DRAFT · TECHNICAL METHODOLOGY

Agentic evaluation scoring methodologies

Author: Frankie Li

ABSTRACT

Abstract

G2's agentic evaluation is designed to grade the autonomous, agentic components of SaaS software. This amounts to asking agents to perform a set of N tasks, {T_1, T_2, T_3,... T_N} and score the trace, outputs, and the environments according to a set rubric using a combination of deterministic evaluators and LLM-based judges. The task score (s_t) across all N tasks are combined into a single number, *agent score* (s_a) using the agent scoring formula. In this document, we'll define the task level LLM judge grading methodologies and agent scoring formula. This document will serve as a single source of truth from the technical perspective.

AGENT SCORING

Agent scoring

We will start with some formalism needed to precisely define the final score of the system. We will skip over the detailed definitions of tasks, as they are unimportant for this current exercise.

Definitions

Trace (Tr)
Trace is the transcripts between the Agent and the [simulated] users, including all observable tool calls in a user-Agent interaction. We will denote the trace of a specific Agent performing a task (A, t) to be Tr((A, t)).

Evaluator (Eval)
A function that evaluates the trace of an agentic interaction. The evaluator produces a score, s_t, where s_t is the score assigned by the evaluator to a specific trace.

Environment (Env)
The collection of data (context), state, and tools available for the Agent and the simulated user. The environment is stateful. Both the Agent and the Simulator User may modify the environment's states.

$$Eval(Tr((A, t)), Env, Output) \rightarrow s_t$$

In our current approach, the evaluator scores each tasks' trace in four dimensions: *Relevance*, *Factual Correctness*, *Policy Compliance*, and *Completeness*, each being a numerical value between 0.0 – 1.0.

Dimension	Scored by	Core question
relevance	LLM	Did the agent stay on the user's real need and the right target?
completeness	Deterministic	How much of the tau assertion set (env + NL) passed? Here, asserts define expected outcome.
factual_correctness	LLM	Are material explicit claims supported, or directly contradicted, by evidence?
policy_compliance	LLM	Did the agent follow the explicit rules in the provided policy text?

Table 1. Scoring dimension definitions

The 4-tuple (R, F, P, C) corresponding to the four dimensions is combined into a final task score, s_t, by the two score combination (judging) modes (see Table 2).

Mode	LLM dimension output	Completeness output	Compound score
Scalar judge (v1_scalar)	score from 1 to 5, normalized with (score - 1) / 4.0	Numeric coverage from 0.0 to 1.0	Average of all four normalized dimensions.
Binary judge (v1_binary)	decision of pass or fail, normalized to 1.0 or 0.0	Numeric coverage from 0.0 to 1.0	1.0 only when all four normalized dimensions are 1.0; otherwise 0.0.

Table 2. Score combination modes.

Few technical notes

Currently, evaluations are executed **only once**. That means each task gets tested once only. In a near-future world, we will be running each task repeatedly. The combined score defined above shall be replaced with a pass@k and a pass^k derived value.

Agent score (S_a)

Composite score incorporating an agent's performance across a collection of selected tasks.

Each agent is being graded on a list of N tasks and their resultant traces. Because tasks have different difficulties and complexities, each of our tasks are associated with a weighting. We therefore assign an Agent Score (S_a) for each of the agents we evaluate, defined below:

$$S_a = sum_i^N w_i * s_t^{(i)}$$

The weights for each of the tasks are defined by the table below:

Task complexity	Weight
Standard	5
Advanced	10
Complex	20

Tasks that are complex today will likely become standard in the future as agents become more powerful.

TASK GRADING METHODOLOGIES

Task grading methodologies

We have employed four structured preprocessing steps to evaluate each trace generated from an agent performing a specific task. This results in what we call an "*evidence packet*." The structure approach lets us manage the size and importance of the context before we perform the LLM Judge based evaluation. Each of the evidence packets is then independently scored by a dedicated LLM judge along with supplemental information. For example, to evaluate the policy compliance dimension of a trace, we would need to supply our LLM judge both the policy packet, the policy related evidence extracted from the trace, and also the task policy document as part of the task definition into a policy compliance LLM judge. The result of the LLM judge is a scalar value between [0.0 - 1.0].

REFERENCE

Reference

1. https://g2dotcom.slack.com/archives/C0B246Q75K9/p1786099740598649
2. https://github.com/g2crowd/g2-agent-eval/blob/main/docs/requirements/technical-overview/llm-judge/structured-llm-judge.md
3. High level diagrams: Agentic eval methodology

APPENDIX

FAQ

Why aren't we using ELO scores?

Vendors are differentiated, unlike models.