CD: Okay, Hello everybody. I'm Chris and I've been doing agent development of some sort or another for almost a year and a half now certainly since March 2023 and had a lot of experience doing this just practical experience. I know we're in sort of a fast-moving space and there's a lot of the ground underneath us changing sort of from release to release, but I thought I'd talk a little bit about my experience building agent systems and how I ended up sort of moving away from a multi-agent approach after trying it very early on I'm gonna got some examples to sort of talk about that maybe a little bit more specifically CD: So just to introduce myself talk a little bit about how I got into this. I built a local news analysis website like a data journalism website called torontoverse.com and a big part of that was that we had an interactive sort of map based infographic that went along with pretty much almost every article one of the things that blew me away when I first got beta access to gpt4 was that it was actually able to code our interactive components basically on day one. I only had to give it a couple of examples of what to do because a lot of the pieces were sort of mapbox and D3 stuff that clearly the model had already learned very, it was able to write that code just for me from my perspective shockingly well, CD: and then around that time in March, I'd read about the react pattern and I think maybe a lot of people started to put two and two together and realized that I could just have gpt4 do a lot of the heavy lifting, I could easily get the journalists that I was working with to work with chat. I think to get articles together but actually loading them into the system and testing them and Publishing them took a lot of manual steps at the time. We're actually using a GitHub repository as our database for Content, which would then be synced into redis and served live off of redis, but, the react pattern was away for me to basically automate the act of taking this code and getting it up onto the site and I also used it for the whole CMS interface at the end because HTML for articles and all that sort of stuff was relatively difficult for the journalist to put together, so CD: That Network shockingly and a lot of the experience this is gonna have here will derive from that experience. Unfortunately shortly after I did that bill C 18 hit and I don't know if you know me to really difficult to do sort of interesting or independent publishing in Canada, because believe it or not audiences tend to come from Facebook and they no longer publish Canadian news. So now I'm developing agents full-time at Tribble where we're building agents for sales enablement, which is a really cool use case, I think so what do I mean when I say multi-agent? I feel like this is maybe a lot of these terms are sort of loosely defined at this stage is a lot of fast moving Parts a lot of things changing and for me, I basically think of multi-agent where we have multiple instances of an agent Loop that have different behaviors that the user is interacting with or interacting with each other. CD: I tend to think of multi-agent as these things operating at roughly the same level of abstraction that doesn't mean they all have the same capabilities but that and large we're talking about multicom multiple copies of the same style of process. We call them all agents. It means they have something in common, and I guess, you do want them in a multi-agent system to be interacting with each other in order to produce that output maybe a simple definition, but that's how I would describe a multi-agent system at this stage. CD: And when I started developing Toronto versus my head, once I got one agent working. I was like, obviously we have to do is just get a bunch of these working together and it's like I have a whole team that I didn't have to hire which I think is sort of what a lot of people think when they see agents working. I noticed that a lot of digital human emphasis on the latest Nvidia presentation, and for me, there are other more practical things to look at other than it's like metaphorical attraction, there are some model limitations that you have to worry about or certainly that you worry about when you're building, one of the things is basically context window how much context can you give a single agent to do something and I think that makes you think okay how many tool definitions gonna give a single agent I managed to get over 20 tools with one agent and never had a problem with it being able to call the right one at the right time. Now, I haven't been doing that with open AI tools because I think their tool definitions are relatively for both being Jason schema in my original react pattern ones were much simpler and I was able to get much more of them into the model. CD: but eventually there's a cost component to this. So you think okay if I can have different agents with smaller toolkits. I can do things a much more focused ways that ultimately will cost less. I think there's also a draw of asynchronous interaction with they have multiple agents working at the same time. One of them can be off busy doing X. the other one is doing Transparency, which is where if the agents are talking to each other using the kind of language that you would use you can sort of observe what's happening and see how the agents are operating with each other I think is a human interacting with technology that's pretty new that's attractive. CD: and then of course, what I like is user composable. So one of my favorite things about agent systems in general is you kind of never know what you've built when you first put it together you get an agent you give it some tools to access some, specific workflows probably give an output ability to write to some sort of output system like a database or an email or a doc or something and you really can spend a lot of time playing around to see what the agents actually capable of and so part of what I thought would be really exciting is having multiple different agents that then can each do individual things that the user can compose together. So that's what I would say is