GitHub 検索ボックスがまだ必要なケース

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

# ...

      - name: TeX Live 2023 のインストール
        if: steps.cache-texlive.outputs.cache-hit != 'true'
        run: |
          # TeX Live インストーラーの依存関係をインストール
          sudo apt-get update
          sudo apt-get install -y perl wget xz-utils

      # TeX Live インストーラーをダウンロード
      wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
      tar -xzf install-tl-unx.tar.gz
      cd install-tl-*/

# ...

      - name: 不足している LaTeX パッケージをインストール
        run: |
          sudo /usr/local/texlive/2023/bin/x86_64-linux/tlmgr install etoolbox adjustbox

      - name: パッケージのインストールを確認
        run: |
          kpsewhich etoolbox.sty
          kpsewhich adjustbox.sty

      - name: make awesome-cv-copy を実行
        run: make awesome-cv-copy
```

上記の GitHub Actions スクリプトを現在作業中です。

`etoolbox adjustbox language:YAML` の正確なコードを探すために、GitHub を検索する必要があります。

以下のエラーに遭遇しました:

```
2025-01-07T22:34:58.6493408Z
2025-01-07T22:34:58.6493741Z ! LaTeX エラー: ファイル adjustbox.sty が見つかりません。
2025-01-07T22:34:58.6494172Z
2025-01-07T22:34:58.6494593Z 終了するには X を、続行するには <RETURN> を押してください、
2025-01-07T22:34:58.6495322Z または新しい名前を入力してください。（デフォルトの拡張子: sty）
```

私は特に `etoolbox adjustbox language:YAML` を検索しており、GitHub での結果は限られており、`etoolbox` と `adjustbox` の両方を含む YAML ファイルは 53 件しかありません。**完全一致**が必要です。

大規模言語モデルの時代であっても、正確な一致を検索する必要性は依然として重要です。これは、何かの正確な意味を確認したり、正確に動作するコードを見つけたりする際に特に当てはまります。同様に、Google や Twitter などのプラットフォームも、意味を正確に検索することに依存しています。AI が生成した結果や、小さなミスを含む結果は望んでいません。

大規模な言語モデルのトレーニングにおいて、正確なマッチングを見つけるシステムを開発することが考えられます。おそらく、**KMP（Knuth-Morris-Pratt）**検索アルゴリズムと**Transformer アーキテクチャ**を組み合わせることで、検索能力を向上させることができるでしょう。KMP と Transformer を併用することで、特定のコード検索においてより正確な結果を見つけるのに役立つかもしれません。

現在、大規模言語モデルは YAML や Python のようなファイル言語でフィルタリングすることができません。しかし、現実世界の情報の大部分はこのように整理されています。これは、ファイルを使って大規模言語モデルを訓練できる可能性があることを意味します。すべてのテキストデータをファイルタイプごとに整理すれば、モデルがそれらをよりよく理解するように訓練することができます。したがって、大規模言語モデルのために、最初にファイル言語を事前に定義する必要があります。デフォルトでは「テキスト」とすることができますが、GitHub Search が行うように、他の言語を定義することもできます。結果は、GitHub の検索結果と同様に、ファイルを返すことになります。

重要なのは、**ファイル形式**または**拡張子**であって、ファイル名ではありません。以下にいくつかの例を示します:

> 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、アセンブリ、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

しかし、ユーザーのプロンプトが通常のテキストとファイルのような表現や記号を混在させている場合、このような検索を行うのは難しくなります。例えば、Stack Overflow のようなプラットフォームでは、質問や回答にはしばしばコードスニペットやファイル表現が混ざったテキストが含まれています。

しかし、自然言語検索とファイルベースの検索の間のギャップを埋めるために、この分野で想像できる新しい製品が確かにあります。