OpenAI

Evaluation

Technique

February 2024

Overview

Evaluation is the process of validating and testing the outputs that your LLM applications are producing. Having strong evaluations (“evals”) will mean a more stable, reliable application which is resilient to code and model changes.

Example use cases

- Quantify a solution’s reliability
- Monitor application performance in production
- Test for regressions

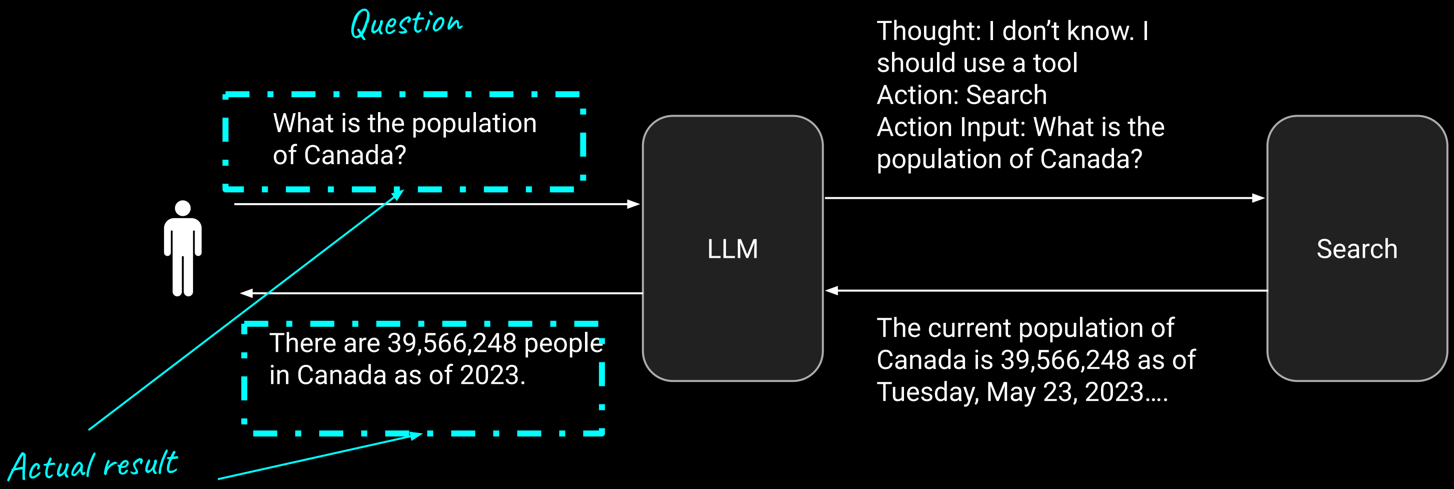
What we’ll cover

- What are evals
- Technical patterns
- Example framework
- Best practices
- Resources

What are evals

Example

An evaluation contains a question and a correct answer. We call this the **ground truth**.

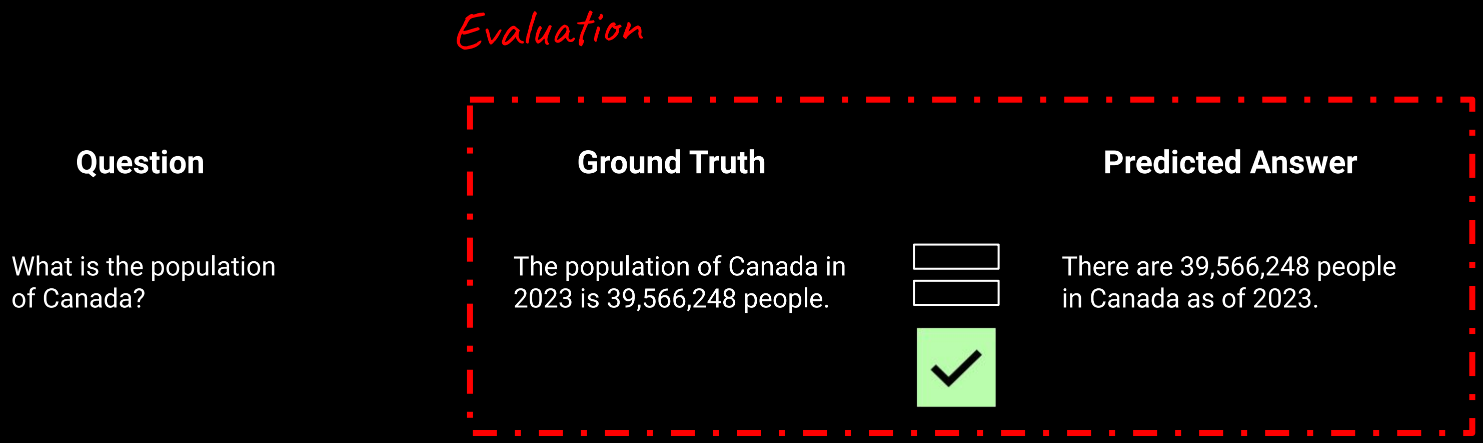


Question
What is the population of Canada?
LLM
Thought: I don't know. I should use a tool
Action: Search
Action Input: What is the population of Canada?
Search
The current population of Canada is 39,566,248 as of Tuesday, May 23, 2023....
There are 39,566,248 people in Canada as of 2023.
Actual result

