

LLM Eval Office Hours #2: LLM Observability

24 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=TZWMLXXFBH4](https://www.youtube.com/watch?v=TZWMLXXFBH4)

<https://www.youtube.com/watch?v=TZWMLXXFBH4>

SUMMARY

The discussion revolves around the importance of building user interfaces (UIs) for better visibility and performance tracking in AI applications. The speaker shares insights on how to streamline the process of creating UIs and emphasizes the significance of logging and observability tools to enhance data analysis and debugging.

- Building a UI can significantly improve the ease of accessing and exploring performance issues in AI systems.*
- Initial UI development may take longer, but experience reduces the time required to create effective interfaces.*
- Using tools like spreadsheets or notebooks can be a simple starting point for data visualization before moving to more complex UIs.*
- Observability tools are essential for logging errors and tracing data paths, which help in understanding system performance and failures.*
- Error analysis is a practical method for evaluating AI systems by categorizing problems based on logged data.*
- Tools designed for logging and observability can save time and resources compared to building custom solutions from scratch.*
- It's crucial to balance between developing internal tools and utilizing existing solutions to focus on product development.*
- Continuous feedback and iteration based on real usage data help refine AI applications and improve their effectiveness.*

I read your articles and you you like brought up some points about like building a UI to make it like to make addressing these problems more more like tangible like you're you're able to access and explore what's wrong with with your service a lot with a lot more ease I actually haven't built any of that and what I've basically been doing over the course of a couple a couple projects including the sales development representative project which I might might have alluded

to is I just utilize for example I use fast API API to determine to explore whether or not my agent is performing is doing what it's supposed to be doing and it is like it's definitely better than not having any visibility but it's still a bit cumbersome to navigate so I I was curious like how much time do you how much time do you

spend into making these these UI to have a better visual of of how the performance of your system yeah it keeps getting shorter and shorter in the amount of time I need to spend making it I think the first time it may have taken me like three or four days just to understand even if this is a good idea or what I'm what am I doing and like kind of figuring out what I even

want but then nowadays it takes me you know maybe like two or three hours to to do something like that and

especially if you if you know what where the data is I guess the the time consuming part is like querying the data from wherever you're storing it and then materializing it in in some front end and you have to have some understanding of like the structure of your data and how you want to show it once you have some workflow that you

like like some place you like storing your data maybe it's you know Langs Smith Brain Trust rise or maybe it's a SQL light database that you have I don't know what you want but you know as long as you have a workflow that you like and you know that data then you shouldn't take too long especially and you know obviously you should be using AI assisted tools to help you build these braw works yeah right Fair yeah it's

just I never really considered using a using a front end but I do recognize that it it would it would make make things a lot easier it just seems like a pretty massive it it just appears like a pretty big undertaking but if you just keep it simple I guess it's not too bad no just keep it really simple so basically the idea is like you want to see all the information you need in one screen you don't want to be jumping

around to like this database here some other screen here your traces in one place you need all the context in one place you know you know bring it all together put it on the screen doesn't have to be pretty or anything it just so you can read it and and you can like flip through data really fast and maybe like label something or take some notes do whatever I mean maybe you don't even need to take

notes like you kind have some idea of what you need probably and that you can just like quickly look at data to where the like the idea is looking at data shouldn't stress you out at all you you you shouldn't think in your mind oh I need to look at data it's going to be kind of yeah like this going to be painful should be like oh look at data oh that's really fast I'm just GNA open

this app and I'm just gonna like check out some stuff and then okay I will say this you so the first place to begin is not even building an app it's like first can you use a spreadsheet that is the simplest way to view data you know that exists it's I would rather actually build a front than use a spr sheet but yeah I mean I can I create a CSV file I can create an Excel doc but not my

strength sure so like yeah Excel can be super easy depending on your data if your data is like a lot of multi-turn conversations or something then like then the spreadsheet might become unwieldy because you have a lot of stuff going on in each row and you might want something else but yeah and like their spreadsheet okay like you can walk up towards UI there's you know so you can do spreadsheet you can even do notebooks so

