

Die nächste Richtung von KI-Code-Editoren

Kürzlich habe ich daran gearbeitet, eine `xelatex`-Pipeline zu GitHub Actions hinzuzufügen.

Ich bin auf ein Problem mit dem `fontawesome5`-Paket im GitHub-Workflow gestoßen. Die von 4o-mini vorgeschlagene Lösung (Installation von TeX Live 2021 und Verwendung von `tlmgr install fontawesome5`) hat bei mir nicht funktioniert. Allerdings hat 4o einen besseren Ansatz vorgeschlagen: ein Upgrade auf TeX Live 2023 und weiterhin die Verwendung von `tlmgr`, um `fontawesome5` zu installieren. Obwohl dies das Problem nicht vollständig behoben hat, hat der Wechsel zu TeX Live 2023 die Situation erheblich verbessert.

Ich habe ChatGPT verwendet, um das Problem zu lösen. Weitere Details finden Sie unter Was ChatGPT O1 kann, was 4o-mini nicht kann.

Zu diesem Zeitpunkt habe ich Editoren wie Cursor oder Windsurf noch nicht verwendet, obwohl ich sie in einem anderen Projekt ausprobiert habe. Das Problem mit diesen Code-Editoren ist, dass sie nur die lokale Testausgabe erfassen, was ihre Funktionalität in Cloud-Umgebungen einschränkt.

In Workflows wie GitHub Actions, Jenkins-Jobs oder jedem Code-Deployment- oder Testprozess müssen Code-Editoren besser integriert werden. Sie sollten eine nahtlose Interaktion mit der Cloud und CI/CD-Prozessen ermöglichen.

Diese Integration gilt auch für andere Tools zur Inhaltserstellung – sei es für Text, Bilder, Audio oder Video. Diese Tools sollten mit A/B-Test-Systemen integriert werden. KI-Tools könnten Inhalte generieren, und A/B-Test-Tools könnten Feedback liefern. Diese Dynamik ähnelt dem Reinforcement Learning from Human Feedback (RLHF), bei dem KI-Modelle basierend auf Feedback aus der realen Welt im Laufe der Zeit verbessert werden.

Das Konzept, RLHF (Reinforcement Learning from Human Feedback) über reine Modellausgaben hinaus zu erweitern – in reale Test- und Einsatzumgebungen – scheint eine vielversprechende Richtung für Verbesserungen sowohl bei Code-Editoren als auch bei KI-gestützten Tools zur Inhaltserstellung zu sein.

Der Test kann entweder sofort oder langfristig durchgeführt werden und kann entweder automatisiert oder durch Menschen unterstützt werden. Wenn die Tests automatisiert sind, wie beispielsweise A/B-Tests für ein KI-Tool, beinhalten sie dennoch menschliches Feedback, aber der Prozess ist automatisiert. Zum Beispiel können wir den Computer täglich oder stündlich basierend auf den Ergebnissen von A/B-Tests überprüfen lassen, um den Erstellungsprozess zu verbessern. Ähnlich können wir für Jenkins- oder GitHub-Actions-Jobs den Computer nach Abschluss ihrer Aufgaben überprüfen lassen.

Wenn menschliche Unterstützung beteiligt ist, kann das Feedback von der Maschine nicht vollständig verstanden werden und ist oft etwas vage. Zum Beispiel, wenn KI-Tools Inhalte wie Bilder oder Videos erstellen, könnten Menschen anmerken, dass der Inhalt nicht lustig genug ist oder dass ein bestimmtes Detail verbessert werden sollte. Maschinen haben noch einen langen Weg vor sich, um alles perfekt zu machen, und ob etwas "perfekt" ist, ist oft subjektiv und hängt vom individuellen Geschmack ab. Es ist das menschliche Feedback, das dazu beiträgt, Dinge besser zu machen.

Theoretisch können alle vom Menschen definierten Regeln als Prompts geschrieben werden. Es gibt Benutzer-Prompts und System-Prompts. Wir sollten uns darauf konzentrieren, die Prompts zu verbessern, anstatt jedes Mal die Ausgabe zu korrigieren.