

# AI コードエディタの次の方向性

最近、GitHub Actions に `xelatex` パイプラインを追加する作業をしていました。

GitHub フローで `fontawesome5` パッケージに関する問題に遭遇しました。4o-mini が提案した解決策 (`TeX Live 2021` をインストールし、`tlmgr install fontawesome5` を使用する) は私の場合うまくいきませんでした。しかし、4o はより良いアプローチを提案しました：`TeX Live 2023` にアップグレードし、引き続き `tlmgr` を使用して `fontawesome5` をインストールするというものです。これで完全に問題が解決したわけではありませんが、`TeX Live 2023` に切り替えることで状況が大幅に改善されました。

私は問題を解決するために ChatGPT を利用しました。詳細については、What ChatGPT O1 Can Do That 4o-mini Cannotをご覧ください。

この時点では、`Cursor` や `Windsurf` のようなエディタは使用しませんでしたが、別のプロジェクトで試してみました。これらのコードエディタの問題点は、ローカルのテスト出力しかキャプチャできないことであり、クラウド環境での機能が制限されることです。

GitHub Actions や Jenkins ジョブ、あるいはコードのデプロイやテストフローといったワークフローにおいて、コードエディターはより良く統合される必要があります。それらはクラウドや CI/CD プロセスとのシームレスな連携を提供すべきです。

この統合は、テキスト、画像、音声、ビデオなど、他のコンテンツ作成ツールにも適用されます。これらのツールは A/B テストシステムと統合されるべきです。AI ツールがコンテンツを生成し、A/B テストツールがフィードバックを提供するという仕組みです。このダイナミクスは、人間のフィードバックによる強化学習 (RLHF) に似ており、AI モデルが現実世界のフィードバックに基づいて時間とともに改善されていきます。

RLHF (人間によるフィードバックを用いた強化学習) の概念を、単なるモデルの出力を超えて、現実世界のテストやデプロイ環境にまで拡張するという考え方には、コードエディタや AI 駆動のコンテンツ作成ツールの両方において、改善のための有望な方向性のように思えます。

テストは即時に行われることもあれば、長期間にわたることもあります。また、自動化されている場合もあれば、人間の支援を受けて行われる場合もあります。例えば、AI ツールのユーザー A/B テストが自動化されている場合でも、依然として人間のフィードバックが関与していますが、そのプロセスは自動化されています。具体的には、A/B テストの結果に基づいて、コンピュータに毎日または毎時間結果をチェックさせ、作成プロセスを改善することができます。同様に、Jenkins や GitHub Actions のジョブについても、タスクが完了した後にコンピュータにチェックさせることができます。

人間の支援が関与する場合、そのフィードバックは機械には完全に理解できず、しばしばやや曖昧です。例えば、AI ツールが画像や動画などのコンテンツを作成する際、人間はその内容が十

分に面白くないとか、特定の詳細を改善すべきだと指摘するかもしれません。機械がすべてを完璧にするにはまだ長い道のりがあり、何が「完璧」かはしばしば主観的で、個人の好みに依存します。物事をより良くするのは、人間のフィードバックなのです。

理論的には、人間が定義したルールはすべてプロンプトとして記述することができます。ユーザープロンプトとシステムプロンプトがあります。毎回出力を修正するのではなく、プロンプトを改善することに焦点を当てるべきです。