Jupyter notebooks or whatever I don't know what do you program in like typescript or python or any yeah typescript typescript python I I don't do a lot of work on collab and Jupyter notebooks yeah yeah so you can kind of like with gradio and and you know in notebooks you can kind of make your own little like sort of UI thingy looking thing and yeah that's also good it

depends what you're comfortable with okay what are we looking at open plug right so this is this is one of the projects that I worked on and what one of one of the major problems that I ran into and it was very hard to very hard to address was that this this entire project had a lot of dependencies like a lot of chaining dependencies one was you have an llm that can fail at like inference time in

FA failing to produce like the function calls for example and then on top of that like the the so this is have you ever used chbd plugins a long time ago yeah yeah yeah so this is that's when this basically existed that's when this was relevant but I think that the concept that I'm going to get at it's is still somewhat relevant so the function call might fail for example then after that the the plug-in that it's trying to

like reach out to which is kind of an unreliable third-party service might fail as well and then presenting that to the user might have failed as well so there was like several chaining areas of potential several chaining pitfalls got it being able to like address that I what I how I addressed it was just print logging as much as I could and particularly like Lo logging the errors whenever they will have let me

tell you something that will help you so if you have are you using any framework like I don't know like Lang chain or something are you are you rolling your own are just curious like no no I'm not okay no problem so what I would do is I would use some observability tools So like there's a lot of them out there it depends like what you like you know there's paid stuff so there's I mentioned some of

this before like Brain Trust there's Langs Smith arise has a paid one there also an open source thing that I quite like it's called Phoenix Phoenix arise it's like the open source kind of thing it has a python SDK so if you if you don't like python then maybe Brain Trust you know could be good as well you know like if you're into typescript but basically you want to log the traces you can instrument your code and it will

there's like very convenient ways to instrument your code such that it'll follow the whole the whole code path or the whole like llm path like from you know you'll be able to see the entire trace of what happened like what the prompt was what the function call was what the response from the function was everything it's a nice way that it's kind of like standard in the industry becoming a standard in Industry to like log your data this way and once you log

your data this this way now you will have a whole host of tools that will allow you to go further so if you log your data into one of these platforms number one you can easily pull it out and you can build your own uis on top of it you may have read in the blog post where I talk about building your own uis like I built a UI on top of lsmith for example CU I was working in lsmith at

the time and wrote that post oh isn't lsmith itself a UI it is but sometimes you want so the point of that post was like you want to customize the way you look at your data for yourself so like sometimes the data you push into here may not be in the format you want to see it you know because ultimately like like this is not necessarily a data viewer this is like more like a it's like

a data dog for llms this is a very bad analogy but it's kind of the closest I can get okay to help you to understand it's like so essentially like you you can put your data in here and it'll help you navigate and filter your data and like query that those logs in a in a very like structured way if that makes sense and Langs Smith lsmith provides

like does lsmith host the yeah all these tools like Brain Trust and lsmith the the paid ones are all hosted as far as I know and but the yeah like it stores the data for you is what yeah yeah it stores the data for you so if you may or may not like that the Open Source One Phoenix which you were just looking at that's like self host if you want actually like host one for you too

if you want but you can host yourself yeah they're all like pretty they're pretty easy to use you know depending on what you want you see you can see like you can see already how you get an idea right like how it makes it easier for you to look at your data yeah if I were able if I was able to configure this like that would have been a huge time saer yeah it's pretty easy to

configure like a lot of these tools have Auto instrumentation so if you're using something like the open AI SDK which you know or whatever or like the cloud SDK or whatever they will have like a wrapper around that that basically intercepts all the calls you're making and tries to instrument everything for you without you having to change that main lines of code interesting

okay yeah so so basically creating creating a system that already like handles much of the like abstractions and just tele so my advice is like you don't want to build a logging tool yourself I think you can build your front end to look at data that is worthwhile but you don't want to build like the like the logging and the observability yourself I don't think you know that's kind of too probably too far away you probably want

