# LLM Eval Office Hours #3: The Importance Of Starting With Error Analysis

24 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=ZEVXVYY17YS
https://www.youtube.com/watch?v=ZEvXvyY17Ys

SUMMARY

*The discussion revolves around the development of an SMS-based caregiving app that utilizes various technologies, including Twilio and Azure AI. The speaker is focused on enhancing user experience for unpaid caregivers and is currently in the prototyping phase, gathering feedback from beta users to identify areas for improvement and evaluation.*

- *The app is designed for unpaid caregivers, addressing self-care and memory retention.*
- *Current features focus on basic self-care suggestions, with plans for more sophisticated functionalities in the future.*
- *The speaker has 10 beta users providing feedback, but engagement is challenging due to the SMS format.*
- *Evaluation methods discussed include error analysis and user interaction categorization to identify common issues.*
- *The importance of a bottom-up approach to development is emphasized, suggesting that user data should guide feature prioritization.*
- *The conversation highlights the distinction between unit tests and user-driven evaluations, advocating for the latter based on observed user interactions.*
- *The speaker is encouraged to start small with synthetic data generation and gradually expand as insights are gained.*
- *Overall, the focus is on leveraging user feedback to drive improvements rather than adhering strictly to preconceived notions of app functionality.*

I I prepared a few things just to make sure you are well apprised so I'm going to share this I think it's just the briefest of like slides and we can look at some code and stuff like that just so you have context I think definitely the app is a sms-based let me get this out of the way this maybe

bigger sms-based caregiving app so if you can kind of see the architecture here twilio to fast API is a krant Vector store m0o for memory Azure open Ai and helicone for observability so all those kind of pieces the personas that I'm working with our early stage steady state and

kind of day-to-day caregivers eventually I'll be looking at something that is a little bit more sophisticated so it's a little bit more basic right now features are very much focused on self-care I think of this middle section really as kind of memory and retention around that so recall and you know since self care is a big component of that there is

you know in built in the system prompt a kind of a Reflection Point so it always kind of folds back on that and there's a bit around like it's not about any medical advice I just laid out some B2 features because I know they're coming and they get a little bit more sophisticated and so I'm just trying to figure out where they fit because they get into like structured output rag type stuff but not not important for now I don't

think scenarios you know someone comes in looking for suggestions so you know the format I tried to think about here is you know what they like you know a t typical kind of user story and then what the app does the expected app behavior so I went through a few of those and then same thing with like B2 expect did ones I wrote out a couple test cases

and I think this is one of those places where you know yeah like what is the caregiver is it like a doctor is it any kind of caregiver of any kind yeah yeah I probably should have probably explained that yeah this is an unpaid caregiver so this is like you taking care of your mother you taking care of your father or this is a parent taking care of like a disabled child gotcha so and that represents about like

50 million Americans so it's not a small T test cases here I'm trying to think about you know again like you know when I'm trying to wrap my header on evals I think the biggest stumbling block I'm running into as a multi-turn conver ation experience with memory I'm not even trying to get into like the the grounding because it's you know maybe medical and things

like that but the multi-turn bit and the memory bit are the things that I'm tripping up over maybe I'm over complicating it but this is where I was just trying to get to the most simple evaluative evaluative heris that I could you know in terms of you know taking some of the the sample user data that I have so I have 10 BET users that have been using it for the past like two months three months and the expected behavior and

then you know how I would kind of creade it so I wrote that up for five cases and then last night because you know I I reread your two blog posts I wrote up some unit tests I haven't run them but this is also where and we can look at them if if it's helpful but you know again this is like you know trying to figure out if I'm if I'm doing these things right

okay yeah that's a good question so some unit tests can I look can we go back like slide eight or nine just want just look at it one more time okay so you have like some of these test cases and okay have some scenarios or I see passive supportive okay so you have some ideas can we look at slide nine just want to remind myself yeah

system provides General selfcare okay go to slide 10 again okay okay yeah okay so this is all using you know not it doesn't go out it doesn't hit any apis right I ran it once and you know I think thankfully right it failed a few so like it gives me something to work with and and then we we'll we'll I'll hop into to cursor in a second right and

then the last thing I want to show you here is in Azure AI Foundry I exported out from helicone my data set which was like 223 you know Rose and I imported that into Azure AI Foundry they have an ebal thing which I

think is very similar to what open AI has and then I I ran it through and I know this is one of those

probably I shouldn't have but like it was what I had provisioned so I just ran with it yeah so like I had I used the same model you know 40 as my judge and I know you typically write use a smarter judge a SM smarter model as a judge or something like that but other than that I set the same parameters and then it ran through it does a you know a standard set I think