what made me think that I would love working in a multi-agent environment. And so I just have a little example of what that would look like. I bought sore called Scotty which is my sort of, themed after Star Trek's Scotty has a sort of Scottish accent sometimes and Rosie who was my programming but CD: and I think this is a case where sort of the rent Kyle who was one of the people working in Toronto verse he'd sort of asked, Scotty to go work on something and he assigned a programming task to one of the other agents Rosie and he said Scott he hadn't ask function for all the other agents that I didn't want to talk to and then you can sort of continue on working with Scotty while that's happening. And then Rosie comes back the response and then Scotty handles it. I think that to me seemed like it would be a really great way to interact with the system. But what might surprise you is that it ended up being complexity at the wrong level and after just maybe four weeks of trying this sort of development. I backed away from it and shifted to a different approach. CD: And so why did

I do that? So few things that points I collected here to capture. why one is that interaging communication in sort of user land text like user understandable text. It introduced a lot of friction in the process this kind of one hint that's here a little bit. But in this case, maybe you can handle these things with more careful output handling but there's a little script marker at the top that made it a little bit hard for the other agent to understand what was going on there frequently misunderstandings and overlooked details, so CD: when two agents talk to each other you may have found this we talked to check GPT. It's sometimes will hit something really well, but it won't get the full c*** or certainly in the early days wouldn't get the full context of the entire message or things just get left behind because it's kind of hitting it's like Target token limit or it's internal token limit. It doesn't like to go on for a long long period of time and sort of Clues things up relatively quickly, and you can manage that friction. I don't think it's impossible to prevent that from you can deal with that for sure the ways I would probably deal with it are using Jason structure for the output and making sure that I really get exactly what I'm expecting from Agents when I interact with them, but once you start getting into more structured output, you lose the true transparency of these things interacting with each other very plainly. CD: I think in general we've all experience that high-level language instructions don't have a hundred percent success rate with most models, they're things you can choose to try to make that better but it does kind of take a little bit of back and forth and when you're coordinating multiple agents when things aren't working correctly. My experience has been that you end up having to spend a lot of time on that friction. And as you start to think what do I need to do to manage this to make sure that these sub, agent interactions can be managed in more predictable and reliable way. the number one thing that I ended up doing was basically defining these workflows. So things like programming for the agent as sort of a higher level agent So flow an element driven workflow, that's call that multiple, steps that have to be taken together but then triggered by a higher level tool call. And so what we end up with is sort of One agent where's multiple hats depending on what tool call has been engaged but you as a user have one interface that you can deal with in order to handle that. CD: Tool calls especially in the modern API, but even before gpt4 is able to call them with a very high degree of reliability. If you having trouble with reliability with tool calls from your agent chances are you're confusing it in your prompts somewhere. It doesn't handle confusion. especially with tool calls, it will tend to just do the wrong thing as opposed to saying I'm confused. do you which is maybe what you would want as the system developer. CD: High level tools that mirror those high level agent instructions really ended up working very well for me. I think there's tends to be a lot of thought about composing tools at a low level can I rake to Google Drive? Can I fetch a website for me the answer became, creating working spaces in data models outside of the agent that could then be manipulated with these sub workflows. So imagine having an article table that the agent can use its own little workspace to build up a little article and then you can sync that article up to GitHub with an additional tool call rather than having separate agents manage different parts of the process. You can work with one agent to develop a workflow that they can captured into a higher level tool called that can be executed in the future. CD: These per agent the state basically to manage what tools are available to the agent. So this is a really easy one. But one of the worries about multi-agent photos, I can't give all the tools that I want you to give an agent, but in fact most of the time if your tools that are sufficiently high level you can give enough of them to the agent and then as the state of either the conversation or the agent itself changes, you can then provide additional tools on future requests that reflect the state of the call. So in my case if you were working on an article at the agent not just doing research but actually writing an article then You would be able to create an article start working on it and then all of a sudden a whole, Universe of article manipulation tools would appear inside the tool set for the agent, but you wouldn't have those on every single call and it wouldn't be carried along with the agent at all times. CD: And so as we get down to that how does that end up manifesting? I'm doing sort of similar type work. Unfortunately all my bad examples. I didn't keep messy when I'm doing development. But here's a great one that I did recently that shows off a sort of like one agent using this approach where it changed these tool calls together into workflow to get the output of one