

찰떡! 검색어 교정 서비스

CONTENTS

목차

CHAPTER 1

공통과제 소개

CHAPTER 2

공통과제 미션

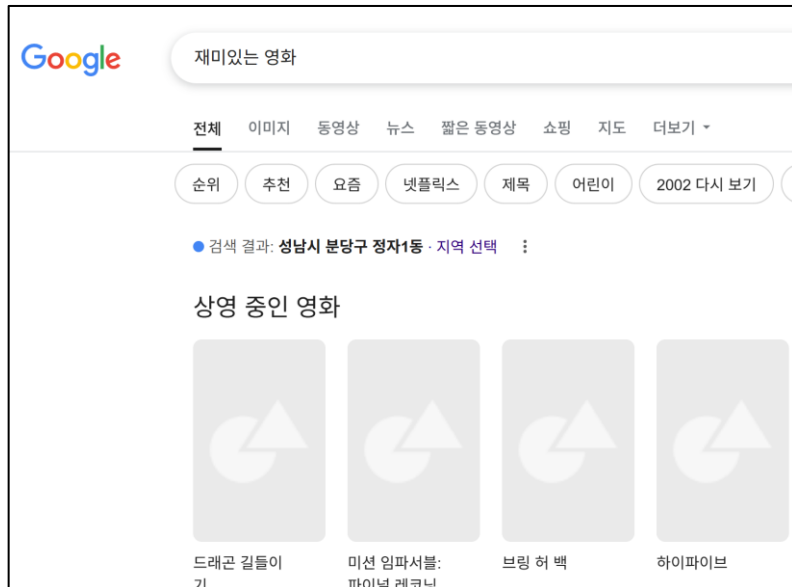
Chapter 01

공통과제 소개

찰떡! 검색어 교정 서비스

찰떡같이 알아듣는 키워드 검색기

검색엔진에서 키워드 검색시, 원하는 키워드를 유추하여 교정한다.
검색어 교정 없이 검색하는 것을 "**원문 키워드 검색**" 이라고 하며,
사용자가 의도한 검색어로 교정해주는 서비스를 "**찰떡 키워드 교정**" 이라고 한다.



키워드 교정없이 원문 그대로 검색
(원문 키워드 검색)



자주 검색되는 키워드로 교정
(찰떡 키워드 교정)

평일과 주말 검색에 따른 차이

평일과 주말 키워드 교정 결과가 달라질 수 있음

- 월 ~ 일요일별 키워드 검색 정보가 주어진다.
- 평일에 키워드 검색시
 - 검색 시도 전까지 평일에 있었던 검색된 정보를 기반으로 키워드 교정
- 주말에 키워드 검색시
 - 검색 시도 전까지 주말에 있었던 검색된 정보를 기반으로 키워드 교정



제공되는 키워드 및 관리 방법

[키워드] [요일]로 채워진 사용자 검색 로그가 제공된다.

- 다음과 같이 관리를 해야한다.
 1. 요일별 키워드 관리 (월요일 ~ 일요일)
 2. 평일 / 주말 별 키워드 관리

교정할 키워드 검색 순서

- 요일별 자주 등장하는 검색어를 기반으로 교정할 키워드를 찾아본다.
- 요일별로 관리되는 검색어 중 교정할 키워드를 찾지 못한 경우, 평일 or 주말별 자주 등장하는 검색어 기반으로 교정할 키워드를 찾아본다.

| | |
|---|----------------|
| 1 | water monday |
| 2 | bread tuesday |
| 3 | potato tuesday |
| 4 | light saturday |
| 5 | shrimp tuesday |
| 6 | pepper monday |



500개 키워드가 제공

키워드 교정기 기본 동작

찰떡! 키워드 교정기 알고리즘은 다음과 같다.

키워드가 지속적으로 입력이 주어진다.

요일별로 최대 10개의 키워드를 관리한다.

- 새로운 키워드에는 기본 점수가 부여된다.
- 10개의 키워드가 모두 채워진 상태에서 새로운 키워드가 주어지는 경우 최저 점수를 가진 키워드와 비교하여 대체될 수 있다.

