Corrections + Few Shot Examples (Part 2) | LangSmith Evaluations

16 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=FML6CB5Q5M0
https://www.youtube.com/watch?v=fmL6cB5Q5M0

SUMMARY

Lance from LangChain discusses the evaluation process for applications using large language models (LLMs) as judges. He explains how to align LLM evaluators with human preferences through a correction flow, allowing for improved scoring of application outputs based on user feedback.

- *Evaluation involves four components: application, dataset, evaluator, and scoring.*
- *LLMs are effective at judging natural language outputs, but require careful prompting and auditing.*
- *A new approach allows capturing human corrections as few-shot examples to improve LLM evaluators.*
- *Demonstrated using a RAG (Retrieval-Augmented Generation) bot with a dataset of question-answer pairs.*
- *Two evaluators for document recall and precision can be added to enhance evaluation accuracy.*
- *Feedback from evaluations can be used to calibrate LLM outputs, improving future assessments.*
- *The process allows for both online and offline evaluations, enhancing customization of LLM judges.*
- *The integration of user feedback into the evaluation process significantly enhances the effectiveness of LLMs as evaluators.*

hey this is Lance from Lang chain so we've been talking a lot about evaluation and there's really four components we've been discussing there is your application itself there's a data set there's an evaluator which takes your applications output along with potentially an output from your data set as a reference and produces a score like a grade for how well your application is performing on the particular like input example in your data set so that's really the flow and

one of the things we've seen is that llm as judge evaluators are really popular and the idea here is simply LMS are really effective at judging for natural language outputs for example many apps like a rag application will produce an answer in natural language you want some way to judge that potentially relative to reference potentially relative to some criteria and LMS can be really effective at that but the challenge is that you actually need to audit your llm

as judge and need to prompt it accordingly and this is you know potentially really a real challenge depending on your criteria and your application so what we've shown and we just released a Blog post today is basically a way to align your LM as judge of valuers with human preferences or kind of your own criteria using a correction flow and capturing human corrections to your judge as few shot examples that you feed to your evaluator so that's kind of the big idea now we

came out with a video today showing how to use that with an online evaluator so that is if you an application in production you can take an evaluator that's running on your application's outputs or inputs and you can improve it using this approach but now I'm going to show how you can do the same thing on a data set so let's just go over to a notebook here so I've created a rag bot I've indexed three block posts I've

already done this so basically here's my bot and it's really simple so basically here's a simple rag prompt I'm going to be retrieving from my Vector store passing those documents to my llm I'll use gbd4 so that's kind of the setup now we just went through in the pess video how to set that up using an onine value or and now I want to show you how to do this using a data set so often times in offline evaluations we

run against the data set so here's an example data set it's some set of question answer pairs from my react application related to my blog post so this is you know related to react how does react work and this is kind of the answer and it kind of goes through so I can just create this data set and now this will be created over in Langs Smith I can go to Langs Smith and I can see I

have this data set here here's my examples inputs outputs so that's great now here's a simple function that I'm just going to use to wrap my initial rag bot and really this is only necessary if you want to for example change the output keys and so what's going to happen is this function is going to take a dictionary M which is an example from my data set and of course run my bot on the example and then

return the response you know the response answer and the response context or documents as you know userdefined Keys that's all I'm doing there now we've talked about this quite a bit before here is a baseline prompt for answer relevance and I'm going to go ahead and Define that so we've gone through this before this is using LM as judge to basically grade my answer as well through to the reference answer and I'm using kind of a standard prompt to do

that and I can kick off evaluation here great so that evaluation ran I can go over to lsmith now I see this experiment is logged to my project I can see the aggregate score answer versus reference score here 60% I can open this up I can look at each one so I can actually open this up here is the reference input here's reference output here's my models output and here is the greater score in this case the answer

versus reference grader scored it as one so this is all pretty nice and I can look at that for my whole data set now if I go back here we can go back and look at the the bot itself very quickly and we can see something that's kind of nice when I return this invoke llm I get both answer and the context or the documents back out so what that allows me to do if I want I can set up a more granular

valuation on just the documents themselves which I may want to incorporate in My overall testing as you can see we only looked at the answer evaluation here and we passed that in through the SDK and that's what we're looking at over here but let's say I want to do a little bit more I want to kind of dig into some intermediate evaluations and I want those to run you know on all the examples that I passed

through my evaluation so what I can do is on my data set here I can just add an evaluator and you can see this add evaluator button I'll call this document recall I'll use GPD 40 and I can use this suggested evaluator prompt here and I can basically pick document relevance recall so what's nice is this is basically going to default me a prompt that's going to grade the documents retrieved by my system so that's actually pretty

nice so I'm going to do a few things here first I want to hook up my chain to my prompt so for facts I can just pick the output documents so we just went through and showed that our rag bot's going to return both the answer and the document so that's really convenient for us we can just grab the documents and I can also just get the input question so I'm going to grab the question from the

input I'm going to get the documents from the output of my chain those are going to get passed in then to my prompt and what I'm going to do is I'm going to add this F shot example thing and we're going to talk about this a little bit later but this is actually where I can inject human feedback to improve this particular grader so here is the placeholder for that so basically this is going to be the facts

the question the user feedback that I'm going to give the greater for anything I correct and the score itself so this is basically going to be the corrected score and what I can add to my prompt is here are a few examples from grading to calibrate your answers your grade let's just say that to calibrate your grade now you're going to see this I

think this pretty nice you can actually select the number of few shot examples you want to add so these examples are going to come from the user Corrections they'll be randomly selected depending on the number of overall Corrections that are there so of course if the number of Corrections is less than this number then it'll just add everything but let's say I went ahead and had dozens of Corrections it'll randomly sample five in this particular case and one other little thing I can