to focus on building your AI apps so yeah that's basically what I was doing with open plugin like for example whenever this happens I I just get I receive a log that is tagged with like failing I don't remember the specific details at this point but yeah not everything was basically just a it was a mongod DB collection that just like logged everything and so I needed to build like additional additional end points to like parse

through this which I think that that that could could have been abstracted by tools like blank lsmith and Phoenix right yeah take a look at those you know use one of them that you know like whichever one looks like it kind of fits your needs the most based upon like the languages you use and your preferences around like open source and closed Source or hosted and not hosted I

don't you know that's all a lot of stuff to navigate through but like you know you get an idea you can probably make a decision really fast like taking a look at these things and yeah and it's it it really helps and it helps you not only like look at your data you can build a lot of things on top of it because they have apis and stuff but it also help you when it comes time to

write evals because all of these tools have facilities for you to like write tests and execute those tests against data

that are in those in those databases like you can curate those traces you can like save them off as like test cases in a way and you can do things like iterate on the prompt and like see what the outcome of that is you know you can write tests and there's it's it's it kind of it's very

helpful right and when you're saying iterating on the iterating on the prompts of these test cases you do you mean like does it actually perform inference after you submit the yeah yeah so what you can do is like you can yeah you can basically use data from your traces as inputs to like inference so like you can have like your prompt and you can say okay like take this data set of these

traces where the you know and use like the inputs that you supplied in those traces to this like other prompt and you can test it out and you can see what the output is and you can do that in parallel and you can like keep iterating really fast to like kind of do they all have the concept of this like prompt playground thing basically has string templating that and the string templating pulls from the data that you

save are you describing a tool that you built or something no that's in all of these things if you adopt any of these tools that I'm talking about they'll have that so I'm just telling you built for LM yeah all of these are specifically built for llms yeah yeah they're all about llms so like they have all these other tools once you put your data in there they try to help you with like bunch of other things so it's not only

about a place to it's not just a glorified database is what I'm trying to tell you it's like a little bit more yeah yeah because that was one of the main issues that I was running into like when I was whenever I was running this agent action I would have it go from like zero to to the to the end and that just led to a lot of wasted wasted time and money so being able to just like directly pinpoint a

specific areas good yeah so the the the last thing I wanted to mention is something that I'm going to be working on now which is this concept of basically generating generating your own actions and saving them in a database that can be accessed by any other agent it's based on this paper dinosaur beyond beyond predefined actions and one of the big things that I see one of the big problems that I expect to run into here is that it is

very it should be like really unopinionated on how the users will end up using these actions so I I just I I don't know how I'm going to be able to effectively evaluate this when it's like so open-ended as to how users can use this because it's basically we're building an internal tool that's going to serve basically other agents okay so you're asking you have this component in your

llm based Automation and you're wondering like because the scope of it is so large you're wondering how you might evaluate it yes okay the best way to evaluate something like that is to do something called error analysis so error analysis I can show you one of my posts I can share my screen let me see if I can do that quick do I need to stop sharing I

don't I'm not sure I don't yeah go ahead and stop sharing it'll help with the recording so there's this post creating

LM as a judge that drives results quite a long post but there's a section in here called perform error analysis mhm and so the main point is hey like when you're doing evals there's so many things you can EV out right there's like so many components of your system there's so many things that your system

does like how do you even start how do you even know what to eval Persona yeah and so basically so what you want to do is ground what you're testing in the errors that you see mhm so you kind of want to go a little bit in reverse and say like okay I can EV out anything in the world Under the Sun but you don't have time for that so what you're going to do is you're going

to go through and start looking at data and classify any problems that you see so like basically so this is like a very there's a video on it this is a very old concept from machine learning there's one this video from Andrew Ang in the blog post is it in production that you're logging these errors or during during yeah during development I guess obvious okay so there's two approaches one is if you don't have production data