완벽 HIT

- 새롭게 입력된 키워드가 관리되는 키워드와 원문 그대로 일치한다면 키워드 점수가 증가한다.

찰떡 HIT

- 새롭게 입력된 키워드가 관리되는 키워드에 존재하지 않으면, 다음과 같은 찰떡 알고리즘을 실행한다.
- 관리되는 10개의 키워드 들 중 점수가 가장 높은것 부터 검색한다.
- 유사도 검사 알고리즘으로 측정 결과 80% 이상 동일한 키워드가 발견되면 찰떡 키워드로 확정된다.

완벽 / 찰떡 HIT가 아닐때

- 키워드 관리목록에 추가된다.

키워드 교정기 세부 사항

세부규칙

새로운 키워드에는 UZ점수가 부여된다.

- UZ점수는 **10점**으로 시작되며, 새로운 키워드가 등장할때 마다 UZ점수가 1씩 증가된다.
- 첫번째 키워드는 10점, 두번째 등장하는 키워드는 11점...

관리되는 키워드에 완벽HIT가 될때, 키워드 점수가 증가한다.

- 완벽 HIT 되면 키워드 점수가 10% 더 증가한다.
- 찰떡 HIT는 점수가 증가하지 않는다.

키워드 점수가 21억에 도달하면 점수 재정렬이 일어난다.

- 모든 키워드는 낮은 순위 키워드순서대로 10점부터 1씩 증가되는 점수가 부여된다.
- UZ점수는 10점으로 다시 초기화가 된다.

요일별 키워드가 관리되는 동시에, 평일 / 주말 키워드도 동시에 관리된다.

- 검색 우선순위
 - 요일별 키워드 - 완벽 HIT 확인
 - 평일 / 주말 키워드 - 완벽 HIT 확인
 - 요일별 키워드 - 찰떡 HIT 확인
 - 평일 / 주말 키워드 - 찰떡 HIT 확인

레벤슈타인 거리계산 알고리즘

A, B 문자열이 얼마나 비슷한지 수치적으로 결과를 알려주는 **유사도 검사** 알고리즘

예시

- $f(\text{"kitting"}, \text{"sitting"}) = 85.71\%$ 유사

원리

- A 문자열에서 문자를 몇 번 수정을 해야 B 문자열이 되는지 계산한다.
이를 레벤슈타인 거리 라고 한다.
 - 위 예시에서 리히텐슈타인 거리는 1 이다.
- 이후 유사도를 다음과 같은 공식으로 계산한다.
 - 위 예시에서 최대 문자열의 길이는 7 이므로, 85.71%가 된다.

$$\text{유사도}(\%) = \left(1 - \frac{\text{레벤슈타인 거리}}{\text{최대 문자열 길이}} \right) \times 100$$

