我們仍需使用 GitHub 搜尋框的情況

```yaml
jobs:
  awesome-cv-copy:
    runs-on: ubuntu-latest
    steps:

     # ...

      - name: 安裝 TeX Live 2023
        if: steps.cache-texlive.outputs.cache-hit != 'true'
        run: |
          # 安裝 TeX Live 安裝程序的依賴項
          sudo apt-get update
          sudo apt-get install -y perl wget xz-utils

          # 下載 TeX Live 安裝程序
          wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
          tar -xzf install-tl-unx.tar.gz
          cd install-tl-*/

      # ...

      - name: 安裝缺失的 LaTeX 包
        run: |
          sudo /usr/local/texlive/2023/bin/x86_64-linux/tlmgr install etoolbox adjustbox

      - name: 確認包安裝
        run: |
          kpsewhich etoolbox.sty
          kpsewhich adjustbox.sty

      - name: 運行 make awesome-cv-copy
        run: make awesome-cv-copy
```

我正在處理上面的 GitHub Actions 腳本。

我需要搜索 GitHub 來找到 `etoolbox adjustbox language:YAML` 的確切代碼。

我遇到了以下錯誤：

```
2025-01-07T22:34:58.6493408Z
2025-01-07T22:34:58.6493741Z ! LaTeX Error: File adjustbox.sty' not found.
2025-01-07T22:34:58.6494172Z
2025-01-07T22:34:58.6494593Z Type X to quit or <RETURN> to proceed,
2025-01-07T22:34:58.6495322Z or enter new name. (Default extension: sty)
```

我特別在搜索 `etoolbox adjustbox language:YAML`，而 GitHub 上的結果有限，只有 53 個 YAML 文件同時包含 `etoolbox` 和 `adjustbox`。我需要一個**完全匹配**。

儘管我們處在大語言模型的時代，搜索完全匹配的需求仍然至關重要。這在檢查某物的確切含義或尋找精確的工作代碼時尤其如此。同樣，像 Google、Twitter 或其他平台也依賴於精確搜索來獲取含義。我們不希望得到 AI 生成的結果或帶有小錯誤的結果。

對於訓練大語言模型，我們可以開發一個系統來找到完全匹配的結果。也許我們可以結合 **KMP (Knuth-Morris-Pratt)** 搜索算法和 **transformer 架構**來增強搜索能力。使用 KMP 與 Transformers 結合可以幫助在特定代碼搜索中找到更準確的結果。

目前，大語言模型無法按文件語言（如 YAML 或 Python）進行過濾。然而，現實世界中的很大一部分信息是以這種方式組織的。這意味著我們可以使用文件來訓練大語言模型。如果我們按文件類型組織所有文本數據，我們可以訓練模型更好地理解它們。因此，對於大語言模型，我們需要在開始時預定義文件語言。默認情況下，它可以是 “文本”，但我們也可以定義其他語言，就像 GitHub 搜索那樣。結果將返回文件，就像 GitHub 搜索結果一樣。

重要的部分是**文件格式**或**擴展名**，而不是文件名。以下是一些例子：

> 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

然而，當用戶的提示混合了普通文本和類似文件的表達式和符號時，執行這樣的搜索就變得困難。例如，在 Stack Overflow 等平台上，問題或答案通常包含混合了代碼片段或文件表達式的文本。

但當然，我們可以想像在這個領域中有新的產品來彌合自然語言搜索和基於文件的搜索之間的差距。