

AI 代码编辑器的未来发展方向

最近，我在 GitHub Actions 中添加了一个 `xelatex` 管道。

在 GitHub 的工作流程中，我遇到了 `fontawesome5` 包的问题。4o-mini 提供的解决方案（安装 TeX Live 2021 并使用 `tlmgr install fontawesome5`）对我不起作用。然而，4o 提出了一个更好的方法：升级到 TeX Live 2023，并继续使用 `tlmgr` 安装 `fontawesome5`。虽然这并没有完全解决问题，但切换到 TeX Live 2023 确实大大改善了情况。

我使用了 ChatGPT 来帮助解决这个问题。更多细节请查看 [What ChatGPT O1 Can Do That 4o-mini Cannot](#)。

此时，我并没有使用 Cursor 或 Windsurf 等编辑器，尽管在另一个项目中我曾尝试过它们。这些代码编辑器的问题在于，它们仅能捕获本地的测试输出，这限制了它们在云环境中的功能。

在像 GitHub Actions、Jenkins 作业或任何代码部署或测试流程中，代码编辑器需要更好的集成。它们应当能够与云和 CI/CD 流程无缝对接。

这种集成同样适用于其他内容创作工具——无论是文本、图像、音频还是视频。这些工具应该与 A/B 测试系统集成。AI 工具可以生成内容，而 A/B 测试工具可以提供反馈。这种动态类似于人类反馈强化学习（RLHF），即 AI 模型根据真实世界的反馈逐渐改进。

将 RLHF 的概念从仅仅应用于模型输出，扩展到真实世界的测试和部署环境，这似乎是代码编辑器和 AI 驱动内容创作工具未来发展中的一个有前景的方向。

测试可以是即时的，也可以是长期的，既可以自动化，也可以由人类协助。如果测试是自动化的，比如用户对 AI 工具进行 A/B 测试，它仍然涉及人类反馈，但整个过程是自动化的。例如，我们可以让计算机根据 A/B 测试的结果每天或每小时进行检查，以改进创作过程。同样，在 Jenkins 或 GitHub Actions 作业中，我们可以让计算机在任务完成后进行检查。

如果涉及人类协助，机器无法完全理解这些反馈，而且反馈往往有些模糊。例如，当 AI 工具创建图像或视频内容时，人类可能会指出内容不够有趣，或者某个细节需要改进。机器在让一切完美方面仍然有很长的路要走，是否“完美”通常是主观的，取决于个人口味。正是人类的反馈帮助使事物变得更好。

理论上，所有人类定义的规则都可以写成提示语。提示语分为用户提示和系统提示。我们应该专注于改进提示语，而不是每次都去修正输出。