and then uses it to call another one and here's a great example of that working. So This is what I did as an example at Global Mail AI journalism conference a couple of weeks ago. And here's sort of the instruction that I'm able to hand to the agent, which I've got a photo that I took with my phone and on slack and I said, hey, can we upload this photo and make it the Articles feature image and… AF: Okay. Beautiful. CD: then add it to the body with a figure. I'm trying to be very explicit for demonstration purposes. AF: Thank you so much, Chris. CD: But the caption AI discussion at the Global Mail offices. AF: So I want to recap so. Can we recap the bottom line you're saying instead of multi-agents create better tools and set up the same agent to use those tools better. AF: Alright and… CD: And so here that … AF: dynamically introduced to some sort of a tool retrieval is… CD: the first thing the agent knows to do is to call the upload to CDN function and then here goes and… AF: what you have in mind there or… CD: sets the article property for the feature image gotten that imageurl from the CDN,… AF: some sort of a subset routing. CD: So s upload it gotten that function back from the CDN and received a URL to use and then it's able to go on and update the contents of the article accordingly. Here's a cool one. We're combines everything together into an additional tool call where it actually just nose to creates an HTML figure inserts the link to the URL at alt text and includes the Fig caption. These are little things but in terms of if your journalist or content creator and you're trying to create automating all these steps and when they might have taken you 15 minutes or… AF: Right, okay, and even like the personality prompts and… CD: 10 minutes to Courtney to do on your own because these are not natural behaviors for people necessarily especially you've got every quirky system like mine and… AF: things like that that people like to use when they say multi agent. CD: you're with Toronto versus and… AF: I given that could be set up as one of the tools that the system uses, right. CD: hiring Freelancers on this streamlined a whole lot of the process and boom you get this which is the perfect figure to drop into the article. All sort of composed by the agent over for I think in this case consecutive tool calls. AF: . AF: yeah, so language is definitely an interesting one because In the more I think you and I when we were chatting we were talking about, micro Services architecture is in how you… CD: Yeah, and… AF: Json unrest… CD: that's kind of my journey at a very high level from using multiple agents to do some of these things to settling into a pattern of using a single agent with higher level tools do accomplish things. AF: where pretty successful there because they had very strict descriptions of how these different Services could communicate and… CD: I've continued,… AF: all of a sudden you're doing this Leapfrog from Dad into. CD: I don't have a good example from triple to show off here. AF: Yeah, these software Services can talk to each other in natural language robustly. CD: But really chase this model that triple aggressively and… AF: What do you think is missing in that conversation based on your observation? CD: I've seen it be very successful there. So, I'm quite confident. there a lot of legs on this approach before we really need to get into true multi-agent systems and maybe my opinionated bottom line on this is that where we really will end up using multi agent systems is when we don't want somebody else's agent to be acting on our behalf not when we need it or when there's a technical Advantage, but when we choose that, ire we prefer to have this agent and not that agent doing the work for us. Yeah. CD: And they stay now so dynamically control what tools are available. You sub workflows within one agent, but use that single interactive agent Loop to drive user interaction that's my bottom line my takeaway. CD: So right now I think it's driven primarily on conversation State or… AF: Definitely. I mean the reliability is definitely interesting. CD: the agent state. So, do any exist in the universe of this agent or… AF: In your experience in your experiments,… CD: is this conversation currently working on… AF: I guess did you ever play with domain a specific languages as something between Json and… CD: why and then we use that to bring sort of an appropriate toolkit to Bear but I also kind of think that in the general case if you needed to limit tool sizes a rag style approach… AF: natural language for a means of communication? CD: where you basically did a vector search over a bunch of tools given a context and got back a subset of them to make available. What will be pretty successful if you wanted to have hundreds of thousands or something like that in the database? CD: Yeah, absolutely. And for me in terms of how did I there's a question here in the chat mode mention that the way we initialize the multiple agents. I just entirely separate software processes running. I'm in the background and I had them actually coordinating with each other via slack that was just the development playground. I was working in it's not the only way to do it. It just worked for me. Each one had it completely independent prompt each agent. AF: Right, but I guess could be clear. I meant DSL as a means of interactions between agents rather than the person. CD: Mm-hmm CD: What is missing from the conversation in our conversation about how these agents could work together? The latter one, I think. AF: right That's interesting. So. one of the things that usually I As you said like that the definition of Asian and what multi Asian means and all of these things are so fluid right now that everybody is somewhere with the definitions. So usually when I try to think about these things I try to ground myself in the reinforcement learning definitions of Asians and... CD: The experience that I've had developing with llms trying to do real productive things with them and hopefully it's something that a lot of people have experienced. AF: start from there and go to places right and... CD: