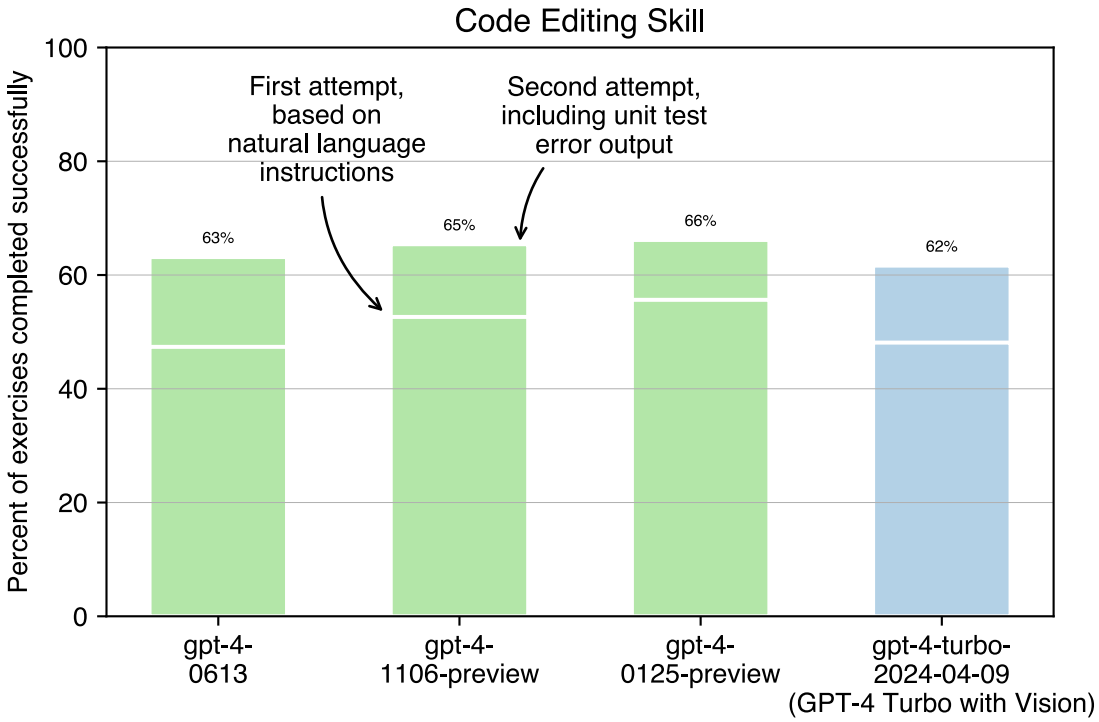
aider

April 09, 2024

GPT-4 Turbo with Vision is a step backwards for coding

OpenAI just released GPT-4 Turbo with Vision and it performs worse on aider's coding benchmark suites than all the previous GPT-4 models. In particular, it seems much more prone to "lazy coding" than the existing GPT-4 Turbo "preview" models.

Code editing skill



Aider relies on a code editing benchmark to quantitatively evaluate how well an LLM can make changes to existing code. The benchmark uses aider to try and complete 133 Exercism Python coding exercises.

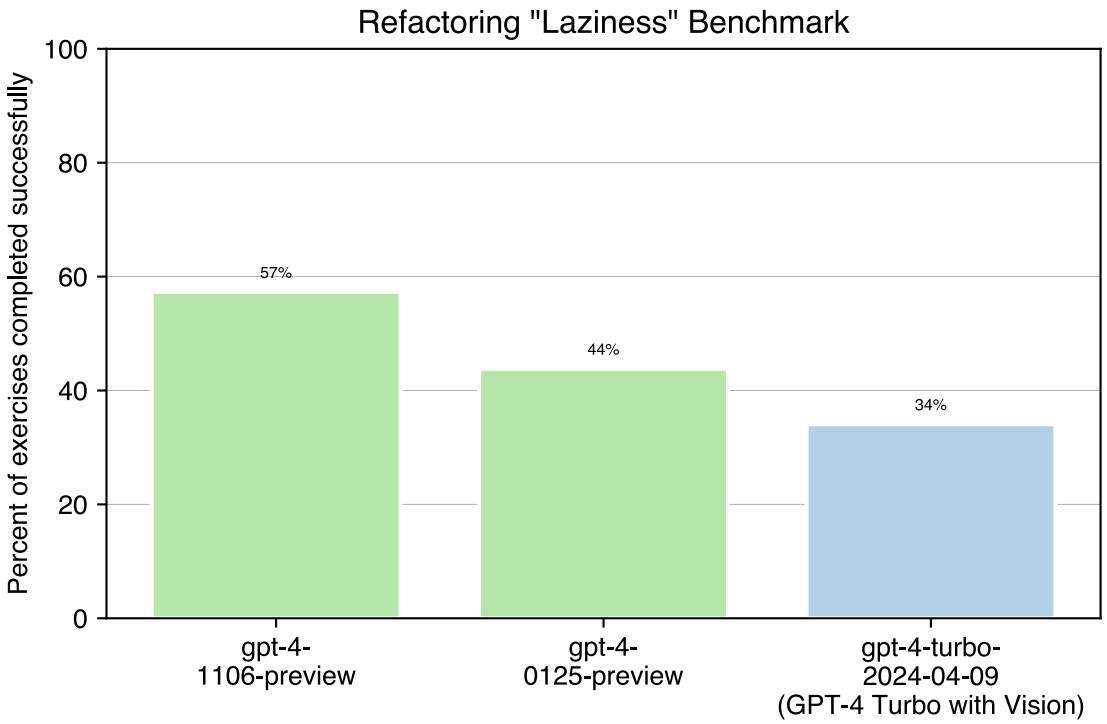
For each exercise, the LLM gets two tries to solve each problem:

1. On the first try, it gets initial stub code and the English description of the coding task. If the tests all pass, we are done.

2 If any tests failed, aider sends the LLM the failing test output and gives it a second try to complete the task.

GPT-4 Turbo with Vision scores only 62% on this benchmark, the lowest score of any of the existing GPT-4 models. The other models scored 63-66%, so this represents only a small regression, and is likely statistically insignificant when compared against `gpt-4-0613`.

Lazy coding



The GPT-4 Turbo “preview” models have been widely criticized for being “lazy” when coding. They often omit needed code and instead leave comments with homework assignments like “implement method here”.

```
def some_complex_method(foo, bar):
    # ... implement method here ...
```

Aider uses a “laziness” benchmark suite which is designed to both provoke and quantify lazy coding. It consists of 89 python refactoring tasks which tend to make GPT-4 Turbo code in that lazy manner.

The new GPT-4 Turbo with Vision model scores only 34% on aider’s refactoring benchmark, making it the laziest coder of all the GPT-4 Turbo models by a significant margin.

Conclusions

Aider has full support for the new GPT-4 Turbo with Vision model, which you can access using the switch `--model gpt-4-turbo-2024-04-09`. But aider will continue to use `gpt-4-1106-preview` by default, as it is by far the strongest coder of the GPT-4 models.