LLM Eval Office Hours #1: Multi-Turn Chat Evals

20 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=4X4FCKNI4VW
https://www.youtube.com/watch?v=4X4fCknI4vw

SUMMARY

Max discusses the development of Windmill, a company aimed at automating management tasks through tools that facilitate peer feedback and weekly focus updates. The conversation delves into strategies for improving multi-turn chat evaluations, the importance of error analysis, and considerations for building versus buying evaluation tools.

- *Windmill automates management chores, focusing on peer feedback and weekly focus updates via a Slack agent.*
- *The need for robust evaluations in multi-turn chat conversations is emphasized, along with the importance of error analysis.*
- *Error analysis involves categorizing and understanding feedback to improve chat interactions.*
- *The discussion highlights the value of aligning AI judges with human evaluators to ensure consistency in assessments.*
- *Considerations for building internal tools versus using third-party vendors for evaluation and observability are explored.*
- *The conversation touches on the importance of prompt management and versioning, weighing the pros and cons of keeping prompts in the codebase versus external systems.*
- *Recommendations for vendors like Human Loop and Brain Trust are provided, emphasizing their strengths in evaluation frameworks.*

hello hello Max yeah I'm good to meet you yeah nice to meet you welcome to the office hours thanks for having me yeah so let's get right into it yeah so so so we're building company called windmill which we're trying to help managers automate a lot of the kind of what we call chores that are traditionally associated with management so there's two main things we're doing today one is we help you collect peer feedback so we

see who's working together and then we have a slack agent that reaches out I can show I can share my screen quickly if we go to it will reach out and basically say hey how's it been working with this other person about a specific work item then it has a little conversation turns that into feedback and then make sure that gets back to the person so that they they know where they're strong or where their are is to improve another thing we do

is every week on Fridays we reach out and have a little conversation about basically where' you focus your attention this week it already knows a lot of information about where the person is spending their time because we hook up to all the different productivity tools but they can kind of go in and be like okay here's what I'm focused on and then you can imagine for a manager they can get a better sense of where people are spending their

time what they're focused on or they're focused on the right things without having to set up a ton of meetings

and stuff like that so overall we're trying to make it a lot easier for managers to and automate a bunch of the stuff that is usually pretty timec consuming and increasingly we're deploying these kind of via an agent that works with your team so this we call it Wendy Wendy will go out to everyone on your team and ask them

questions or get feedback or stuff like that at different times of the day and so we're going a lot deeper we've using LMS for a variety of things but increasingly we're going a lot deeper into these little chat conversations primarily inside of slack and we're now at a scale where we've been kind of just eyeballing and improving prompts by hand and that was working pretty well but now the prompts are pretty good and we need some more

robust evaluations specifically in the context of like a multi-turn chat conversation and so we're currently looking at we have some in-house tooling which I can show you and we're looking at some vendors I don't know like human Loop and Langs Smiths and those kind of people and just would love some guidance on like where you think we should be started what are the right areas to focus on specifically in this kind of multi-turn chat type of

approach yeah that's kind of some background yeah so okay like multi-turn chat is very open-ended you know you could potentially evaluate infinite number of things and a lot of people get stuck at this phase when they're like okay where do I start with evals very understandable and so the counterintuitive thing is like one thing that's really good to do is to do an error analysis first and so what that

means is you gather your data like you know and whatever way that you collect it in a way that you can read a lot of examples do you if I record this Sor yeah it is being recorded on my side but and I'm going to post okay cool so essentially what's going to happen is like you do an err analysis and an a analysis really is and let me I can share my screen to give you kind of

some pointers let me one second how do I am I still sharing I think it'll still let me share so yeah you you you won't block me so no worries is it can you see my screen hold up I cannot wait what's going on

Zoom oh here we go here we go no I can oh here so my screen let me close out okay cool okay so what you want to do is like you want to like perform an error analysis and basically what you want to do is like you want to look at your data there's this video by Andrew Ang that kind of tells you what it is it's it sounds a lot fancier than what it is basically what you do is

like you look at a bunch of data and start seeing like kind of categorizing by hand like any errors that you see just roughly okay it's nothing nothing fancy at all it's like very like like very manual in a way yeah so we just did one of the things our users can do is basically saying like hey I don't have any feedback which is usually a sign now like it wasn't a good question why'd you ask about this thing and so we just went

through that and did an analysis of all the different reasons why people essentially dismissed the chat so was

kind of similar we didn't do every single chat we did the ones where it was negative and so we kind of have a good sense of like we have a bunch of examples that are not good but we kind of as we it's kind of you you get into that wacko game where you fix one and then other stuff gets worse and it's

just at the point where we need something more automated so but yeah Sor I didn't want to interrupt yeah no worries okay so if you already done it seems like you've already done some air analysis which is great so what you should do is like Focus your tests on those errors yep there's a lot of ways to go about it one is like okay if you can narrow down like the problem and kind of figure out like