Is that getting to a really impressive demo is often relatively straightforward compared to productionizing into a reliable piece of technology. AF: in the context of AF: And there are a lot of parallels obviously there and... CD: And I think that I've been able to myself develop a bunch of... AF: in the context of what you're talking about policies is a thing that is usually missing in these multi-agent element conversations,... CD: what I thought were really cool multi-agent demos, but when I came to getting the system to produce consistently and... AF: essentially... CD: reliably in the way that you... AF: how does system behave and if I'm reading that in your lines... CD: I needed it for the business goals, you multi-agent interviews friction that just took away from that at this stage. AF: what you found was that the only really meaningful policy that is happening in most of these systems in a robust way is going to be chaining and... CD: Maybe it's possible that models will get better. But as the models get better. They also become more capable in the single agent use case. AF: routing. a bunch of self-asks that you build together CD: So for me, I think I'm struggling to understand what I can do in a multi-agent use case that I couldn't do in a single agent use case with a lot more reliability. That's what's missing from my perspective. CD: So that's an interesting point. I think in general when I work with an agent, I'd like every single one is developed a kind of vocabulary that I know works. and that becomes effectively a DSL that I know will work. with that agent, when I think about hardening them for broader uses use cases, I think it means it's like eliminating the dsls that you develop when you're working intimately with an agent that you're very familiar with and then making it work really well when somebody walks at it cold and doesn't know the specific,... AF: Bed and... CD: I'm trigger words that are really gonna get the agent to do... AF: that generated a thought usually helps in,... CD: what you want. So You... AF: hitting the right keywords only in that context. CD: I think in general I try to eliminate dsls and... AF: in some cases other CD: just use straight programming what I want to get something done. If I wanted to do something like, algorithmic, I'll use a programming language to do that. And otherwise I've been trying to break the dsls or the domain specific approaches to doing things to sort of loosen things up when another user approaches the system. AF: Yes, okay, right coming back. Yes. Sorry. My internet is a little funky. Sorry. I was saying that the only place that we have seen that thought generation as the first step helps is sometimes if the question asked requires a specific keywords to AF: The keywords versus just using the question that was asked and outside of that core expansion use case. We haven't seen a lot of usage for that thought process generation CD: Yes, Yeah, so to an extent absolutely I think that I ended up basically stumbling on eventually a DSL becomes an effectively and an open API definition right if it gets sufficiently complicated and you want to add descriptions to all of your parameters that they're well understood and that sort of thing, I think what I had was what it was effectively a DSL that we're like handwritten tool calls that I ended up using sort of they were not Jason schema functions. They're a little function signature in my own little language that could then be composed into essentially micro programs. but that approach it just veered so closely to just straight tool composition and straight tool composition by the model worked. So well that I never lingered on it for any period of time. in development CD: Muhammad CD: he CD: Those are the ones that I found work reliably for me. is chaining and routing but at the same time it's been generally very good at figuring out how to approach those tasks earlier days. I use sort of in the react pattern. there was a sort of a self prompting approach that we used a lot to get it to sort of say… SP: I can hear. CD: what you're thinking and then call the functions and I ended up pulling that out over time as I found that it was able to do the routing like the task composition. SP: Did we lose Amir though? CD: I'm figuring out what to do better with less instruction. the more accurate. I gave the more constrained it was and less reliable. It was the more I generally gave it the idea of the tools how well they could be used in a high level understanding of sort of what the use case was what the Persona of the agent was. It was able to make those decisions more reliably on its own rather than I mean, I was incredibly surprised by how well it performed when I first started building. It was definitely an inflection point in my software development career when I realized how good it was at doing this. CD: Muhammad CD: adi I think I lost you for a second there. Maybe you lost me. I'm back Iraq. SP: Amira seems to be back. SP: It's a bad for the encore. CD: I do like that the question expansion you say is I like rag as a tool call because the model has an opportunity to sort of see what the user said digest it and then generate a query for rag that's independent of what the user independently said, we do that at treble. I think we've not fully exploited that opportunity yet. It's one of the ones I feel like we have in our pocket to really improve things in the future. But I like that be able to think about it and then explicitly generate the query to then use for rag rather than just going with whatever the users first question was. In fact, we do both the triple, so CD: There's another question there in the chat about changing the tool calls. SP: Nice, so I actually joined the call a little late. So I may be repeating things so we can talk about something else if my next question or discussion point is something you have already covered, but I was wondering a key aspect of agent systems is it's your short term memory or… CD: I spent a lot of time thinking about what kind of decision making is happening when it's chaining these tool calls together. SP: memory that