Corrections + Few Shot Examples (Part 1) | LangSmith Evaluations

15 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=3GCTA0LI4EW
https://www.youtube.com/watch?v=3gCTa0Li4ew

SUMMARY

The discussion focuses on enhancing LangChain's evaluator system by incorporating human feedback to improve the accuracy of scores generated by language model (LM) judges. This is particularly relevant in applications like retrieval-augmented generation (RAG), where precision and recall evaluations are crucial for assessing document relevance.

- *Users desire the ability to modify evaluator scores, especially when using LMs as judges.*
- *Human feedback can correct LM mistakes, ensuring nuanced preferences are captured.*
- *The setup involves creating evaluators for precision and recall checks on retrieved documents.*
- *Incorporating corrections into the evaluator's training can improve its scoring accuracy.*
- *The process includes defining inputs and outputs for the evaluators and using few-shot examples for calibration.*
- *Feedback from users can be integrated into the evaluator's prompt, enhancing its performance over time.*
- *The ability to adjust scores based on human input is a powerful tool for refining LM judge evaluators.*
- *This approach is particularly beneficial in applications where precise scoring is critical, such as RAG.*

hey this is L Lang chain so we've been talking about a lot about Langs Smith evaluations and recall the four major pieces you have a data set you have some application trying to evaluate you have some evaluator and then you have a score now one of the things that we've seen very consistently is that users want the ability to modify or to correct scores from the evaluator now this is most true in cases where we talked about this quite a bit

previously you have an llm as a judge as your evaluator we kind of talked about different types of evaluators you can use human feedback you can use heris evaluators LMS judges is one in particular that's very popular for things like Rag and really any kind of like string to- string comparisons LM as judges is really effective there's a lot of great papers on this but we know that LMS can make mistakes and in particular in the case of judging you know often

times they may not capture exactly the nuanced kind of preferences that we as humans kind of want to encode in them so we're going to talk today about a few ways you can actually correct this and incorporate human feedback into your evaluator flow now I'll show you one of the most popular applications for LMS judge evaluators is rag so if we go down here we can look at remember there's a bunch of different ways to use LMS as

judges within a rag pipeline you can evaluate the final answers you can evaluate the documents themselves you can evaluate you know answer hallucination R to the documents so we're going to show today about how we can

set up an online evaluator for ragot that will do document grading and then how we can actually use corrections to improve it so I'm going to go over to notebook I'm going to create a rag bot here so you know I'm just going to index

a few blog posts so that all ran and here's my rag bot it's super simple doesn't even use Lang chain this is just kind of rawad GPD 40 I'm doing a document retrieval step I set my system prom up here this is like a standard rag prompt your helpful assistant use the following docs to answer the question that's all we're going to do here and let's go ahead and run that once on a simple kind of input question about the

react agent so this is a good question asked because one of our documents in particular this guy right here is a blog post about agents so that went ahead and ran now we go to lsmith and we can see we have a new project with one Trace here we go we can look at the trace we can see it contains retrieve docs invoke our llm so that's great now let's say I want to build an evaluator for this project so I can go

to add rules I'm going to call this recall I want to I want to perform a recall check on the retrieve documents I go to online evaluator create evaluator look at our suggested prompts and I see this one for document relevance recalls this pretty nice and I'm going to go ahead and use GPD 40 it's better llm for grading and you're going to see a few things here that are kind of nice so this is basically setting up an

evaluator that'll run every time my project runs and this is really nice if you have a you know an app in production and you want to for example greatest responses flag things you know that are egregiously wrong and so forth so what I'm going to do is this mapping allows me to take the outputs of my chain and map it into the inputs for this prompt so it's pretty cool so I can see my chain has these two outputs answering

context context just the restri docs which I pass through so there we go question just input question so now my two my inputs and outputs are defined they map into my prompt right here so facts question now what's going to happen here in the prompt this is going to be grading relevance recall and so you know I'm going to give a question I'm giving it a set of facts and basic basically I'm asking a score of one says any of the facts is relevant

to the question so again this is kind of recall test so recall just basically means is do the documents contain facts that are relevant to my question now it can include lots of things that are not relevant but as long as there's a kernel of relevance I'm going to score that as one so that's the main idea here and now I'm going to do something that is kind of nice I'm going to use this corrections as few shot example so this

is going to create a placeholder for for me that I'm going to use later after I correct my evaluator and so what this does this basically just sets up a placeholder right here that contains I'll call this this is basically a set of facts question reasoning and score so what's going on here well fact and question are just two of the things that are input to my prompt so that's kind of these are going to be provided by the user

reasoning is going to be an explanation for the corrected score that I'm going to give it so these are going to

come from the human feedback and this is going to be basically a few shot example that I'm going to tell the grader to use in its consideration of the score so what I'm going to do I can go ahead up here and I can say here are several examples and explanations to caliber R your

scoring so what's really happening here if we kind of step back is I'm basically creating a placeholder where I can incorporate human Corrections into my prompt that's really all that's going on here and what's really nice is so this is all here I can go ahead and name this output score recall let's just change this just so it's a little bit easier in our logging later on now what I can do is I can use this preview button

