

오픈소스 ISSUE_TEMPLATE 기반 이슈 검색 솔루션

목차

01 아이디어 고안 배경

02 문제 정의

03 솔루션

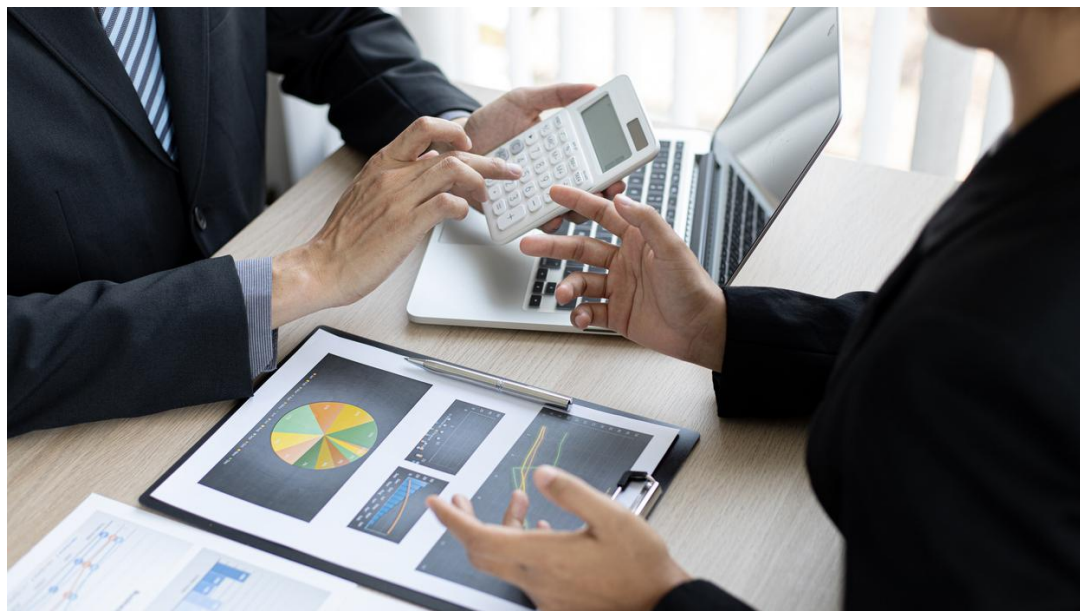
04 솔루션 구현 과정

05 데모

06 마무리

아이디어 고안 배경

오픈소스 소스코드 사용시 에러가 발생했을 때 시도하는 여러 방법들



구글링

장점: 다양한 출처로부터 정보 확인

단점:

- 관련 없는, 혹은 구버전에 대한 해결방안
- 신뢰도 문제: 답변의 신빙성 불분명



LLM

장점: 검색 결과 통합

단점:

- 정확성 및 신뢰도 문제
- Hallucination



Github Issues Search

장점:

- 실제 사례와 해결 방법을 담은 Github Issues 검색 가능
- 소스 코드 관리자로부터 얻는 신빙성 있는 답변

Github Issues에 대해

= Github Repository내 사용자 및 기여자 지원 창구

1. 사용자가 이슈(에러 보고, 기능 요청, 개선 사항 등) 제출
2. 관리자, 사용자들이 읽어보고 댓글을 통한 토의
3. 해결방법 제시 혹은 코드 개선(시스템 내 버그일 때)

즉, 이슈 하나는 모든 핵심 정보(에러 상세 내용, 해결 여부, 해결방법)를 담고 있는 중요한 문서이다

The screenshot shows the GitHub issue template for yt-dlp. At the top, it says "yt-dlp / .github / ISSUE_TEMPLATE / 1_broken_site.yml". Below this is a "Preview" button and a note: "This file is used as an Issue Form template. Give Feedback." The repository name "yt-dlp" and version "Release 2024.11.04" are displayed. The template includes a table with columns: Name, About, Labels, and Assignees. The "About" column contains the text "Report issue with yt-dlp on a supported site". Below the table, there is a section titled "DO NOT REMOVE OR SKIP THE ISSUE TEMPLATE" with instructions to fill all fields. A checklist follows, asking the user to verify that the issue is not already reported, that they have updated to the latest version, and that they have provided the necessary information. The "Region" section is also visible, with "India" selected. At the bottom, there is a section titled "Provide verbose output that clearly demonstrates the problem" with instructions on how to run the command and provide the output.