stores certain aspects that are required during the task while you are performing the tasks so things like Decisions made a list journaling. CD: Certainly a triple we've seen it call, 20 plus sometimes it gets carried away and just starts call functions back to back and that's one of the failure cases of specialty gpt4. I think if anyone who's in enough agent development to seeing an agent get into a loop once or twice that has happened before for sure and… SP: I would say a list of decisions made or… CD: I think we've been defensive about that from time to time. SP: versus something that you need to receive at. CD: I haven't seen it with turbo or… CD: with four. yet, but I have seen it with four just in my anecdotal experience right just some building this I think… SP: quite frequently into the context stuff like that and… SP: have you been so the earliest I guess approach that came out in the open source world,… CD: if we're saying yeah the less explicit you are with the model the more powerful. It seems to be or the more I'm willing to instead of open-mindedly. solve your problem. SP: I guess was mem GPT,… CD: It will be yeah,… SP: but I guess They were a little too creatures stick for the time when they came out to last year,… CD: I tend to really give it a deep description of what the tools are and how they work. I find tool descriptions in the prompt as well as in the actual Jason description of the function are key you need both. But if you can give it a really strong understanding about the tools are an understanding… SP: but I think many Frameworks have come for handling memory… CD: what it's role is in the company. SP: since then. I was wondering … CD: It's extremely good at making decisions on… SP: what do you think about … CD: what to do to accomplish that goal using the tools available. SP: how do you handle a short term memory storage and receivable? And do you use any of the current Frameworks? CD: Nope, maybe I left this time. Did I? CD: Okay, Good. I'm just making sure. CD: It seems like maybe we did. CD: Definitely. I mean maybe if anybody else has any questions, I think we can definitely pick up in the meantime rather than sitting here. SP: Yeah. CD: Yeah, okay, great. I mean so I CD: It's a great question. So for me the Things fall into a few categories, so just for this agent use case the triple use cases is a different one, but I'll dive into this one because I'm very comfortable just like being is the company shut down free transparent but everything does so one big area was just creating and preparing articles. So a pre-step for that is research. So one of the ages I introduced Rose to use my programming but I think that's more interesting. We had another one called Toronto bot that would do lookups on have open data sets. You could query them around the time that the frontal budget came out. it's big complex set of spreadsheets that are kind of hard to understand and what we needed was a great way to try to like mine something unique to say about what was in there. So then we had recently article creation. We'd research web research that URL but also this kind of cool show me the cities open data and then we also had basically publishing to the website so create an article, but then you've got this article in your local memory you want to be able to publish that to the CD: That was its own workflow and then getting it into GitHub. So my favorite one if you get everything built and then so great. Hey, can you cut me a GitHub PR it's able to take all of the stuff that it's put together and… SP: Muhammad Alright, yeah. Got it. CD: then create a pull request to load all that into GitHub… SP: You make some very interesting points there. CD: which is still our like, you… SP: So one we do. CD: I never rebuilt at the Toronto versus database of record. So yeah,… CD: and it was able to basically like,… SP: Implement the memory similarly one thing we also do is Whenever there is a preference. CD: depending on the nature of your task. It could easily chain all those tools together to do basic article creation like a first pass and… SP: That is stated we store that and… CD:

a prompt and a couple of Corrections. SP: that's going to be applied through the list of the user history. CD: hey, he's back. All right. SP: And in our case it's not actually users. These are financial documents and it's like rules that pertain to financial document. So for example in this particular doc in this set of documents all numbers are representative million. CD: You figured we just kept talking and keep talking. Yeah,… SP: And that's a role and… CD: but that,… SP: when we encounter such a role that role stays in memory and… CD: it putting those together and then of course, it's got this conversation state or… SP: the other point that I raised with I found interesting is and… CD: this agent state that both help as well. So, if there's no CDN configured then it knows not to upload to the CDN. SP: which also happen to completely agree with the Frameworks rather than empower the constrain,… CD: I do have a CDN figure. that's fine and per conversation State are we working on an article? Do we have an image? And then … SP: … CD: what tools are available based on whether those things are sort of in scope or not. SP: especially with agents. I guess you expedient meta GPT and what is that The one by Microsoft I forgot its name. harajem all of these. Yeah. I think they already made a lot of decisions for you. They decided This is going to be the Paradigm without the Community Battle testing it and we are left with the abstractions that don't really work. and then I had to build my own agent framework myself. CD: he SP: So yeah, but I guess it is hard for anyone building an agency framework to build the right set of abstractions. before at least a large community comes up, with the consensus. So yeah, hopefully those change over time what I heard was similar to the more General Frameworks like Landing etcare. had very hard to use abstractions and prematurely optimizing when they first came onto the scene, but what have been noticing recently that many large companies have lunch in