The Cases Where We Still Need the GitHub Search Box

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

     # ...

      - name: Install TeX Live 2023
        if: steps.cache-texlive.outputs.cache-hit != 'true'
        run: |
          # Install dependencies for TeX Live installer
          sudo apt-get update
          sudo apt-get install -y perl wget xz-utils

          # Download TeX Live installer
          wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
          tar -xzf install-tl-unx.tar.gz
          cd install-tl-*/

      # ...

      - name: Install Missing LaTeX Packages
        run: |
          sudo /usr/local/texlive/2023/bin/x86_64-linux/tlmgr install etoolbox adjustbox

      - name: Confirm Package Installation
        run: |
          kpsewhich etoolbox.sty
          kpsewhich adjustbox.sty

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

I am working on the GitHub Actions script above.

I need to search GitHub to find the exact code for `etoolbox adjustbox language:YAML`.

I encountered the following error:

```
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)
```

I'm specifically searching for `etoolbox adjustbox language:YAML`, and the results in GitHub are limited, with only 53 YAML files containing both `etoolbox` and `adjustbox`. I need an **exact match**.

Even though we're in the era of large language models, the need to search for exact matches is still crucial. This is especially true when checking the exact meaning of something or finding precise working code. Similarly, platforms like Google, Twitter, or others rely on exact searches for meaning. We don't want AI-generated results or ones with minor mistakes.

For training large language models, we could develop a system that finds exact matches. Perhaps we can combine the **KMP (Knuth-Morris-Pratt)** search algorithm with **transformer architecture** to enhance search capabilities. Using KMP with Transformers could help in finding more accurate results for specific code searches.

Currently, large language models can't filter by file language like YAML or Python. However, a significant portion of information in the real world is organized this way. This means that we could train large language models using files. If we organize all text data by file types, we can train the model to understand them better. So, for large language models, we would need to predefine the file languages at the start. By default, it could be "text," but we could also define other languages, just like GitHub Search does. The result would return files, just like GitHub search results do.

The important part is the **file format** or **extension**, not the file name. Here are some examples:

> 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

However, when a user’s prompt mixes normal text with file-like expressions and symbols, it becomes difficult to perform a search like this. For example, on platforms like Stack Overflow, questions or answers often contain text mixed with code snippets or file expressions.

But certainly, there are new products we can imagine in this space to bridge the gap between natural language search and file-based search.