



Trajectory Evaluators

Trajectory Evaluators in LangChain provide a more holistic approach to evaluating an agent. These evaluators assess the full sequence of actions taken by an agent and their corresponding responses, which we refer to as the "trajectory". This allows you to better measure an agent's effectiveness and capabilities.

A Trajectory Evaluator implements the `AgentTrajectoryEvaluator` interface, which requires two main methods:

- `evaluate_agent_trajectory`: This method synchronously evaluates an agent's trajectory.
- `aevaluate_agent_trajectory`: This asynchronous counterpart allows evaluations to be run in parallel for efficiency.

Both methods accept three main parameters:

- `input`: The initial input given to the agent.
- `prediction`: The final predicted response from the agent.
- `agent_trajectory`: The intermediate steps taken by the agent, given as a list of tuples.

These methods return a dictionary. It is recommended that custom implementations return a `score` (a float indicating the effectiveness of the agent) and `reasoning` (a string explaining the reasoning behind the score).

You can capture an agent's trajectory by initializing the agent with the `return_intermediate_steps=True` parameter. This lets you collect all intermediate steps without relying on special callbacks.

For a deeper dive into the implementation and use of Trajectory Evaluators, refer to the sections below.



Custom Trajectory Evaluator

You can make your own custom trajectory evaluators by inheriting from the `AgentTrajectoryEvaluator` c...



Agent Trajectory

Agents can be difficult to holistically evaluate due to the breadth of actions and generation they can m...