EvalOps

Evals are not just a dev tool. Continual Evals provide our clients with trust and security knowing that we have our eyes on the ball and our client's best interests at heart.

"How do you make a blueberry muffin? You insert them at start and not stuff them in at end." – Anon from a talk on web accessibility.

If we had a quality product, how would we and our clients know?

Rather than split hairs about whether it is a tool, agent, app, we treat the word function in the mathematical sense of input->function->output.

The function can also use other functions to get revenant context.

Unit tests are based on one function.

Integration tests are based on functions that also use functions as well as tests on more than one function in a sequence.

End2End tests are for USER_START to USER_END.

Parameters:

- Input (question)
- Output (answer)
- get_context – content to base answer on (optional)
- tool_calls – other than content retrieval
- next_action – used to determine next action in agentic flow
- ground_truth – domain expert's expected output

Recall/Precision for Retrieved Context

How useful was the context retrieved in relation to the output answer?

- output > context **addition/hallucination**
- output < context **omission**
- output != context **contradiction**
- output == context **accurate**

How relevant to the ground truth was the retrieved context?

- context == ground_truth **accurate and complete**
- context < ground_truth **poor recall**
- context > ground_truth **hallucination**
- context != ground_truth **inaccurate**

How useful was the output compared to the ground truth??

- output > ground_truth **addition/hallucination**
- output < ground_truth **omission**

- output != ground_truth **contradiction**
- output == ground_truth **accuracy**

TOOL CALLS

How many of the expected tools where called and were any tools called that were not expected?

- **actual_tools_called** compared to **expected_tools_called** PRECISION/RECALL
- We can apply UNIT TESTS to each tool_call.

NEXT_ACTION

Used where workflow path accuracy needs to be determined. *Did we go off course?*

- actual_next == expected_next **accurate**
- !actual_next **incomplete**
- acutal_next != expected_next **inaccurate**

EVALUATORS

- Numerical
- BERT/Semantic/Embeddings
- ML analysis for custom metrics
- LLM as judge - {factual-complete-accurate-contradictory-tone}. There is a large body of approaches where we can have an agent assess one particular aspect of the output. Of course, we need to judge the judge too.
- Human evaluations for domain expert or end user. In the scheme of things, to generate a golden dataset of say 50 input/outputs is half a days work and is generally a one-off. Human evaluation is also the best and possibly final judge of our system.