallagher 22:55

You can say that user Logan has permission to create whole new tables in the database and access any of the data across any of the tables. And you might have another user that has far fewer permissions, can only access certain records or certain tables.

Logan Gallagher 23:10

And you can define that permission structure for who has access to what. Unfortunately, with generative AI models, we really don't have a mechanism like that to control which users have access to what data that the model was trained on or the model is grounded on.

Logan Gallagher 23:29

We have no role -based access control mechanism or no entitlement management system that can say when user Logan submits a prompt to this model, he should only be provided to certain subset of the data because this other subset of the data, he doesn't have permission to access and he should not be made available to him.

Logan Gallagher 23:53

We don't have any mechanism like that when we're asking a question to a model. We have access to all of the data that it was trained on or that it's grounded on.

Jon Gallagher 24:04

I was just sitting here thinking of the challenge of designing something that has an internal awareness of what databases we would call views. You know, if you are the customer service rep who is working with ABC Corp, you have, you have access to ABC Corp records, but not DEF Corp, because that would be a privacy violation.

Jon Gallagher 24:24

You have no authorization to do that. That is easy to implement on a wide variety of databases. And that is the mindset that people approach these things with. But how are you going to do that based on the model retrieving data from its own training data, or maybe accessing databases where it doesn't have the ability to enforce what a view, right?

Jon Gallagher 24:47

So we, we have the, the security that exists for the generative AI LLM is literally accessing the tool itself and not really thinking about what the tool does and looking to constrain the tool. And that is going to be the next step because if you're going to apply to provide a medical device under HIPAA or a financial instrument under, under financial information, all of these things are going to require that there is that people who do not have direct authorized access to this data are denied access to this data and the, the systems have to provide that and don't look to be able to do that right now.

Logan Gallagher 25:26
Yeah, I would say for generative AI models to become successful, they will, especially in industries that have sensitive workloads like healthcare and finance, they will have to build this eventually.

Logan Gallagher 25:42
There will have to be some entitlement management capability that can restrict a user's access to a certain subset of the data that the model has available to it. This is going to have to happen if these models are ever to become successful in many organizations, but the model architecture itself really doesn't lend itself to it.

Logan Gallagher 26:06
It's not going to be easy to accomplish that.

Jon Gallagher 26:10
Another thing about security is the transparency of the underlying system. There is an old saying in computer security, there is no security in obscurity. We know this based on any number of, of security tools that have been released into open source and defects in them have been detected by the community itself because the processes of providing the security are transparent.

Jon Gallagher 26:33
The actual secrets obviously are not the people who are authorized to use the security are not transparent, but the transparency of the process is key. That's going to be challenged with generative AI, isn't it?

Logan Gallagher 26:44
Yeah, with these models, because they're so complex, because there are so many factors that contribute to the response that the model will generate, it can be really difficult to kind of determine what were the steps in logic that led the model to arrive at this response.

Logan Gallagher 27:01
And that type of thinking, sort of trying to reverse engineer how a model came to some prediction or other response is referred to as interpretability, and interpretability is really important for many other machine learning models, where we can determine how it arrived at the prediction or the other inference that it came to.

Logan Gallagher 27:24
We can use things like what's known as a feature -based explanation, where we can see which features of the training dataset were the most influential, and that model arriving at that particular prediction.

Logan Gallagher 27:36
This becomes a lot less possible, a lot more difficult to achieve with a generative model, and I think that that really contributes to the other two issues that we already rose. Hallucinations and the security issue, it's going to become a lot more difficult to really troubleshoot and debug, because these models are quite opaque, and that's not even getting into the problem where a lot of these proprietary models, we have very little insight into their actual training data.

Logan Gallagher 28:04
Because their training data is a competitive advantage, they don't really want to tell us what data the model was trained on, and that can get even more difficult to really understand its behavior.

Jon Gallagher 28:14
Another thing about transparency is just security requires that we yank on the padlock, you know, if you have a security person who's, who's going around to ensure that a building is secure, they are responsible for testing the doorknobs, yanking on the, on the padlocks to make sure it's locked and the, the infrastructure, the generative AI model, the billions of pathways things can take through, it's going to be very challenging to create a system that can guarantee I made this request and it is

denied under every scenario inside the engine itself.