Chapter 02

공통과제 미션

찰떡! 검색어 교정 서비스

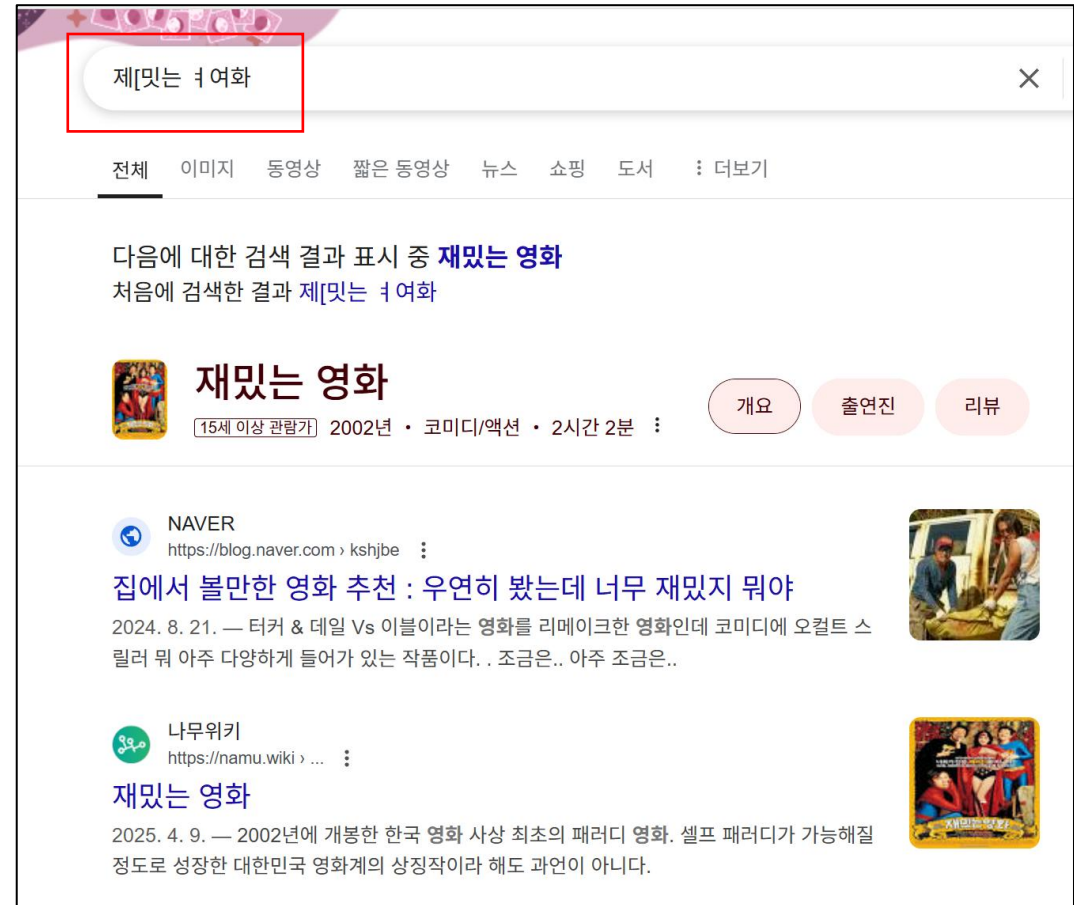
[미션 1] 메서드 레벨 리팩토링

함수 레벨로 클린코드 만들기

요구사항

가독성 좋은 코드
안전성이 좋은 코드

- 안전한 문법 사용
- 순수함수 권장



[미션 2] 클래스 레벨 리팩토링

모듈화가 잘 되어있는 구조로 리팩토링 한다.

요구사항 (D1 ~ D5)

1. 클래스 레벨 리팩토링

D1 - 가독성있는 코드로 메서드 추상화, 가독성 좋은 네이밍 사용

D2 - Regression Test를 위한 Unit Test 개발

2. 확장성을 고려한 설계

D3 - 유사도 검사 방법이 바뀌더라도 기존 코드에 변경이 없도록 한다.

3. 디자인 패턴 사용하기

D4 - 리팩토링에 디자인 패턴을 적용한다.

4. 코드 커버리지 100%

D5 - 리팩토링이 끝난 코드에, 코드 커버리지가 100% 되어야 한다.

제출 방법

개인 Repository 를 Github에 생성한다.
Repository에 다음과 같이 제출한다.

mission1 폴더에는 메서드 레벨 리팩토링 후 코드가 포함 되어야한다.
mission2 폴더에는 클래스 레벨 리팩토링 후 코드가 포함 되어야한다.

README.md 파일에는 다음과 같은 내용이 적혀야 한다.

1. D1 ~ D5 까지 적용 여부 체크리스트
2. Code Coverage 최종 측정결과 캡처화면

| 요구사항 체크리스트 | |
|------------|---|
| D1 | O |
| D2 | O |
| D3 | X |
| D4 | X |
| D5 | O |

체크리스트 예시

평가방법

- D1 ~ D5 까지 체크리스트에 체크가 된 항목은, 수동으로 진행 수준 평가 예정
- D1 ~ D5 까지 체크리스트에 체크가 안된 항목은, 오답처리
- Coverage 캡처화면이 없는 경우, 오답처리

감사합니다.