then you can generate synthetic data to try to simulate what a user might do and this blog post will kind of give you an idea of how you can try to simulate that data but you need some data to push it through either like a you need either need like fake humans or fake users or real users or whatever or real you know real interactions or fake it doesn't matter of course real is better but during development you may not have a

choice mhm but basically like what you want to do is say okay like you want to go through and it's very simple error analysis sounds like a very fancy term it's really not fancy it's like kind of dumb which is good it's very accessible what you do is like you go through your data and you start categorizing like what are the problems you see and you just and you like maybe you look at I don't know 50 examples

and you start like making a note okay this is the problem I see this is the problem I see and then when you're done with the 50 examples you just add it up and you kind of see okay like what is the what are the largest problems I see and you can like just do something like this like this is just a very toy example like okay most of my problems are like around user education or you

know what my internal tool has some authentication or access issue and then you can just just ground it in whatever the errors are that's that's what I would do okay and this is like potentially something you could determine after running the like feature Persona scenario ablations yeah and like you know don't you don't feel like you have to go take you know to follow this blog post so closely I mean do it make sense for

you this is just like an example of like to get you started thinking about how you can simulate like your data maybe there's maybe there's something that is not a feature scenario Persona maybe there like some other dimension that makes sense right but I was just trying to give people an example like okay this is how you can think about it you know when you yeah the the feature scario Persona is something that really stick with like just getting yourself thinking

about the different use cases of how a program might be used yeah seems pretty useful I'm definitely going to be just playing around with that yeah I mean but certainly start with your start I mean you should be using it too like this thing you should be trying to use it yourself and hopefully you are generating some data yourself as you use your own tools yeah you should certainly start there as some intuition so like these blog posts

that I have like they say okay you can't you you shouldn't rely on Vibe checks but you should certainly you should start with some Vibe checks you just can't that will only take you so far and at some point you will need to do this kind of simulation and generate data or whatever if you're trying to make it robust before you let's say release it to the outside world you know but before that like it is good you

should be using yourself and seeing what it's doing and you know that's really important yeah yeah what I was the the experience I had with the sales development representative agent was that it everything I was doing was just basically Vibe checks that's how I was evaluating everything and just over time gradually I ended up building out some sort of just like internal API internal endpoints to better be able to

understand things and to save myself money and time but I I feel like maybe if I used a tool that was you know designed for that it might have saved me a good amount of time but at that point I don't know like when when when should I like switch over to this tool as opposed to just building my own tooling like what's the threshold for for yeah it's hard to know that it's hard to know

that I would say like the logging stuff you definitely want to go to your tools U you don't definitely want to use tools but then you definitely want to build tools to look at data it's actually a presentation that you may find helpful let me share my screen again it's like let me see if I can find it so yeah if you Google haml AI engineer evales there's this short presentation like 15 minutes or so that

kind of like I talk a lot about okay like what tools to use like what to buy what what to build how to think about evals it sounds like I don't know I have an intuition that you might find it useful like this specific video you know I'll talk about like different kinds of tests you know like I talk about logging your traces kind of like what we just talked about and I'll say like you know like look at your traces but build

your tools like you know in this case if you need it yeah as I'm like as I as I read through your article and as you're like sharing these these ideas I just constantly keep thinking like H there's so many there's so many opportunities to just like automate automate this work and like create workflows between like for example yourself and your expert domain expert to just better analyze llms and it's so tempting to just like go immerse yourself in like

building these additional tools as opposed to like focusing on the product itself yeah I mean it is tempting I will say there's a lot of people building tools in that space like it's not we are not the only you know it is an extremely crowded area basically everyone yeah it's it has occurred to a lot of people let's put it that way yeah okay yeah that that's a that's good amount good amount of con

resources to work with and ideas to play around with and again I appreciate I appreciate your time and if it's okay I'll let you know how things are going or yeah please send me an email and let me know or get in touch and let me know how it goes really appreciate it yeah I appreciate it