Name	About	Labels	Assignees
Broken site support	Report issue with yt-dlp on a supported site	triage,site-bug	

DO NOT REMOVE OR SKIP THE ISSUE TEMPLATE
Fill all fields even if you think it is irrelevant for the issue

☐ I understand that I will be **blocked** if I **intentionally** remove or skip any mandatory field *

Checklist
Carefully read and work through this check list in order to prevent the most common mistakes and misuse of yt-dlp:

- ☐ I'm reporting that yt-dlp is broken on a **supported** site *
- ☐ I've verified that I have **updated** yt-dlp to **nightly** or **master** ([update instructions](https://github.com/yt-dlp/yt-dlp#update-channels)) *
- ☐ I've checked that all provided URLs are playable in a browser with the same IP and same login details *
- ☐ I've checked that all URLs and arguments with special characters are [properly quoted or escaped](https://github.com/yt-dlp/yt-dlp/wiki/FAQ#video-url-contains-spc) or [using the escape function](https://github.com/yt-dlp/yt-dlp/wiki/FAQ#video-url-contains-spc)
- ☐ I've searched [known issues](https://github.com/yt-dlp/yt-dlp/issues/3766) and the [bugtracker](https://github.com/yt-dlp/yt-dlp/issues?q=) for similar issues **NOT** to avoid duplication
- ☐ I've read the [guidelines for opening an issue](https://github.com/yt-dlp/yt-dlp/blob/master/CONTRIBUTING.md#opening-an-issue) *
- ☐ I've read about [sharing account credentials](https://github.com/yt-dlp/yt-dlp/blob/master/CONTRIBUTING.md#are-you-willing-to-share-account-details-if-needed) and I'm willing to share it if needed

Region
India
Enter the country/region that the site is accessible from

Provide verbose output that clearly demonstrates the problem

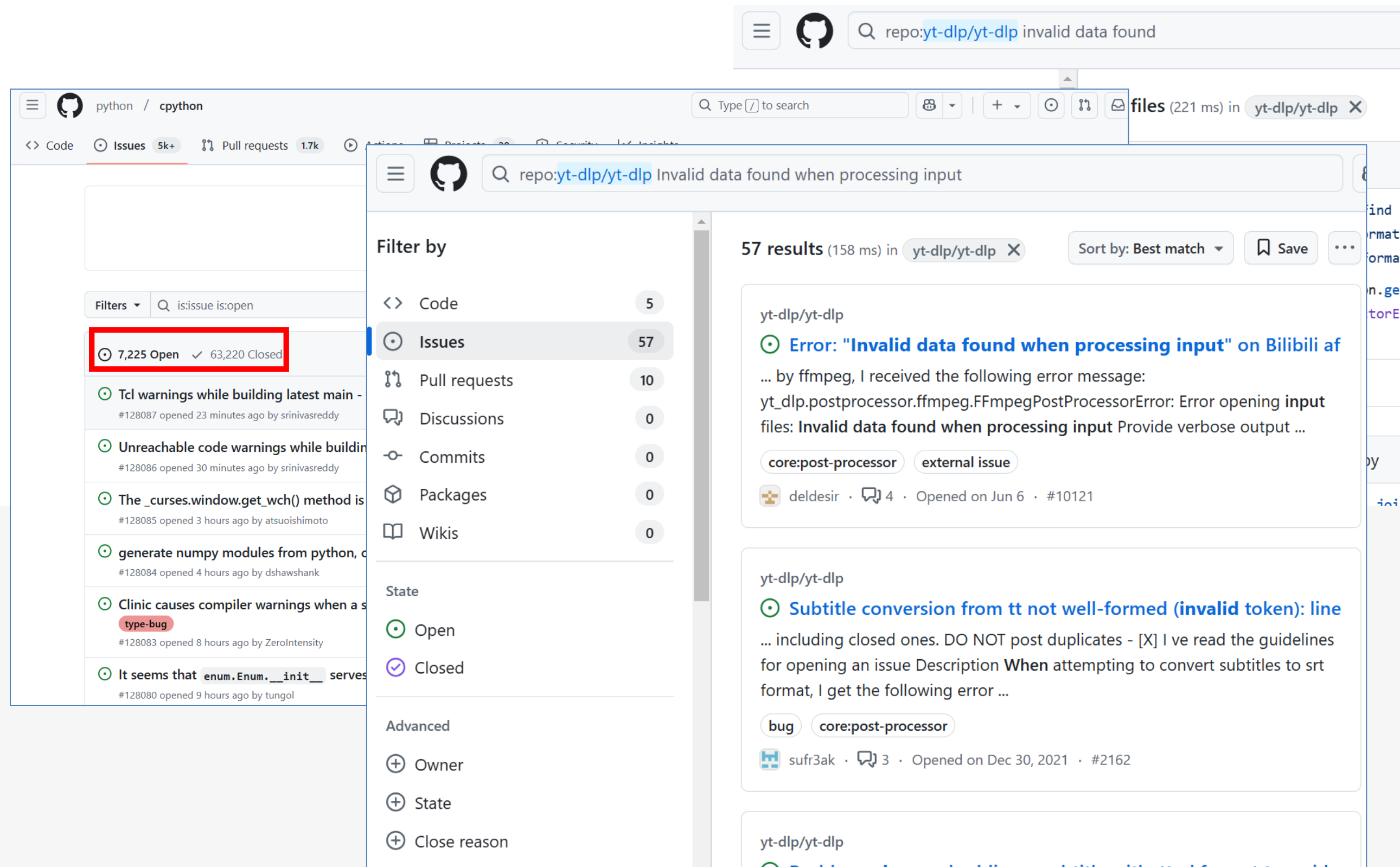
- ☐ Run **"your" yt-dlp command with --verbose** flag added ('yt-dlp -vU <your command line>') *
- ☐ If using API, add **"verbose": True** to 'YoutubeDL' params instead
- ☐ Copy the **WHOLE** output (starting with '[debug] Command-line config') and insert it below *