What are evals

Example

Our ground truth matches the predicted answer, so the evaluation passes!



Evaluation

Question

What is the population of Canada?

Ground Truth

The population of Canada in 2023 is 39,566,248 people.

=

Predicted Answer

There are 39,566,248 people in Canada as of 2023.

✓

Technical patterns

Metric-based evaluations

- Comparison metrics like BLEU, ROUGE
- Gives a score to filter and rank results

Component evaluations

- Compares ground truth to prediction
- Gives Pass/Fail

Subjective evaluations

- Uses a scorecard to evaluate subjectively
- Scorecard may also have a Pass/Fail

Technical patterns

Metric-based evaluations

ROUGE is a common metric for evaluating machine summarizations of text

ROUGE

Metric for evaluating summarization tasks

Original

OpenAI's mission is to ensure that artificial general intelligence (AGI) benefits all of humanity. OpenAI will build safe and beneficial AGI directly, but will also consider its mission fulfilled if its work aids others to achieve this outcome. OpenAI follows several key principles for this purpose. First, broadly distributed benefits - any influence over AGI's deployment will be used for the benefit of all, and to avoid harmful uses or undue concentration of power…

→

Machine Summary

OpenAI aims to ensure AGI is for everyone's use, totally avoiding harmful stuff or big power concentration. Committed to researching AGI's safe side, promoting these studies in AI folks. OpenAI wants to be top in AI things and works with worldwide research, policy groups to figure AGI's stuff.

=

ROUGE Score

0.51162

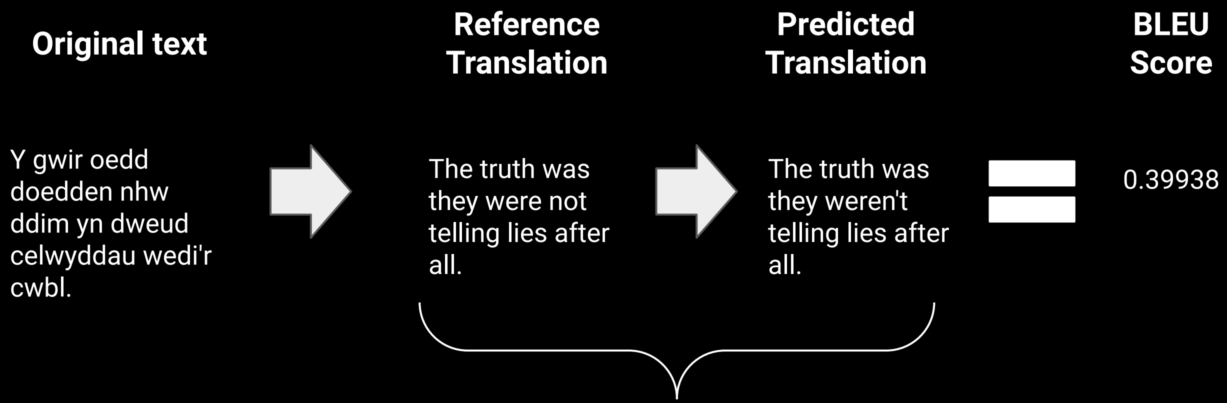
Technical patterns

Metric-based evaluations

BLEU score is another standard metric, this time focusing on machine translation tasks

BLEU

Metric for evaluating translation tasks



Original text

Y gwir oedd doedden nhw ddim yn dweud celwyddau wedi'r cwbl.

Reference Translation

The truth was they were not telling lies after all.

Predicted Translation

The truth was they weren't telling lies after all.