production and I SP: So I haven't used land chains since a year and a half. But I wonder if did they change did they become better so that people are actually using in production or… CD: So I don't use any of the Frameworks. SP: is it just that? CD: I roll my own agents. I feel very comfortable doing that working. SP: It's just like there's no other option and… CD: I working, Pronto versus written and… SP: people just want to get started with something. CD: go but we also work in typescript at triple and… SP: I don't know CD: The way I feel very strongly that at least for me the Frameworks are sort of more constraining than they are empowering… SP: if you have. experiences CD: but I do think that's the memory topic is a super important one and I like to split it into two pieces and to me this is one of the key things to building very valuable agents in a specific domain. I think there's a general sense of memory that you need to have for your agent to be functional and then I think there's domain specific memory which I think of as in any specialist in the world who's gonna be really good at something any human specialist is gonna have tools that make them better at that. It's almost a universal truth, and that's like these are the domain specific tools where we want our agent to be powerful. That's where we can differentiate as the developers of the system and I like to treat that with a really special CD: Little case where I'll have really, type database tables that tightly model the domain meaningful, elements and things that it can manipulate so it's got a lot of power there but in the general sense, I think there's a sliding scale of things that you need for memory. The first one is really good context window management like conversation history management with an agent. The simplest approach that is recursive history summarization that gives you the basics and... SP: Yeah. CD: then from there. I like to build on memory remembering specific facts. I think in one of my favorites is remembrance, look reflecting back on the conversation periodically and asking yourself. How could I better work with this user and remembering that on an individual user basis and just keeping this paragraph on a pre-use or basis up to date and dropping it into the system prompt as time goes SP: Yeah. Exactly speaking of Frameworks and abstractions. What it think or do you have experience with using patterns like DS Pai? SP: DS Pai SP: yeah. DS and py now and P Y so it's ds5. Yeah. Yeah. SP: No, no dspy the Spy. Yeah. CD: And so yeah, so for me, there's sort of these various elements one is the conversation state that is just a list of facts that can be stored and then pulled into the system prompt actually drop us something in the Scotty case to drop a summary message which has all this data. It's like the state setting system message that follows the original system message at the top of the agent Alum request and... SP: sure, so. CD: it will have that and it'll have the data about the user how the conversation has gone so far and then the conversation States. what do we know? Is there an ongoing article? What's the uuid for the article we're working on are there any images that have been uploaded in sort of as a part of our conversation State? You can expand this idea that if you're doing research collect a bunch of articles for research and then rag over just those articles with the tool called because you've collected sort of a list of those and store them in conversation State and then one that I've built but not used very much is then, Global agent State on top of that. So I tend to you so that's my generic approach to memory. SP: it's a design pattern for building based applications and what they have. So they structure it into two types of software components modules and then optimizers so modules can be like your control flow like the prompts from chaining and then the optimizer are SP: Things that you can optimize or change which is let's say A model who's parameters that you can change it could be few short learning where you can dynamically change the few short examples and so So what happens is they have you can Implement these either use existing from this particular framework or Implement your own rule for changing either the parameters or the few short. So either fine tuning or... CD: autogen SP: Dynamics you start selection and that basically happens under the hood now because you have built a function for that. So now you can iterate through different models or different control flows while keeping everything else like another it's a little I guess a difficult to get into initially but yeah, it's SP: yeah. SP: Yeah, I think for now the Dynamic fine tuning is still futuristic because it's something that it's not that easily controllable, but I think the biggest gain, that I get right now is the fused Dynamic few short selection where that is something. yeah. yeah, it is being used I guess. SP: in I know some Vector DBS are putting this hard as part of the red workflow. So many of the vector Davies are putting out notebooks that try to implement this pattern and yeah. Yeah. SP: Okay. SP: Nice and have you looked at the lmql? CD: So I think if you're not a programmer and. . . SP: Lmql. . . . CD: you're not super comfortable building this stuff and. . . SP: yeah, it's already there. CD: familiar with the open AI apis intimately. SP: Yeah, yeah. CD: I feel like those libraries have a lot to offer to somebody who's building. SP: So this one yeah,. . . CD: I personally watched lot we were at triple oxide. SP: so this one I guess the main. Thing it provides. CD: You're like trying to tackle some problem. . . SP: This is also another programming pattern for building a little applications. CD: what we're looking at the Lama index implementation of something in this space. I steal ideas from those Frameworks all the time. And I feel like they do act as like,. . . SP: I guess It has that structure generation part. . . CD: collection points For Thoughts From lots of companies and people that are working on this. SP: where you can provide regular expressions. in pythonic form and also