Jon Gallagher 28:47
But I'm never going to get security classification or pass a security audit unless the auditor can prove that an attempt to break in is denied.

Logan Gallagher 28:57
And that does take us to our last of our concerns, I suppose, which is testing that these models, again, because they can provide myriad different responses to the same request. You ask it a question five times, it could give you five different responses.

Logan Gallagher 29:14
And that presents a problem for many existing testing strategies. Most software testing strategies involve passing some input into the software and expecting a determined output to be returned. With these models that can return many different outputs all to the same input, testing becomes a lot more difficult.

Logan Gallagher 29:37
You have to take more advanced testing patterns where you might set up what can be known as gold case set of outputs and then compare the outputs that you got from the model to your gold case outputs because there's not gonna be a one -to -one correlation.

Logan Gallagher 29:54
Or you might need to use humans to actually annotate and judge the outputs that are getting from the model. So these processes may have to be less automated to incur existing software testing patterns because they do sometimes require humans to help evaluate how accurate the answers were that the model generated.

Logan Gallagher 30:16
Regardless of the testing pattern you're going to take, it's gonna be a lot more complex than the existing testing patterns for a lot of existing software.

Jon Gallagher 30:26
Another part of the testing patterns, and this is the PTSD of your applications be a gateway into someone cracking a system, things like buffer overruns, things like submitting a value over and over again and testing the range of values that can be submitted just so that the mechanical process of taking data in and processing that data can be validated is gonna be incredibly difficult with these systems.

Jon Gallagher 30:52
It doesn't lend itself right now to a gold case of I was able to send this amount of volume to it. One of the classic cracking scenarios is to have a process that's listening on a box that you think may be working in root mode and you shoot data at it until that can no longer absorb the data and it crashes and sometimes leaving its permission level available for you to operate in.

Jon Gallagher 31:18
How do we know whether the generative AI systems are vulnerable to that? That's a fairly basic one. One assumes they've done proper buffer checking, but

Logan Gallagher 31:30
I think that the field of genitive AI model penetration testing is going to be new and emerging.

Jon Gallagher 31:38
I think a lot of people are going to be driving a very expensive cars on in both the white hat and particularly the

Logan Gallagher 31:44
Yeah, it also presents a soft target.

Jon Gallagher 31:48
So this has been kind of our cautionary episode. We think that there is a lot of potential for this technology. As with every other technology, and again, we'll put up the hype cycle, it's been oversold, not necessarily maliciously, but the result is that

the mind space it occupies is incorrect, that we have this tool that we can plug into, we can easily get answers out of, and what we would like to say is that without the work, without collecting the data, understanding your operations, and managing the generative AI environment, you are not going to get something that is production ready or you can rely on.

Jon Gallagher 32:27

If you haven't done the work, if you haven't collected the data, if you don't have a team with subject matter expertise, you're gonna be subject to things like hallucinations and data exfiltration. So there's no such thing as a free lunch.

Jon Gallagher 32:41

I think that's been a common theme to all of our episodes, and we want to emphasize that the lack of a free lunch here.

Logan Gallagher 32:49

I mean, I think one other thing I would say about that, about it being oversold is, I think that calling this technology AI was very deliberate and intentional. They want us to evoke the feelings of science fiction and HAL 9000 and iRobot.

Logan Gallagher 33:07

But the fact of the matter is that we're not there. And this is gonna take a lot of legwork to actually make these systems work effectively in the present day.

Jon Gallagher 33:15

Yeah, we, we're not there because we haven't collected all the world's information to feed through a machine learning model. The question then is, is it, is it ever possible to collect all the world's information, therefore create conscious life, but we'll leave that one right there.

Logan Gallagher 33:28

Yeah, not right now, and not anytime soon, I'm willing to put down that marker.

Jon Gallagher 33:35

Okay. So again, hopefully the cautionary tales helped and our next episode of some, some point in the future is going to be talking about some of the preliminary results we've made, some preliminary successes we've had in using AI for, for products.

Jon Gallagher 33:52

We are taking it slow. We're not looking for funding right now, for example, but on top of everything else, we are trying to make sure that we've learned these lessons before any commitment of, of time and effort.

Jon Gallagher 34:04

So thank you all for listening. See you next time. Bye -bye.