BLEU Score

0.39938

Technical patterns

Metric-based evaluations

What they’re good for

- A good starting point for evaluating a fresh solution
- Useful yardstick for automated testing of whether a change has triggered a major performance shift
- Cheap and fast

What to be aware of

- Not tuned to your specific context
- Most customers require more sophisticated evaluations to go to production

Technical patterns

Component evaluations

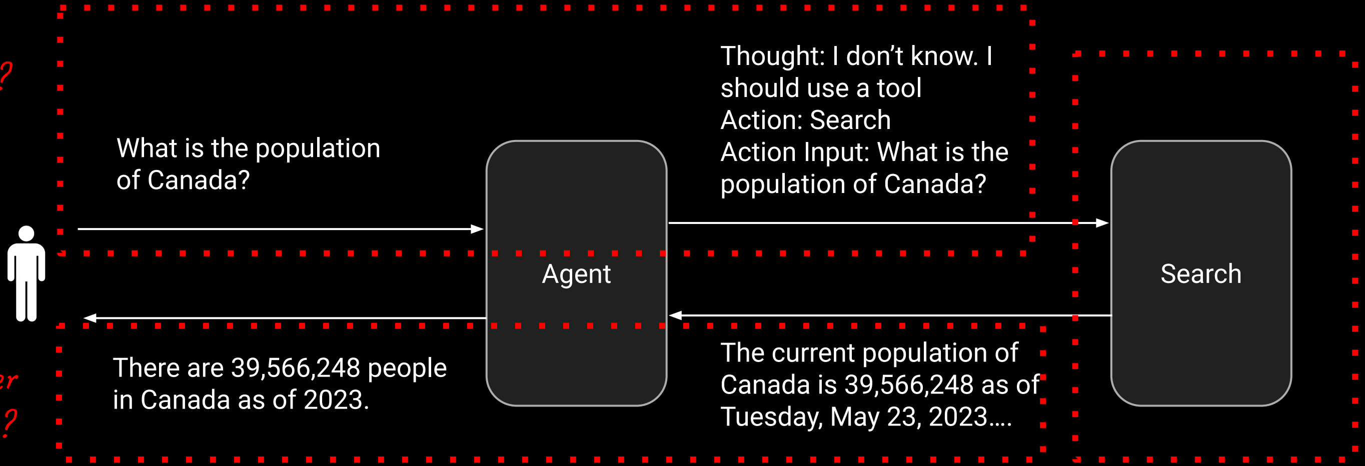
Component evaluations (or “unit tests”) cover a single input/output of the application. They check whether each component works in isolation, comparing the input to a **ground truth** ideal result

Is this the correct action?

Exact match comparison

Does this answer use the context?

Extract numbers from each and compare



What is the population of Canada?

Thought: I don’t know. I should use a tool
Action: Search
Action Input: What is the population of Canada?

There are 39,566,248 people in Canada as of 2023.

The current population of Canada is 39,566,248 as of Tuesday, May 23, 2023....

Is this the right search result?

Tag the right answer and do an exact match comparison with the retrieval.

Technical patterns

Subjective evaluations

Building up a good scorecard for automated testing benefits from a few rounds of detailed human review so we can learn what is valuable.

A policy of “show rather than tell” is also advised for GPT-4, so include examples of what a 1, 3 and 8 out of 10 look like so the model can appreciate the spread.

Example scorecard

You are a helpful evaluation assistant who grades how well the Assistant has answered the customer’s query.

You will assess each submission against these metrics, please think through these step by step:

- **relevance:** Grade how relevant the search content is to the question from 1 to 5 // 5 being highly relevant and 1 being not relevant at all.
- **credibility:** Grade how credible the sources provided are from 1 to 5 // 5 being an established newspaper, government agency or large company and 1 being unreferenced.
- **result:** Assess whether the question is correct given only the content returned from the search and the user’s question // acceptable values are “correct” or “incorrect”

You will output this as a JSON document: {relevance: integer, credibility: integer, result: string}

User: What is the population of Canada?
Assistant: Canada's population was estimated at 39,858,480 on April 1, 2023 by Statistics Canada.
Evaluation: {relevance: 5, credibility: 5, result: correct}

Example framework

Your evaluations can be grouped up into test suites called **runs** and executed in a batch to test the effectiveness of your system.