provide CFG context free grammars that You want your output to be structured in ETC? But I think it's also provides A pythonic way to John. Both specify input prompts and output constraints Etc. So yeah, so I guess these are some of the more, abstractions the programming after actions that I'm seeing. CD: So I consider them a really good source of ideas for how to build things and how to approach, a lot of these problems. SP: Yeah, yeah. CD: I think even prompt Construction in the early days. I spent a lot of time looking at how language chain was constructing his prompts. I don't think I do that very much anymore. But there might be some wisdom still there to pull out of it. I definitely do think they're a great source of ideas,. . . SP: So, I'm curious what? CD: but I feel like and the people that are trying to build these agents for real practical use cases and getting user feedback when they don't work are really learning so much that. . . SP: And I think other people can jump into afterwards, but I'm curious. . . CD: if you step back and you're sort of at the framework level already in a technology,. . . SP: how do you implement structure generation? CD: that's so immature. How do you even know. . . CD: what abstractions to build right? I mean, it's so difficult to do that. SP: Do you? Structure generation,. . . CD: You don't have this. I don't even know and I feel like because I'm getting constant feedback that makes me question a lot of these decisions and. . . SP: for example, the output needs to be of a particular format. CD: that's where I feel like I'm super uncomfortable committing to one way of doing things just yet at least in this model. SP: Let's say it should be well formed Json or it should have a very particular structure that can be easily possible later on Etc. CD: We maintain a lot of control over, exactly what the major architectural abstractions and our agents are. I think that means that's gonna be very important as the ground beneath the ships with new models and CD: And things like that like GPT 5. What's that going to do to everything? It's gonna make a huge difference. CD: No, I don't the DFI. CD: DS by how do I spell it? CD: and py dsnpy. No, okay. So I want to write the sense. I want to learn what it is after this, but no that you caught me on something. I don't know yet. CD: Dspy. This okay, and I've seen this before. I know I have not worked with okay, so I have seen this but not worked with it for programming up prompting. SP: remember CD: Do you want to talk a little bit about it for context? SP: That's a nice. So I usually work with the smaller models where it's a little harder to do that. So sometimes we build our own. Checking tools verifiers and… CD: Mm-hmm SP: then have some back off methods. But there are also some libraries like Json There's a library from Microsoft called guidance… CD: Muhammad SP: But dude also notice. other forms of structure generation, for example, let's say you want it to be in a very SP: ADI want a date and a Time Etc. And it needs to be in a particular format. Of course, you can resolve this with software later on, but maybe that was not the best example, but any other SP: okay. CD: It sounds cool. I mean so this sounds like a way to sort of be more Dynamic with the actual structure of your subtask workflows. AF: Or what have you used? CD: So for me a lot of it is really detailed there man. CD: Muhammad CD: So one thing is based in formatting, one thing maybe I should say on this is that this is newly become for a long time. I think we had a really strong degree of reliability that it kind of made me feel a bit complacent. We've recently been seeing some chaos with 40 doing tool calls… CD: I'm after this call. CD: things like that that have led me to be a little bit more defensive about some of these interfaces. CD: I'm gonna digest it for sure. This looks really cool. CD: One thing is the Jason's not formatted correctly, but for me, using the Json that's captured. Yeah, probably we're gonna start validating the schema. So the first failure is look you didn't respect the schema for the tool call, but from that tool called providing a rich error message… CD: Okay. CD: if something was incorrectly formatted or didn't look correct that describes what actually went wrong usually don't need to quote the source back because it'll already be in the message history with the llm if you're sending it back… CD: It sounds really interesting. CD: but basically being able to do detailed domain specific or… CD: This is exactly the kind of thing. I'm here to learn about. CD: domain relevant error messages about the contents of the content and… CD: So I'm looking forward to digging into it. CD: saying look, I didn't like this and having to come back and try again. That's Lmql this one I have seen but I don't remember why I didn't know lmql Is there perfect for element interaction? There you go. This is cool, right? That's been sort of a deterministic way to do it. I also often do loops where I ask it to try again verbally because I don't like the result of tool called generation generation D. But that's the extent of what I've experimented with. CD: SP: ADI we want Iator, you could talk a little bit more about gbnf. CD: Mm- Please I'm curious. SP: Sorry can't speak now. Yeah. CD: It's okay. one of the things I love about using open models like llama 3 was being able to specify the grammar for output and I feel like there's a lot of opportunities there in the future for doing some pretty cool stuff. CD: Yeah, this looks very cool. SP: Okay, nice, and then there's the comment from jayant thing. CD: cool SP: Yeah instructors. So yeah, this is another Library. for structure generation So I guess there is instructor there is guidance. There is lmql. CD: Nice, I mean I did this is really interesting. So I think we're at your very early stages and I think there's a Temptation when you see something awesome like agents Yeah, this looks very cool. Okay, this is really interesting. And hey,… SP: Yeah. CD: agents taking off to really want to find the infrastructure my favorite language go. There we… CD: infrastructure piece that's