why what's triggering it like oh okay when the user is asking about XYZ it's always doing this or it's always like giving like boilerplate answers and or something I don't know I'm just making it up what you want to do is like start you know crafting evaluations for that now it's a little bit of a like back and forth It's not like there's no like linear workflow through this so okay like first I just want to

say like if you find something that's like obviously really wrong all the time just go fix it don't get too don't get nerd sniped by evals like in the extreme sense yep so like after you've kind of solved like the really like obvious stuff they you're like okay let me just go fix that right now so it's like whatever then you can start thinking about okay you know can you write tests for those and like okay how do you write tests for

something that's like multi- turn conversation you don't like it's subjective maybe like you can't necessarily write an assertion for it like and so that's when we get into LM as a judge potentially and that's when you want to like sort of see if you can craft an LM as a judge that matches your opinion of things yep so that's what this blog post kind of

walks you through in in a way I mean this blog post is one attempt at describing a possible workflow through doing that so don't yeah don't like just take it with a grain of Sal but essentially like what you want to do is you want to write down specific critiques of like what is going wrong like you take like you drill into one of those categories of the error analysis you write down like okay what went wrong

like exactly once you have like 10 of those or so you can try to see if you can go update like craft a prompt for LM as a judge what we want to do is want to go back and forth and observe like the agreement between you and the LM as a judge until you bring it into like a high amount of agreement so you want basically prove that Ken yeah so so so the first thing

is to make sure that the LM match you as a judge like a human judge like can can we treat can we teach it to make sure that it can evaluate okay so like can we trusted as a judge is the first thing to try to figure out and and the way this makes sense okay yep I got it yeah yeah that's the key part people Miss they're like they kind of just throw the LMS a judge

they write their prompt and then they go about their day but then if you don't do the exercise where you try to

align it with you and measure it yeah then you can't ever trust it because you don't really know if it's like do people run into issues where the just the judge is not good enough like generally can you get it to work or I mean maybe that's too vague of a question but so the

problem is yeah like this is a very interesting question every time I've done this exercise with people LM as a judge people change what they're looking for significantly because like the process of like doing it changes your mind about what should be done like it's like if you're G to write critiques it like forces you to like really think about it

yeah and every single time 100% of the time people are like oh you know what like actually it should be like something else so just going through the process is sometimes enough to help improve yes yeah yeah it's the joke it's like the inside joke amongst like like U ml or AI folks is like it's kind of like you go through this process but it's really the process that actually ends up benefiting you more so than even the L as a judge

because it's like just forcing you it's just like forcing you to look at the data very very closely yeah and it's much easier it's much easier I mean the problem we have is like a little tricky where it's it's definitely easier to critique than to actually write like the perfect like given ex result you can critique it but TR but obviously the hard part is trying to write that so okay this makes sense okay so so so let's say we do

that and we have a judge that's working well and there's some nuances in here so like one is like you know try to make binary decisions about if it's good or bad and then like write a critique of why it's good or bad like try to distill it don't don't be wishy-washy like oh it was like kind of good here but it was kind of bad here and maybe and blah all that stuff is like you not it's not actionable and

you're not going to go anywhere with that you kind of have to draw some line in the S like is this possible or not yes or no and you want binary over like multi like a I don't know one through five or something bin yeah absolutely 100% because if you get a one to five score what are you going to do with that like if you get a three like what do you that's not going to mean anything to you

over a four you're going to be like okay like whatever yeah okay yeah that makes sense and basically like when it comes to the you know what you want to do is like you want to put examples in the prompt as well so like there's some you know exam like you want to put examples of the your critiques in the judge because what you want to do is like you want to have your judge use Chain of Thought to do the

judging and like your critiques that you wrote are your Chain of Thought So you want to train it to think like you does that sense yeah yeah we're doing this in like all of our prompts now very similar pattern it's pretty effective so this makes sense so yeah okay so that that's kind of like how you can think start thinking about this like you know this eval for for that case yep and then in terms of okay so

so we have that evil what about in terms of like the maybe you're saying don't worry about this for now but like

in terms of like the tooling and the infrastructure to be able to we're basically trying to decide around building stuff in house versus like moving our like right now all the prompts are in the codebase and all version of the codebase that kind of setup and there's obviously a lot of vendors that promise good stuff if you

move the prompts to their system and use their evals and logs and observability Etc and those are definitely problems we're facing but we have a question around like do we continue to build stuff ourselves do we try to use some sort of like an open source eval framework do that we kind of just plug into our tell me a little bit about your text that are you python typescript what kind of are you have like machine

learning like data science type people more developer type give me a little bit like just few P stack Engineers all typescript okay small team not not really trying to solve the problem where we have like different people doing the prompting versus doing the actual like engineering stuff more just like we're struggling with are you using any Frameworks internally like Lang chain or something okay no so we call we call the we have a we have our own like homebuilt rapper