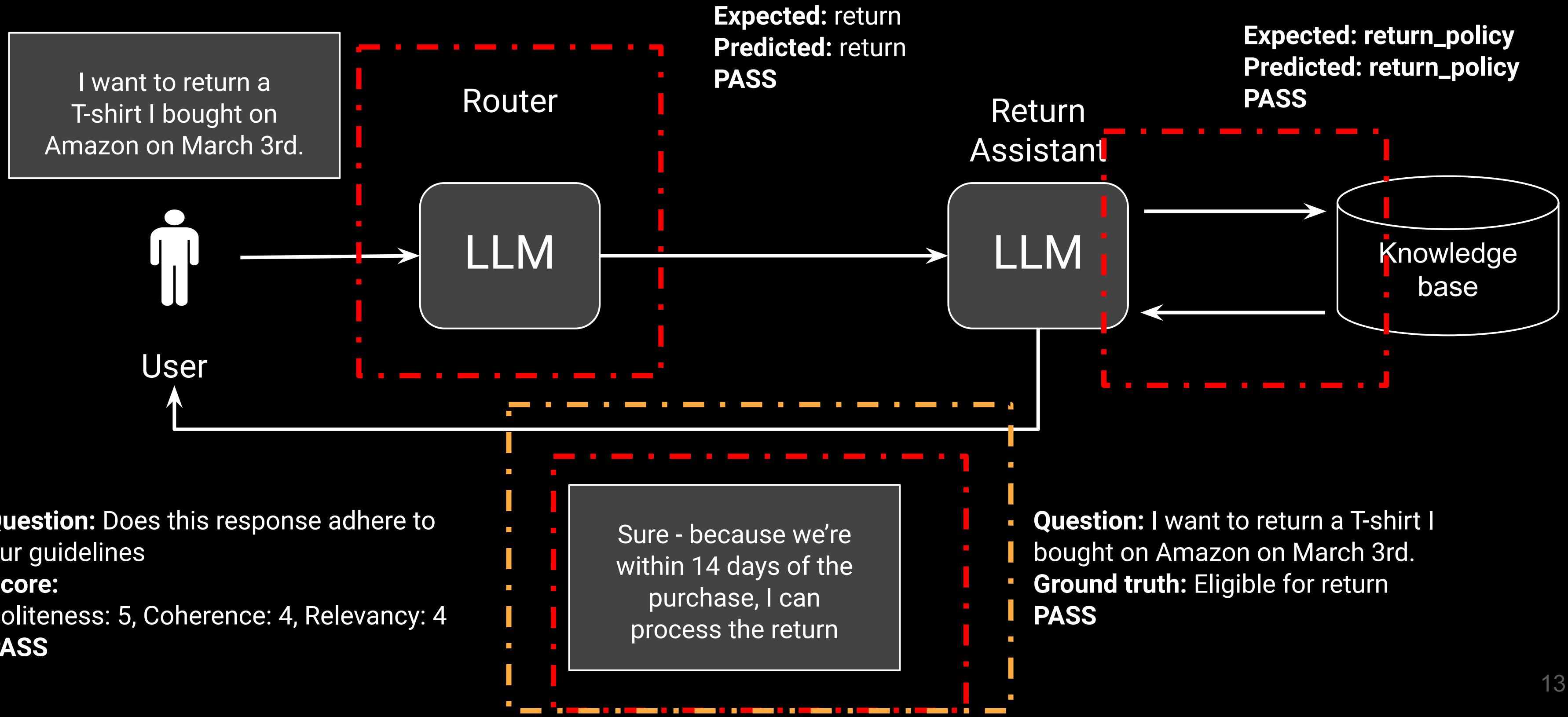
Each run should have its contents logged and stored at the most granular level possible (**“tracing”**) so you can investigate failure reasons, make tweaks and then rerun your evals.

Run ID	Model	Score	Annotation feedback	Changes since last run
1	gpt-3.5-turbo	28/50	● 18 incorrect with correct search results ● 4 incorrect searches	N/A
2	gpt-4	36/50	● 10 incorrect with correct search results ● 4 incorrect searches	Model updated to GPT-4
3	gpt-3.5-turbo	34/50	● 12 incorrect with correct search results ● 4 incorrect searches	Added few-shot examples
4	gpt-3.5-turbo	42/50	● 8 incorrect with correct search results	Added metadata to search Prompt engineering for Answer step
5	gpt-3.5-turbo	48/50	● 2 incorrect with correct search results	Prompt engineering to Answer step

Example framework

Component evals

Subjective evals



I want to return a T-shirt I bought on Amazon on March 3rd.

Expected: return
Predicted: return
PASS

Expected: return_policy
Predicted: return_policy
PASS

Sure - because we're within 14 days of the purchase, I can process the return

Question: Does this response adhere to our guidelines
Score:
Politeness: 5, Coherence: 4, Relevancy: 4
PASS

Question: I want to return a T-shirt I bought on Amazon on March 3rd.
Ground truth: Eligible for return
PASS

Best practices

Log everything

- Evals need test cases - log everything as you develop so you can mine your logs for good eval cases

Create a feedback loop

- Build evals into your application so you can quickly run them, iterate and rerun to see the impact
- Evals also provide a useful structure for few-shot or fine-tuning examples when optimizing

Employ expert labellers who know the process

- Use experts to help create your eval cases - these need to be as lifelike as possible

Evaluate early and often

- Evals are something you should build as soon as you have your first functioning prompt - you won’t be able to optimize without this baseline, so build it early
- Making evals early also forces you to engage with what a good response looks like