you can also set some other criteria and you know there's a CSV that that accompanies this as well yeah I mean I have some questions but then you can just yeah thanks for preparing those yeah can you so is so before you get into like these evals and stuff Are there specific failure modes

where you at in the app are like are you kind of still building it still prototyping it do you have anyone using it already what's St yeah I have 10 10 beta users okay that are that are using it on a on a semi-regular basis right like it's a it's SMS so it's very hard to kind of keep them kind of in the loop because I don't have anything that is I don't have any kind of push out

to them so it's very reliant on them picking up and using it okay and then that's a common thing yeah like lot of people SMS route makes sense and I'm building out a separate like production version which is slightly different but but yeah okay and The evals that you think you're thinking of or that you have right now like are they based on failures that you're seeing from

those users not exactly I think they're they just represent really more of I think the the general criteria that I I think that the Apper experience should adhere to right are you looking at that data from the users like their interactions and yeah kind of seeing how it's going yeah and that's that's where some of the you know I think that's where some of

this I think came from and when I was looking at it as well so I and I know some of like I think the easiest stuff to pull out is you know and I think what I do like about this is the the coherence and the fluency bit or the coherence and the relevance because this is conversational is how friendly because that's how you know I think a big feature right

have you looked have you looked at the individual conversations and yeah have you been have you done error analysis yet on those conversations to try to categorize like what where where they might be failing no I wouldn't even know how to kind of like Define that yeah okay that's good yeah I think that's one thing that's like being surfaced in the office hours actually that is interesting is like a lot of

people they don't know where to start with the evaluations there's a lot of different things you can evaluate you can have all these different metrics you can come up with a lot of different tests you can come up with an infinite number of tests you can brainstorm all kinds of stuff and then ultimately like that you know that can you don't want to just keep doing that because yeah you need to develop your application so like you have to

prioritize what you're evaluating based on something just because it's an infinite space and the way you prioritize that is okay if you have users if you have a fair number of users you have 10 hopefully you're getting some you know consistent data you know because you have usage and what you do is like you read every single like the order of 10 users I would say you can read everything at

this point which is fine yes yeah and you know is to go through each interaction very carefully and just categorize like what are the different kinds of Errors you're seeing and you can do that in like just think of it's like in a in a dumb way like it doesn't have to be anything fancy it could just be in a spreadsheet where you just add columns for different categories and you just put in X like oh

okay this is that this kind of error it's not retrieving the right context it's not friendly enough whatever what whatever and then after you look at let's say I don't know 50 interactions you will know you can say like oh you know what like this is my biggest issue here this is like the most important thing you might even need to get to 50 like you might get through 25 and it would just be extremely obvious that like this is one

thing that you need to do so when you first start doing that the first phase is like you don't even need to do evals maybe because you will see something like so obvious that you're just like okay you know what I'm just going to fix it like like forget evals let me just work on that right now but then like after you get to some point where it's like okay you kind of fix some of the bigger things then you

have like some other things that are like a little bit more difficult you're like not sure how to fix those are like that you know you kind of plucked some low hanging fruit then you can like focus on creating EV vals for that and like kind of let that help you prioritize I mean what you're doing is okay like it's okay to have ideas of like stuff you want to test for especially if like those are important

to you but let it if you can let it be driven by what you're seeing that's wrong and that will like that will influence everything else like you know like CU this coherence and fluency groundedness all this stuff like it may not mean anything to you you know like I don't know what this means like if you get a score of three.72 tomorrow you get a score of 4 4.2 yeah does it really mean that it's

better we don't know that's the problem with this you know and if you you don't want to waste time on this because like you have limited time and so that's where it should start okay so like even more I start with like the multi-turn bit of the is a non-issue right multi-turn is fine yeah I mean like multi-turn is yeah there's nothing wrong with multi-turn in fact most applications that I work with are

multi-turn but I mean like evaluate it because like each you know conversation response is a row so just evaluate each of those and see how and judge those on an individual Bas when you the error analysis yeah okay so when you do the error analysis you're really looking at it as a whole yeah does this session I'm just gonna say session yeah like a trace right yeah kind of like a s I know it's like SMS so it's like what

is a session you have to have some way of kind of breaking up things into sessions you know maybe it's like some time cut off I mean there's always there's some logical aspect to it like yeah but like okay in a given session there might be X number of turns did you know did your assistant accomplish what you wanted it to with the user and like if not if you're not doing it in the right way we need to like categorize

that and you look at it like more holistically yeah and then based on the failure that you see like oh okay it's not has not retrieved the right context or it's just you know or it's like doing something dumb like emitting user IDs into the output or doing something other it's like any kind of era you see then we can decide like how you evaluate that like how you write test for that sometimes you have to look at the whole

