

Agentic AI with Orchestrated Workflows for Engineers

Clair J. Sullivan
clair@clairsullivan.com



CLAIR SULLIVAN
& ASSOCIATES

bill

Schedule

9:00 - 9:30	Module 0: Introduction to Agentic AI
9:30 - 11:30	Module 1: Foundational Agentic Patterns with n8n
11:30 - 2:00	Module 2: Advanced Patterns & Custom API Integration
2:00 - 4:00	Module 3: LLM Evaluation with Braintrust
4:00 - 4:30	Awards!

Module 3: LLM Evaluation with Braintrust

Credit where Credit is Due



Q What do you want to learn?

[Browse courses](#)

[Lightning Lessons](#)

[Apply to teach](#)



[All courses](#) > [Product](#)

FEATURED IN



Lenny's List

AI Evals For Engineers & PMs

★★★★☆ 4.7 (647)

Hamel Husain ML Engineer with 20 years of experience

Shreya Shankar ML Systems & Applied AI Evals Researcher

[View Syllabus](#)



This course is popular.
9 people enrolled last week.



Types of Automated Evals

- Code-based
 - Typically assertions to check for things like rule-based failures
 - Deterministic
 - Use these whenever possible!
- LLM-as-a-judge
 - Assesses subjective criteria
 - Probabilistic since using an LLM
 - Can be expensive, slow
 - Requires significant amount of development to do right
- Guardrails
 - Code-based checks or small classifiers
 - Block failures before they reach a user

What is Braintrust?

- Powerful AI observability platform for automating evals (among other things)
- “Captures model behavior, surfaces patterns, and provides a unified interface to understand failures and optimize prompts”
- Framework agnostic
- Provides LLM-as-a-judge built in
- Has CI/CD integration
 - Fail builds if model quality drops

LLM-as-a-Judge

- Frequently necessary, especially as workflow gets more complicated and more question-answer pairs needed
- “The only way to trust an LLM judge is to measure it against human labels.” -Hamel Hussain and Shreya Shankar
- Split your human-labeled data into 3 sets
 - Train: where you draw your few-shot examples from
 - Dev: used to optimize your judge
 - Test: final check that the LLM never gets to see

Steps of Running an Eval in Braintrust

1. Create an agent
2. Create a dataset of question-answer pairs you want to test
3. Create a scorer that will evaluate how well the agent answers the questions relative to what you expected the result to be
4. Run an experiment
5. Look at where the agent failed to get the right answer and iterate on both the agent and scorer to improve

What Types of Question-Answer Pairs to Include

- Should span the space of what the agent does
- Should include as many **user-generated failures** as possible taken from actual traces
 - Base this on proper error analysis, using open coding to annotate traces with brief, unstructured notes on the problem
 - Group these open codes into categories and assign one or a few of these categories to each failure
- Should include edge cases and unexpected events
- Should include question-answer pairs that cover every tool the agent has access to
- For production systems, should have approximately 100 different question-answer pairs

Some Tips from the Trenches

- Look at your data first to understand specific failures
- ALWAYS prioritize user failures over engineering team assumptions
- Measure the judge against human labels
- Use very clear language in your grading prompt
- Don't use vague metrics like "helpfulness" or "tone"
- Give very specific scores
 - Binary pass/fail is ideal because it drives to a definitive decision
 - A perfect score means the question-answer pairs are too easy

Schedule

9:00 - 9:30	Module 0: Introduction to Agentic AI
9:30 - 11:30	Module 1: Foundational Agentic Patterns with n8n
11:30 - 2:00	Module 2: Advanced Patterns & Custom API Integration
2:00 - 4:00	Module 3: LLM Evaluation with Braintrust
4:00 - 4:30	Awards!

A brief survey:

Pollev.com/clairsullivan399