around so we can easily like swap out CLA and open Ai and most of the stuff we're using Sonic 35 for and so we have a little bit of logging and tooling that we built in house but pretty much just making API calls directly pretty raw okay are you using anything already are you just shopping at the moment okay we're looking we're looking at some some different vendors and I don't know if it's yeah we we need to figure we need to get

better at this because we're as I mentioned like we're hitting the wall of so like I think the setup I want to start to have is where we have a nice eval system and then when we're doing prompt engineering we can kind of quickly iterate on prompts and see how those are changing and then also there's some sort of probably like always running thing that's looking at logs and doing sem vals and then maybe some cicd like system that kind of like

runs before deploying also running some like EV bows so something I mean that's like high level but some stuff like that we're trying to figure out infrastructure okay so and is it are you okay with paying for tools are yeah yeah we're okay paying so those the ones that you mentioned are all very decent they all do those things you know so like my favorite vendor so far human Loop is really good as well so I guess for you since you're like a

typescript shop and you're like mostly full stack developers you know like Brain Trust is worth looking at I don't know bra yeah yeah definitely take a look at them take a look at human Loop is Brain Trust your favorite you would say I like them a lot they're they're pretty good yeah it's hard to like have a favorite honestly

there's like different things like for different purposes like are they're kind of like people different good at different things but like it's definitely you should definitely take a look at both of them get a feel for like what you think resonates with you the most yep that's that's the one I would add to the mix okay we looked at Lang fuse Lang Smith so L brand trust yeah is is cool I mean I really like Harrison so

like their team had amazing support like they're all over it but just add Brain Trust to the mix and kind of and

with the people you've worked with the people who have made the jump I'm guessing there have been people in our scenario where they're kind of some Half Baked internal tools and then they're like hey let's use a vendor like is that usually a good I mean i' in my experience so so yeah yeah like for

eval stuff you want to use a vendor for a lot of other other stuff I don't think you you should like I'm not a fan of Frameworks that much yep of like I agree you know llm Frameworks or like you know orchestrator type stuff but evals is things like I don't know it's kind of like observability and stuff like that you don't want to be building that you know it's probably really far away from your core thing definitely is yep and so

it's kind of like yeah I recommend just using like so I I don't know how related Lang chain and Lang Smith are I didn't have a great experience with Lang chain is Lang Smith like even if you're not using Lang chain that's still worth looking at you can use it you can use it with a lang without any Lang chain Lang graph stuff is to 100% compatible I would say like it's kind of the reverse is true if you are really bought into

Lang chain then just use Lang Smith because like the integration is going to be like best in class so like but if you're not it's kind of like maybe it's worth like getting a look at at these like maybe these three that you pointed out y just because like also they keep changing they're like kind of so just and you'll like you'll like one more than the others do you have an opinion on where to store prompts because

they all want you to store prompts in their framework and version them there or at least the ones we've looked at and use their prompt templating system which obviously I understand how it can make all the tooling they have a lot better because they have the template itself you don't have like you don't have to I know like I was looking at Lang fuse the other day and they have like a prompt management system versioning and stuff

I would say it's up to you there's some there's some drawbacks to doing to having your prompt in a separate thing that is not a part of your code base yeah and you need to reason about how you feel about that like do you need lots of non-developers iterating on your prompts maybe the answer is no and then if the answer is no maybe just leave it in your code base because once you have something outside your code base then

you have to deal with all then you have a sh you know you know how it is you basically are having some code outside your code base in a way y yep yep yep okay that all that all makes sense and you see most people versioning the prompts alongside their just like General code versioning or are they having like a whole separate scheme where you're kind of storing multiple versions of the prompts and using some sort of a system to kind of like I don't

know roll stuff out some people are so some people are versioning it like separately especially people that have like domain experts and non-developers like they need toate on it I will say that like when you get okay like a lot of times your prompts don't live in a silo so like they need access to application all yeah they need to like query a database they need to do something and so it can be a little bit

difficult to house The Prompt like to test the proms outside your codebase yep and you kind of have to like design a data set that can like be templated in and when you like you know it's get a little bit tricky so you there's a trade-off there but yep yeah you have to think through that okay all right all super helpful thanks for taking the time yeah if this was helpful for you please I'll send you a email with

like a feedback just let me know you know how awesome it could be more helpful no it's super helpful I appreciate you taking the time just make sure I didn't share anything sensitive before you post yeah if there is let me know yeah I don't know I don't know exactly what I I guess the I showed you slack so I don't know what what was in there so okay I can edit that bit out or if

it's yeah I mean if it's just I don't know if I can see it before I can tell you it's probably okay I'll send you I can just send you the video yeah send like privately and then you can let me know okay perfect sweet great talking all right thank you all right thanks bye