session yeah to kind of evaluate it sometimes you can do something depending on the error you can like you know look for a certain message or only look at like the last message or first M you know depending on the error you know kind of drives how you test it see I think what I'm hearing this is really fundamentally like a Bottoms Up type of a thing as opposed to like a like a top down right like I should look

at what's there right as the basis to determine what is you know what are the areas to build around as opposed to come in with a preconceived notion right of what I believe to be you know the the core features or experience of the app and then shape it around that or is it kind of like bring those two together excellent

question I so different people have different flavors that they like I really like I'm I'm kind of a little bit more bias towards the bottoms up like you said yeah there's a way of being a little bit more top down which is instead of writing the tests up front for what you want to the behavior you want to be simulate that behavior that you want

to test and let it appear in the data so that you can then decide is this worth testing because what happens is you have this idea of what you want to test in your head yeah however it's often not it's only until you can see something and react to it that refines the idea of even what it is you want a test like it's easy to like write a test but then so like you have to in my opinion you kind of have to see

it so you you know like if some of these things are important like the things that you kind of outlined on slide eight and nine you know you can try to simulate those user interactions with an llm and then you can like then you can see so this is the go down a little bit of a synthetic data and then simulation route and then you know determine like is are those you

know scenarios and evaluation criteria useful in you know the feature set in the app right like that type of yeah absolutely because it give you something tangible to work with yeah because hey if you want to do if you want to accomplish those goals that you laid out in slide eight and nine yeah you know if you generate synthetic data that test those goals then you have something to work with and iterate on yeah you know

it's like make those intera like make those interactions work well in in response to those synthetic users is there I

mean have you maybe you've written about it like have you written about just general good guidance on like number of runs or how to generally approach like that from just you know I guess starting right in which part the synthetic users or synthe data or which which

aspect I guess both parts right like setting up the synthetic users the simulation to arrive at like your you know a conclusion presumably right like on yeah okay oh yeah this is a good question everyone asked this question and just just start small you don't have to have a set number in mind people really stress out about this they're like okay how many should I look at

should I look at a thousand should I look at 100 I don't have time to look at a 100 that's not going to be fun blah blah blah you know just look at just start looking at it yeah you will get the heris stic is keep looking at data until you feel like you're not learning anything new and what that means is either a when you look at data you're going to see like one glaring problem

that overshadows every other problem and then you'll know okay let's just stop looking at data and let's work on this problem or as you're looking you'll be like wow okay like this is I'm getting like really new information as I look at each row of data or each instance of data let me keep going a little bit and then you reach a point where you're like oh I kind of feel like I kind of have an

idea and you can follow some of your intuition and it doesn't often have to be that much data and then this like far as synthetic generation just start really simple like just generate like one or two test cases there you know for each of these things and you may decide when you look at it you may look at it and say okay you know just iterate like slowly from there and then you can just generate a little bit more and more

but don't make it overwhelming you there's some amount of faith I guess you have to have I mean but it's surprising like when you start looking at data you you get it's valuable much faster than you think yeah no I it makes perfect sense so I think that that's helpful and I think that I think helps me I think that eases I think a lot of anxiety I had just

you know thinking about just the the you know when I when I hear evals around you know what I'm doing right so approaching it that way I think makes it much more approachable which is I think good and just so I'm super clear right it's probably the dumb question right that I'll ask which is these what we're talking about now this is distinct from unit test assertions

right yeah I mean it's distinct it's going to be like so the the errors you see will motivate you to write the errors and assertions don't write the error and assertions because you're trying to eat your vegetables and you're trying to be do some kind of thing that I wrote about you know when you look at the data you're going to be you know you'll find somebody like oh my goodness I want to this error I want to make sure that I

know about it yeah and it'll kind of like you'll want to write the test like this this error keeps me up at night or

like I don't want users to go through see this you know this is embarrassing let me write a test if you do it in advance you're just going to it's yeah you have to go the other way yeah no I think that that's helpful I mean some it was also helpful for me to do it this way too

because now I know not to right yeah yeah no I mean I think yeah I think people this is not uncommon like that people have skipped a step and have started WR doing evals like just by writing tests yeah so I also have to maybe write something on this or you know let people know

yeah I mean I think that that that kind really the whole thing the whole eval thing it's like the most important thing is the looking at data is like it's almost like you will you will understand like even the eval the test themselves is like pales in comparison to what you will learn from looking at some data it seems like unreasonable like oh it's such a clerical task or feels like okay why am

I look I'm going like look at like logs but yeah it's like really the highest leverage thing you can do yeah I mean eval is is a second order you a bit from from it so yeah look at the first order yeah yeah great yeah let me know how it goes you know after you kind of start looking at stuff let me know let me know how you how you go be really interested to to

hear about your progress yeah really appreciate it thank you so much yeah thank you