gonna make sense and invest heavily in it. And with so many ideas out there right now, I think for me it's like, That very cool. I'm definitely gonna follow up on this one. CD: SP: Okay. SP: Yeah, I think. CD: we need to learn about all these at the idea level. SP: I still. CD: It's really What are the ones that are gonna stand the test of time. SP: shy away from using these and… CD: It's pretty exciting to work in a space like that to be honest like a very technology like that. SP: prefer building something internally one way that I do it is if SP: ADI using open AI this one of their arguments is where you can zero out CD: We see. SP: the Logic or the preludgets? So that you can prevent. all different tokens from being generated and… CD: Mm-hmm SP: then you can SP: restrict the generation to only a small subset of tokens and SP: worker is enabling well for the restricted use cases that I was in. But yeah. any other questions from anyone else in the fall? SP: And Amir have one big question. it's AF: I have so many questions. Are there any use cases I guess? It either yourself Chris suhas or anybody else on a call any use cases for fine tuning language models for Tool usage or do you just find that with good enough descriptions of what the tool can do they can usually pick the right one. CD: I think I have it hit the limit of what we can do with just prompting just yet. CD: So yeah,… CD: I feel like we've got a little bit of ways to… CD: so in the early days before the arrival of Jason mode and… CD: I'm excited for when we get to find two models to do our agent work right now the models themselves. CD: turbo, I was basically begging the llm for things went ahead and… CD: I'm getting better faster than I think I could make them better by fine-tuning. CD: error. I would count I'd record the air and that's in the air back to the llm and… SP: And there are a few to use data sets. CD: hopefully in most cases get properly structured data back and… SP: I think we have discussed them in the… CD: sort of like the second reply that worked. He surprisingly Mm-hmm. CD: surprisingly I would show it with… SP: I don't remember. the names of them CD: what the problem was with the original call and then send it back usually will be something like super common one would be,… SP: but I mean, I remember we discussed them in previous sessions at least maybe six months ago or… CD: hard Carriage returns in adjacent string right which can't work in Jason. So, eventually I think I switched to Jason 5 and… SP: something and… CD: maybe I'm not sure about… SP: I do think those are useful,… CD: what solution solve that in the long run but for a while it was begging I had great luck with Jason mode for outputs and… SP: especially if CD: then tool calls for a lot of structured again, which is effectively Jason mode for those outputs. SP: you have these sequence data sets for Your very specific domain and… CD: And I've been using that with open Ai and it's been Rock Solid for me that that's been sort of like getting good structure Jason out again,… SP: use cases. CD: when it's failing I've almost always been able to trace it to something confusing in my prompting that they m- Yeah. SP: We have yeah. CD: I know that one if you guys know this to sort of funk, there are some fine-tuned open source models for function calling that I'm curious if anybody's played with them. lifting instructions something ambiguous that had misinterpreted but otherwise things usually have been quite solid for me using that to get structure back out. CD: CD: give me SP: I haven't checked this particular one. ad CD: It's just one that I think this is not a very popular one, but there are a bunch. SP: but yeah, I don't think for the smaller models. You definitely need the fine tuning. and I guess SP: yeah your comment about saturating the extent of prompting is applicable to gpt4 or something like that. But yeah, I think for the smaller models you'll definitely need the fine training portal use CD: I mean, I'm wondering you think if you've been using Xenon had good success with small models, that's a really interesting use case. SP: I use only small models right now. I mean we use gp4 for one particular aspect... CD: right SP: which is ironic. So I don't know if this is a widespread phenomenon, but over the past few months or over the last year our DPT for proprietary model. You say just come down a lot and open source model you say this one up and right now the only use case for which we use gp4 is for a summarization thing where the only reason we are keeping it is not... CD: mmm SP: because of the quality,... CD: Yeah, so I have used tool calls for exactly that use case before... SP: but because our customer expects it in a particular tone,... CD: where I need something in certain data when it spit out and... SP: which we can't replicate using open source,... CD: I will actually have the agent make a tool call and then manually output like the data via slack from that tool call and today... SP: so It's just the cost of switching. CD: I've done that before and... SP: It's yeah. CD: also gives you a chance to error check the age if it generated bad data, then you can reject that data send back an error That's very encouraging to hear that's super encouraging. SP: yeah, ... CD: get proper Data before displaying it to the user SP: yeah, literally yeah,... CD: but I will use that approach and SP: that's the only case... CD: then actually SP: but yeah, I think it's CD: instead of having the print slack message directly. The tool called will... SP: with fine tuning A lot of and... CD: then print the slack message that says what the agent should have said but with a higher degree of reliability. SP: this is Possible only in recent times with Lama 3 5 3 5 3 and... CD: Muhammad SP: not I think before that there was still a huge qualitative difference. CD: That's a really good to know because I think the encourages me to go back and take another shot at these open models the smaller models and see what they can do. SP: right and jayant asks small models are 7 B. Unfortunately. Yes, that's what I mean. Yeah. Yeah at least seven weeks.