Complete Verbose Output *

```
[debug] Command-line config: ['-vU', 'https://www.youtube.com/watch?v=BaW_jenozKc']
```

<Yt-dlp/yt-dlp의
ISSUE_TEMPLATE>

Github Issue Search의 한계



검색어 설정에 따라
검색 결과가 크게
달라짐.

검색 결과가
만족스러울 때까지
끝없는 서칭을 해야
함

+ 검색되는 이슈의
수는 레포지토리
크기에 비례

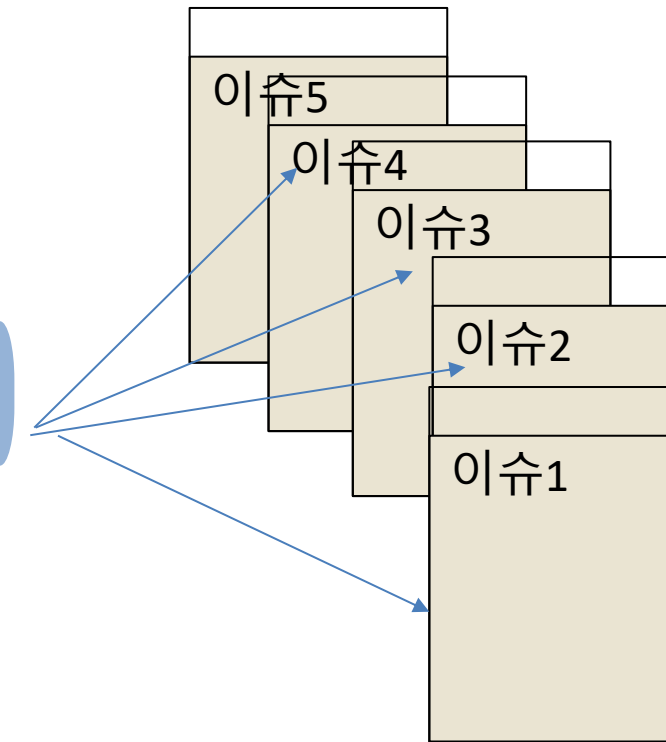
문제 정의

“ 한번의 검색으로 내 에러와 동일한 이슈가 Github에 존재하는지를 찾을 수 있다면? ”

