

Los casos en los que todavía necesitamos el cuadro de búsqueda de GitHub

jobs:

awesome-cv-copy:

runs-on: ubuntu-latest

steps:

...

- name: Instalar TeX Live 2023

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

run: |

Instalar dependencias para el instalador de TeX Live

sudo apt-get update

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

Descargar el instalador 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: Instalar Paquetes Faltantes de LaTeX

run: |

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

- name: Confirmar Instalación del Paquete

run: |

kpsewhich etoolbox.sty

kpsewhich adjustbox.sty

- name: Ejecutar make awesome-cv-copy

run: make awesome-cv-copy

Estoy trabajando en el script de GitHub Actions anterior.

Necesito buscar en GitHub para encontrar el código exacto para ``etoolbox adjustbox language:YAML``.

Me encontré con el siguiente error:

```
2025-01-07T22:34:58.6493408Z 2025-01-07T22:34:58.6493741Z ! Error de LaTeX: Archivo ad-justbox.sty no encontrado. 2025-01-07T22:34:58.6494172Z 2025-01-07T22:34:58.6494593Z Escribe X para salir o para continuar, 2025-01-07T22:34:58.6495322Z o ingresa un nuevo nombre. (Extensión predeterminada: sty) ""
```

Estoy buscando específicamente `etoolbox adjustbox language:YAML`, y los resultados en GitHub son limitados, con solo 53 archivos YAML que contienen tanto `etoolbox` como `adjustbox`. Necesito una **coincidencia exacta**.

Aunque estamos en la era de los modelos de lenguaje grandes, la necesidad de buscar coincidencias exactas sigue siendo crucial. Esto es especialmente cierto al verificar el significado exacto de algo o encontrar código funcional preciso. De manera similar, plataformas como Google, Twitter u otras dependen de búsquedas exactas para obtener resultados precisos. No queremos resultados generados por IA o aquellos con pequeños errores.

Para entrenar modelos de lenguaje grandes, podríamos desarrollar un sistema que encuentre coincidencias exactas. Tal vez podríamos combinar el algoritmo de búsqueda **KMP (Knuth-Morris-Pratt)** con la **arquitectura de transformadores** para mejorar las capacidades de búsqueda. Usar KMP con Transformers podría ayudar a encontrar resultados más precisos en búsquedas de código específicas.

Actualmente, los modelos de lenguaje grandes no pueden filtrar por el lenguaje de archivo como YAML o Python. Sin embargo, una parte significativa de la información en el mundo real está organizada de esta manera. Esto significa que podríamos entrenar modelos de lenguaje grandes utilizando archivos. Si organizamos todos los datos de texto por tipos de archivo, podemos entrenar al modelo para que los comprenda mejor. Por lo tanto, para los modelos de lenguaje grandes, necesitaríamos predefinir los lenguajes de archivo al inicio. Por defecto, podría ser "texto", pero también podríamos definir otros lenguajes, tal como lo hace GitHub Search. El resultado devolvería archivos, al igual que lo hacen los resultados de búsqueda de GitHub.

La parte importante es el **formato del archivo** o la **extensión**, no el nombre del archivo. Aquí tienes algunos ejemplos:

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

Nota: Los nombres de los archivos y extensiones no se traducen, ya que son términos técnicos universalmente reconocidos.

Sin embargo, cuando el mensaje de un usuario mezcla texto normal con expresiones y símbolos similares a archivos, se vuelve difícil realizar una búsqueda de este tipo. Por ejemplo, en plataformas como Stack Overflow, las preguntas o respuestas a menudo contienen texto mezclado con fragmentos de código o expresiones de archivos.

Pero ciertamente, hay nuevos productos que podemos imaginar en este espacio para cerrar la brecha entre la búsqueda en lenguaje natural y la búsqueda basada en archivos.