Running Log for Sam’s work trial test

Jun 8

- Setup Python env for project, bootstrap basic script for testing and confirm that provided keys are working.
- Go through METR website and their task sets
- Go through LangChain agent creation and usage pipelines
- Explore MMLU dataset on HuggingFace
- Downloaded a subset of the MMLU dataset
- Did some preprocessing on the dataset to present it to the custom agent
- Began writing the custom agent
- Figured out the pipeline through multiple hiccups and bugs (which were fixed)
- Constructed the system prompts for the seed model and the test model
- Decided to use meta/meta-llama-3-8b-instruct as seed model during dev and babbage-002 as test model during dev
- Tested an initial framework for function calling by the agent which did not work
- Came up with a rudimentary mapping based function calling protocol for the agent (side note: what about security concerns? current the index is matched, failing which an Exception is raised)
- Could have broken steps from evaluate_model and delegated them to be orchestrated by the seed_model. This would ensure us to make the agent much more flexible and could be “programmed” using natural language.
- Ran a simple request to `get_response()` and received a valid output.

Jun 9

- Started to scale up the agent to handle iterative queries by adding the `evaluate_model()` function.
- Decided to change the test model to a Llama 7B and found that the code for comparing for correctness is too fragile. Decided to let the seed LLM make the verdict.
- Added the `check_if_correct()` method, crafted the check system prompt, debugged and handled for edge cases. This is more versatile than a hardcoded equality test.
- Test for OpenAI model fail with a RateLimit error. Confused.
- Fixed a bug in the final accuracy reporting code
- Running multiple rounds of small batch tests
- Decide to run for 3 randomly picked models. Focusing on model param difference.
- Successfully ran tests on all 3 models.
- Decide to leave OpenAI models. Think it's a key fault.