솔루션

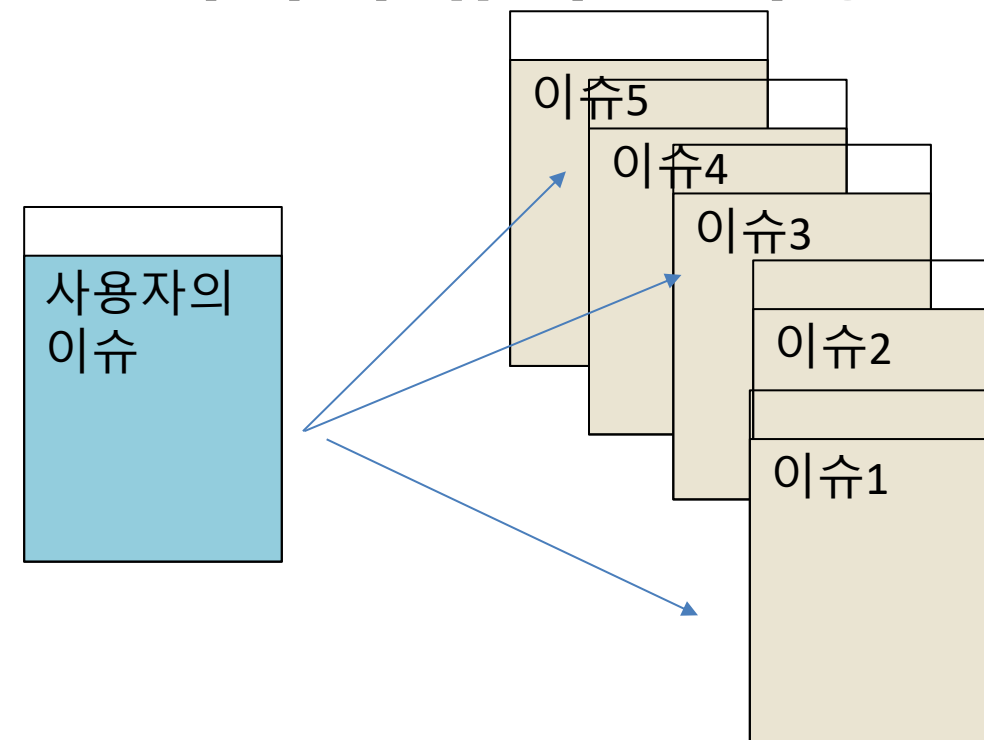
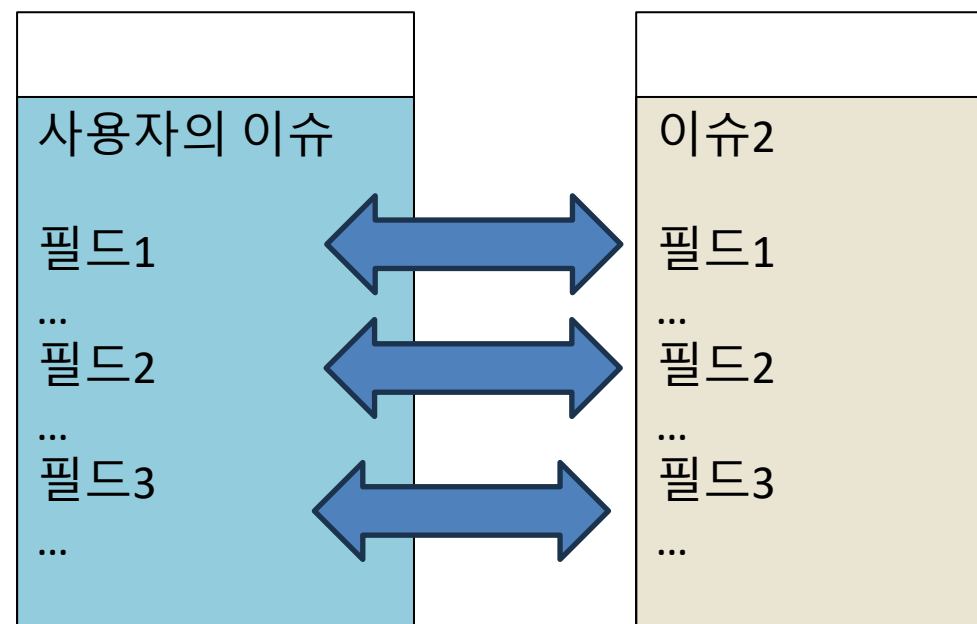
Github Issue의 기존 검색 방식:

특정 단어



Vs .

본 솔루션의 제안 방식 : 이슈 필드별 데이터 유사도 측정



솔루션 구현 과정



1. 솔루션 검색 필드 설계

2. Duplicate issue 검색 알고리즘:

- 0) Input 데이터로부터 검색 키워드 후보를 여러 개 추출
- 1) 각 검색 키워드로 Github Search Issue를 수행하여 관련 이슈들 수집
- 2) 검색된 각 이슈들에 대해서:
 - 데이터 전처리
 - 각 필드별 입력값의 Jaccard 유사도를 계산 및 최종 유사도 도출
- 3) 최종 duplicate issue 선정

3. Duplicate 이슈의 대화내용 분석 및 요약: OpenAI API를 통해 Duplicate 이슈의 문제상황 + 이슈 해결법을 요약 제시.

4. 자동 이슈 템플릿 생성: duplicate 이슈가 없는 경우, 사용자가 레포지토리에 새 이슈를 등록할 수 있도록 redirect

데모 시현

```
(env1) C:\Programming\gsv-app2> yt-dlp -f bestvideo[ext=mp4]+bestaudio[ext=m4a] --merge-output-format mp4 https://www.youtube.com/watch?v=f5Uzch756nc
```

결론 및 제언

01

기대효과: 검색어 다양성 감소

사용자가 검색어를 다양하게 시도하고 그에 따른 검색 결과를 확인하는 과정이 생략된다. 이는 사용자가 여러 번의 검색 시도를 할 필요 없이, 한 번의 검색으로 필요한 정보를 얻을 수 있게 합니다.

02

기대효과: 개발 효율성 증대

즉, 사용자는 오직 에러에 대한 해결 방법이 있는지 없는지에만 집중할 수 있습니다. 이를 통해 에러 해결 과정이 단순화되고, 더 빠르게 문제를 해결할 수 있습니다

03

추후 연구

Input 데이터로부터 검색 키워드 후보를 여러 개 추출하는 코드 개선이 필요

들어주셔서 감사합니다

