

Die Fälle, in denen wir die GitHub-Suchleiste noch benötigen

jobs:

awesome-cv-copy:

runs-on: ubuntu-latest

steps:

...

- name: TeX Live 2023 installieren

if: steps.cache-texlive.outputs.cache-hit != 'true'

run: |

Abhängigkeiten für den TeX Live-Installer installieren

sudo apt-get update

sudo apt-get install -y perl wget xz-utils

TeX Live Installer herunterladen

wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz

tar -xzf install-tl-unx.tar.gz

cd install-tl-*/

...

- name: Fehlende LaTeX-Pakete installieren

run: |

sudo /usr/local/texlive/2023/bin/x86_64-linux/tlmgr install etoolbox adjustbox

- name: Paketinstallation bestätigen

run: |

kpsewhich etoolbox.sty

kpsewhich adjustbox.sty

- name: Führe make awesome-cv-copy aus

run: make awesome-cv-copy

Ich arbeite an dem oben genannten GitHub Actions-Skript.

Ich muss GitHub durchsuchen, um den genauen Code für ``etoolbox adjustbox language:YAML`` zu finden.

Ich bin auf den folgenden Fehler gestoßen:

2025-01-07T22:34:58.6493408Z 2025-01-07T22:34:58.6493741Z ! LaTeX-Fehler: Datei `adjustbox.sty` nicht gefunden. 2025-01-07T22:34:58.6494172Z 2025-01-07T22:34:58.6494593Z Geben Sie X ein, um zu beenden, oder , um fortzufahren, 2025-01-07T22:34:58.6495322Z oder geben Sie einen neuen Namen ein. (Standard-Erweiterung: `sty`) ""

Ich suche speziell nach `etoolbox adjustbox language:YAML`, und die Ergebnisse auf GitHub sind begrenzt, mit nur 53 YAML-Dateien, die sowohl `etoolbox` als auch `adjustbox` enthalten. Ich brauche eine **exakte Übereinstimmung**.

Obwohl wir uns im Zeitalter der großen Sprachmodelle befinden, ist die Notwendigkeit, nach exakten Übereinstimmungen zu suchen, nach wie vor von entscheidender Bedeutung. Dies gilt insbesondere, wenn man die genaue Bedeutung von etwas überprüfen oder präzisen, funktionierenden Code finden möchte. Ebenso verlassen sich Plattformen wie Google, Twitter oder andere auf exakte Suchen, um die Bedeutung zu ermitteln. Wir wollen keine KI-generierten Ergebnisse oder solche mit kleinen Fehlern.

Für das Training großer Sprachmodelle könnten wir ein System entwickeln, das exakte Übereinstimmungen findet. Vielleicht können wir den **KMP (Knuth-Morris-Pratt)**-Suchalgorithmus mit der **Transformer-Architektur** kombinieren, um die Suchfähigkeiten zu verbessern. Die Verwendung von KMP mit Transformern könnte dabei helfen, genauere Ergebnisse für spezifische Codesuchen zu erzielen.

Derzeit können große Sprachmodelle nicht nach Dateisprachen wie YAML oder Python filtern. Ein erheblicher Teil der Informationen in der realen Welt ist jedoch auf diese Weise organisiert. Das bedeutet, dass wir große Sprachmodelle mithilfe von Dateien trainieren könnten. Wenn wir alle Textdaten nach Dateitypen organisieren, können wir das Modell darauf trainieren, sie besser zu verstehen. Für große Sprachmodelle müssten wir daher die Dateisprachen zu Beginn vordefinieren. Standardmäßig könnte es "Text" sein, aber wir könnten auch andere Sprachen definieren, genau wie es GitHub Search tut. Das Ergebnis würde Dateien zurückgeben, ähnlich wie die Suchergebnisse von GitHub.

Der wichtige Teil ist das **Dateiformat** oder die **Dateiendung**, nicht der Dateiname. Hier sind einige Beispiele:

Python, JavaScript, Java, Ruby, Go, C++, C, C#, TypeScript, HTML, CSS, PHP, Swift, Kotlin, Rust, Objective-C, Bash, Markdown, R, Lua, Haskell, MATLAB, Perl, SQL, Dockerfile, YAML, JSON, TOML, VHDL, TeX, LaTeX, Assembly, GraphQL

.py, .js, .java, .rb, .go, .cpp, .cc, .cxx, .h, .c, .cs, .ts, .html, .htm, .css, .php, .swift, .kt, .kts, .rs, .m, .h, .sh, .md, .r, .lua, .hs, .m, .pl, .pm, .sql, Dockerfile, .yaml, .yml, .json, .toml, .vhdl, .vhd, .tex, .asm, .graphql, .gql

Wenn jedoch die Eingabe eines Benutzers normalen Text mit dateiähnlichen Ausdrücken und Symbolen vermischt, wird eine solche Suche schwierig. Zum Beispiel enthalten auf Plattformen wie Stack Overflow Fragen oder Antworten oft Text, der mit Codeausschnitten oder Dateiausdrücken vermischt ist.

Aber sicherlich gibt es neue Produkte, die wir uns in diesem Bereich vorstellen können, um die Lücke zwischen der Suche in natürlicher Sprache und der dateibasierten Suche zu überbrücken.