No Code LangSmith Evaluations

7 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=DPIMOC1CSZG
https://www.youtube.com/watch?v=DpimoC1Cszg

SUMMARY

Lance from Lang Chain introduces a new feature in Langraph Studio that simplifies the evaluation process for Langraph agents, making it accessible to both developers and non-technical users. This enhancement allows users to run evaluations effortlessly, modify configurations, and view results without needing extensive technical knowledge.

- *New evaluation feature added to Langraph Studio for easier access.*
- *Non-developers can now run evaluations on Langraph agents without coding.*
- *Users can select datasets and initiate evaluations with a simple button click.*
- *Evaluations log results in Langsmith, providing a clear overview of experiments.*
- *Users can modify graph configurations, including models and prompts, through a no-code interface.*
- *The setup process involves defining datasets and pinning evaluators, which can be done by developers upfront.*
- *The feature supports rapid iteration, allowing for quick testing of different configurations.*
- *Results from evaluations can be easily compared to previous experiments for performance assessment.*

Hey, this is Lance from Lang Chain. Evaluations are one of the most important ways to build effective agents. And we wanted to lower the barrier to entry so that anyone, not just developers, can very easily run evaluations on agents that you're building. So, we've recently added the ability to run evaluations on Langraph agents directly from Langraph Studio. This is an agent, Opend Research, that we've developed over the past few months. It's a very popular repo and many people use it. Some of the people that use it are of course developers, but others are not yet still want the ability to test different configurations of their researcher. So, we've recently added this button you see on the upper right to Langraph Studio that allows you to quickly run an evaluation for any Langraph agent that's spun up in Studio.

I can click this and I can just select a data set that I want to run the evaluation on. This data set's already configured for this particular agent. It's open deep research workflow examples. And I can hit start. And you can see the status of the experiment is shown here in the upper right. And when the experiment is done, you'll see this experiment completed notification. And when the experiment completes, you can see I can just click this to view it. Here I am. These are some prior experiments I've run. This is the most recent.

And now we're simply in the Langmith data set view. We can see here are our evaluation examples. These are research topic inputs. We can look at each of our data set examples. Each contain an input topic for research as well as some reference source docs. And we can see our model outputs. Here's the final report. Very nice. And the scores from grading the output on our evaluator criteria, which we'll talk about in a bit. But this is showing you the overall flow going very simply from an agent studio to an experiment that's logged to Langsmith. Now, I do want to underscore briefly what is really the case for this and why is it interesting?

Well, the case for evaluation should be pretty clear to many at this point. Evaluations are very important for building effective agents. The challenge is that typically evaluations are fairly limited to developers. For example, you need to know how to use the various evaluation SDKs like the Langsmith SDK for example. You need to know about Piest. You need to know about the evaluate API. You need to know the guts of how to set up evaluations. And that really limits the audience of evaluation to developers. So our interest here was building a noode way that anyone could run an evaluation on a Langarf agent. Now this is useful because there's plenty of things that non-technical users can have enough intuition on and would want to evaluate the model choice prompts and so forth.

And so with this feature, anything in your graph configuration can be very easily toggled in the studio interface and use the basis to kick off an evaluation again using this run experiment. Now let me show you a little bit more about how we set this up. So all I did was I have a data set. Okay. Now the data set design and definition is indeed often a challenging and timeconsuming part of setting up evaluations. And that's something that a developer can do upfront. So in this case, for example, someone else on our team set up this evaluation for OpenDep Research. It has a bunch of input topics and has a bunch of reference outputs. The reference outputs are not reports. They're just sources related to the topic which can be used by some evaluators. Okay, so this is just the data set. You can see it's just an input topic and again reference sources.

Now the only thing I needed to do ahead of time to set this up is I pinned an evaluator to the data set. Now this is a very nice thing that you can do in Langsmith using the UI. So you basically can go to this evaluator button in this evaluator tab to add a new evaluator. There's many different choices here including different pre-built templates. In my case, I've already done that. It's evaluate overall quality. It's an LM as judge evaluator. We can open this up and look at it. And we can see I just defined this system prompt. Your expert evaluator tasked with assessing the quality of a research report. I give it a bunch of different evaluation criteria. Again, this is using an LMS as judge. I give it a scoring rubric.

And this is where we pass in the report topic from the data set and the final report from our graph. Now you'll see something kind of interesting in this drop down here. You can see the input from our data set, the output from our graph and the reference outputs from the data set are all available as drop downs. And so it's very easy to configure this evaluator and pin it to your data set. And I can even define specified fields for the output scoring.

And that's actually all I needed to do. I've defined a data set. I pinned an evaluator to it and the data set is configured to work with this particular graph that I'm testing. So that's the work I've done up front as a developer. Now what's nice is you can spin up that particular graph or agent. This is again is open deep research in studio. And if I go to manage assistance, anything here can be configured and modified to run different evaluation. As an example, let's say I wanted to modify the planner model as well as the writer model. Those can be very easily done. No code in the configuration as can prompts. As you see here, this is actually the planner prompt. It's all available in the configuration. And I could easily modify this and create a new assistant. So now I've created a new assistant.

I've modified the writer model and the planner model. And I can go ahead and kick off a new experiment with those modify configurations. Select the data set. Again, we were just looking at the data set. This is already configured to work with this particular agent. Start. And I've kicked off a new evaluation with a different configuration setting. So even if you're a developer, this is actually quite useful. I do this all the time. I want to modify my graph configurations and quickly run an evaluation to see how different models work, how different prompts work. And the ability to do this in studio in a no code way is quite convenient not only for nontechnical users but also for developers for rapid iteration.

So that experiment completed again. We just click the link to view it. We can see the updated scores and we can compare against prior evaluations. In this case, you can see that the assistant name, new assistant, is included in the experiment, which is quite nice because then I can see, for example, the assistant that I just evaluated. The score is about the same. So there's no change in performance by tuning up the model to use a reasoning model, which is a good thing to know.

But the point is, it's very easy to modify the configuration and kick off different experiments across different configurations of your graph or agent. So this new feature really unifies agent building and testing with rapid evaluation. Easily done, no code in the Langraph Studio UI, allowing you to modify any graph configurations that you want to and easily view different experiments that are named based upon the assistant name. So hopefully this makes evaluations easier and unlocks the ability to better test the agents you're building. Thanks.