basically set this output key of my grer to be any value so it's actually pretty nice I'll call this recall and it's going to return a score of 01 based on my prompt right here and it's going to have a placeholder for any corrections I make to this grader so that's really all that's going on here and I can actually preview it so I can test and make sure that it's actually formatted correctly and yep documents and questions all get

added this is just taking example feed piece of feedback so there we are nice so that's one of our evaluators this will be for document recall and I can add another for document precision I'll use 40 try and evaluate a prompt Precision so basically this is going to be the same as we did before I can add this as a set of few shot

examples nice I'll call this precision and I'll say here are some examples to calibrate your grading and I'm just going to hook up hook this up to my chain so again I can get the output documents and the input question and that's it that's all I need to do so I'm going to save that so now I

have two graders tied to my data set for recall and precision so anytime I run an experiment on this data set

those graders are going to run so if we call I ran my first experiment and only looked at the answer evaluation now I have two additional graders that'll look at the document relevance and recall so I have this all configured and I can go back I can kick off another run on my data

set great so our second experiment finished and what's nice is we can now see that Precision recall are logged so that's pretty cool I can open the this up and now we can see three different scores are a run for each of our examples so again this is our input question we can even open this up here input question reference output and then llm output answer and we can see this was our original kind of answer versus reference

scoring now Precision recall are added as evaluators pinned our data set these will always run every time we run an experiment so that's pretty nice it kind of gives us a an easy ability to make sure that some diagnostic checks are run every time you run an experiment so I can kind of scroll through here quickly I can see something kind of interesting so here's a case where the answer is gred as incorrect but the Precision and recall of the

retrieved documents are both graded as correct so that's kind of an interesting and a weird case you would expect if the documents are effectively are perfect you know it's a bit odd then for our answer to be you know incorrect so I can open that up here and I can look a bit more at the particular Trace so let's have a look

there nice and I can scroll down and look at the documents themselves so here's the documents that we get out right so the question was related to the difference between reaction and reflection and if I go down and look at the documents here let's just have kind of a quick look so the first document definitely references react and let's just see if it really talks about the mechanism of action at all

okay so this is good so it talks about react and reflection framework equips agent dynamic memory self reflection so this first document looks pretty good and relevant self-reflection is aital part so the second document does talk about react the third document talks about reflection the fourth document mentions

react but it really only mentions it in kind of a superficial way it basically says react combines Co prompting with queries it doesn't talk about reflection at all so if you look at the other documents either talk about reflection in some detail or they talk about react you know in reasonable detail so what I might say is this fourth document for example I don't think really counts as a relevant document relative to the question of like the

mechanism of action for react versus reflection and particular there self-reflection approaches so what I would do here here is I can I think the recall is one so recall again is scoring whether or not any of the documents are relevant to your question Precision is scoring or any of the documents irrelevant so Precision is kind of grading you negatively if any of the documents are are spous and I think that's the case here so what I'm going

to say is here I'm goinging make correction and I'm basic going to say the final document only super superficially mentions the self mentions react and does not not talk about the self reflection mechanism so we should not

score it as a one for precision so that's it I can update this I can add this as feedback and so there we go so I've updated my grading I've added feedback and now we can go to our valuators here and we can go doc in Precision we just say edit and we can go and see this F shot data set we can open this up and we can

see that this now contains the correction we just made this is pretty nice so here's the updated score and here's explanation so what's cool here is if I go back this will now get sucked into my prompt as one of these few shot examples and it's going to be added right here and I can even see that so I can hit this preview button and I can see that this VI shot example should be added and let's just make

sure yep there it is the final document only Superion superficial Dimensions react so we're going to score it as Precision zero so there we go go out so now what I've done is I've corrected my evaluator for precision based on one of the grades and I've added that now as an example to the grader so for future iterations it should capture that feedback and

hopefully calibrate its outputs based on my userdefined feedback so I now can go back now let's say I run a future pass of this particular experiment so we can see our experiment ran I can open it up and I can go ahead and check now let's look at that third question was the difference between reactor and reflection now you can see here the evaluator is calibrated now the Precision is actually scored to zero which is what we would

expect and that's what we added to the feedback so again let's just open that up and see the expanded view so we can see you know here's the question here is the reference answer here is our model output and our document retrieval Precision now is greater than zero we can even open that fur we can look at the trace and again it's the same probably is the same three documents that were retrieved yep exactly so this this

fourth document is what we deemed to be irrelevant and it looks like our evaluator feedback has been captured and now our evaluator has been calibrated accordingly based on our feedback so again if we pull all the way back up you know what's really going on here what we've done is we've set up a rag agent or or in this case actually it's more like a simple rag chain we set up a rag chain we run evaluations with our

rag chain on a data set and we've pinned two evaluators to our data set one for document Precision one one for document recall and we had a look in the second experiment at the outputs of those evaluators we graded them so we said okay here's some cases where in in our case we didn't like the fact that it graded the precision as perfect in a case where some of the documents were not particularly relevant to the

question we modified the grade and we updated our evaluator such that that feedback is actually rolled in and you can see right here as few shot examples and so it basically calibrates our llm as judge evaluator based upon feedback that we're giving it and so it's a really nice tool we can use with online EV valuers like we just showed and here we actually show that we can use it with data sets as well so for offline

evaluation with data sets and lsmith so it's a very nice tool and it's particularly useful in these case of llm as judge

EV valuers which just through prompting alone can't always capture all the human preferences that you have the ability to add feedback with explanations significantly enhances your ability to kind of customize these LMS judge evaluators