

Les cas où nous avons encore besoin de la barre de recherche GitHub

jobs:

awesome-cv-copy:

runs-on: ubuntu-latest

steps:

...

- name: Installer TeX Live 2023

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

run: |

Installer les dépendances pour l'installateur de TeX Live

sudo apt-get update

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

Télécharger l'installateur de TeX Live

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

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

cd install-tl-*/

...

- name: Installer les paquets LaTeX manquants

run: |

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

- name: Confirmer l'installation des paquets

run: |

kpsewhich etoolbox.sty

kpsewhich adjustbox.sty

- name: Exécuter make awesome-cv-copy

run: make awesome-cv-copy

Je travaille sur le script GitHub Actions ci-dessus.

Je dois rechercher sur GitHub pour trouver le code exact pour ``etoolbox adjustbox language:YAML``.

J'ai rencontré l'erreur suivante :

```
2025-01-07T22:34:58.6493408Z 2025-01-07T22:34:58.6493741Z ! Erreur LaTeX : Fichier
adjustbox.sty introuvable. 2025-01-07T22:34:58.6494172Z 2025-01-07T22:34:58.6494593Z
Tapez X pour quitter ou pour continuer, 2025-01-07T22:34:58.6495322Z ou entrez un nouveau
nom. (Extension par défaut : sty) ""
```

Je recherche spécifiquement `etoolbox adjustbox language:YAML`, et les résultats sur GitHub sont limités, avec seulement 53 fichiers YAML contenant à la fois `etoolbox` et `adjustbox`. J'ai besoin d'une **correspondance exacte**.

Même si nous sommes à l'ère des grands modèles de langage, la nécessité de rechercher des correspondances exactes reste cruciale. Cela est particulièrement vrai lorsqu'il s'agit de vérifier la signification exacte de quelque chose ou de trouver un code fonctionnel précis. De même, des plateformes comme Google, Twitter ou d'autres dépendent de recherches exactes pour comprendre le sens. Nous ne voulons pas de résultats générés par l'IA ou ceux contenant des erreurs mineures.

Pour entraîner de grands modèles de langage, nous pourrions développer un système qui trouve des correspondances exactes. Peut-être pouvons-nous combiner l'algorithme de recherche **KMP (Knuth-Morris-Pratt)** avec l'**architecture de transformateur** pour améliorer les capacités de recherche. L'utilisation de KMP avec les Transformers pourrait aider à obtenir des résultats plus précis pour des recherches de code spécifiques.

Actuellement, les grands modèles de langage ne peuvent pas filtrer par langage de fichier comme YAML ou Python. Cependant, une partie importante des informations dans le monde réel est organisée de cette manière. Cela signifie que nous pourrions entraîner des grands modèles de langage en utilisant des fichiers. Si nous organisons toutes les données textuelles par types de fichiers, nous pouvons entraîner le modèle à mieux les comprendre. Ainsi, pour les grands modèles de langage, nous devrions prédéfinir les langages de fichiers au départ. Par défaut, ce pourrait être "texte", mais nous pourrions également définir d'autres langages, tout comme le fait GitHub Search. Le résultat renverrait des fichiers, tout comme les résultats de recherche de GitHub.

La partie importante est le **format de fichier** ou l'**extension**, et non le nom du fichier. Voici quelques exemples :

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

Cependant, lorsque l'invite d'un utilisateur mélange du texte normal avec des expressions et des symboles ressemblant à des fichiers, il devient difficile d'effectuer une recherche de ce type. Par exemple, sur des plateformes comme Stack Overflow, les questions ou réponses contiennent souvent du texte mélangé à des extraits de code ou des expressions de fichiers.

Mais certainement, il existe de nouveaux produits que nous pouvons imaginer dans ce domaine pour combler le fossé entre la recherche en langage naturel et la recherche basée sur les fichiers.