to see how this is going to look and make sure everything works so this is pretty cool this just injects an example kind of facts question reasoning score so this is this is just like a placeholder for what the user will input these of course will be input by me later and then here is the actual input for this particular chain example so this is just confirming that everything's hooked together correctly so cool and I'm going to go ahead and

continue so that's our recall grader I'm going to save that let's add one more we'll call this Precision great I'm going to say online evaluator I'm going to use GPD 40 again try a suggested prompt document relevance precision and again I'm just going to change my my key name for the output score I'll again use these few shot examples let's just kind of format these slightly based upon how we' like them to

look so this is nice good Precision question facts that's fine cool and again we'll just instruct here are some examples to calibrate your grading cool so that's all going on here and let's go ahead and do the final piece here so we have to hook up

basically here's our chain outputs context are the documents that retrieved here's the input question and we are done so this is our Precision grater and we're all set so now we have two graders attached to our project I go back to my project now I'm just going to mock let's say I'm a user playing with this app let's just go ahead and ask a couple questions so how does react agent work and I'll go ahead ask another a

few other relevant questions what's the difference between react and reflection approaches what are the types of agent memory or llm memory cool I'll just run these what is memory and retrieval model in the generative agent simulation run that cool so we've just gone ahead and ran a few different questions against our data set very nice so now what you're going to see is these are all now logged to our

project and over time we're going to see feedback rolling from our evaluator so again what's happened is I set up a project I've added two precision and recall document evaluators to my project these will grade the retrieve documents for precision and recall relative to the question and I'm just going to go ahead and let those evaluators run on my four new examp exle inputs okay great so we can see is that now I have online evaluator feedback

that's rolled in and these are against my four input questions this is pretty cool now what I can do is I can go

ahead and review so let's just kind of move this over and my outputs contain the answer and the retriev documents in these context so it's pretty nice says I can go ahead and quickly review this and say hey do I agree with my evaluator or not so let's look the question was react how's a react agent work and let's look

at my documents briefly I can even just copy over react maybe that could be kind of make it quick so I can see this first document those mention react twice so that's pretty nice yeah I think that's that's pretty reasonable the second one Yep this is definitely correct again third one looks right again talking about the react kind of action observation and thought

Loop and this final one does not actually mention react so in this particular case I look at the scoring the Precision is zero the Fe the recall is one I think that's about right because this fourth document does not mention react at all so I'm happy with this we close this down let's look at the second one in this particular case it's talking about react and reflection so two different approaches what's the difference right so here okay so the first one clearly talks

about react that's good the second one also clearly talk abouts react cool the third one does not but it mentions reflection okay so we can look at a little about at this last document now you can see that it mentions irco kind of a complimentary approach to rea Dimensions react it doesn't say too much about how it really works it says it combines coot prompting with queries in this case to Wikipedia

so you know I can be a little bit critical here and what I'm going to say is here I'm going to say the Precision is not one and here's how I'm going to do this so I can basically make a correction to my greater so I'm going to say okay I'm going to say the great is zero and when I'm going to tell it now this is really nice I can actually give it my feedback explicitly so what I'm

going to say is the final document does mention react but it doesn't actually discuss how react Works in any level of detail as opposed to the other docs which discuss the react

reasoning Loop more specifically cool so for this reason I do not give it a Precision score of one that's it so I go ahead and update that cool so now we basically said look I I consider this last docum be a little bit of false positive I get it says the word react

that's probably why it was retrieved it doesn't really talk about like the functioning of the react agent in in any level of detail and so you know again this is just an example the kind of feedback so we can go back to our evaluators we can look at the Precision evaluator we can go down and see this F shot data set now and see that actually includes our correction and our explanation let's have a look at the preview and see how

this going to be formatted so here are some examples of calate your grading here's a whole bunch of facts cool here is the question the final document does mention react but doesn't specifically discuss how it works so I do not give it a Precision I do not give it a Precision score of one Precision zero Okay cool so it looks like it sucked

in our feedback nicely it's now part of the few shot example so that's great that's included in

our valuation prompt now let's go ahead and check so let's rerun on this question and see if our example kind of was correctly captured by our evaluator great so we can see our evaluator just ran so again here was the question we asked what's the difference between react and reflection approaches are scoring now is precision zero recall one now look at the last time we asked

this question what's the difference between reaction reflection with our correction if you look at our scoring here Precision we corrected it to be zero we provide that as feedback now the evaluator is correctly calibrated it's course it is zero and one of course look this is a case of overfitting I completely understand that we've literally added this particular example to the F shot prompt but it's a case where we can actually highlight that performing feedback and Corrections

rolling that into your evaluator few shot examples can actually correctly calibrate it to produce scores that are more align with what you want so it's a really useful tool for building LM judge evaluators that adhere to the type of kind of scoring rubric that you actually want and this is really useful because oftentimes It's tricky to actually just prompt it to produce the correct kind of scoring giving it specific examples from Human Corrections is really powerful powerful and so that's the big

Insight here it's a really useful feature and particularly because elements judge Valu which are so effective and they're you know increasingly widely used the ability to incorporate feedback really easily is is just a really powerful and nice tool so encourage you to play with it thanks