1차 전처리 정책

작성일: 2024-08-21

작성자: 유정연

목적

본 문서는 수집된 데이터를 단순 전처리를 하기위해 작성된 것으로, 출처 사이트에 따라 작성된 규칙에 따라 위에서 아래순으로 진행하시면 됩니다. 최종 저장 key 명이나 데이터 스키마 포맷에 대해서는 정제 데이터 스키마 의 **Table1** 스키마를 참조 해 주십시오.

Wanted

- **id key** 생성
  - `wanted symbol-company name-job_id` 로 key 값 생성
  - 생성된 key는 `id`로 지정
- **position**
  - 단순 처리만 필요: `특수기호`를 `공백`으로 치환
  - `join` 메소드로 불필요한 공백 제거
- **task**
  - `\/`를 `,`로 치환
  - `\/`를 `,`로 치환
  - `\n` 및 `온점 및 반점을 제외한 특수기호`를 `공백`으로 치환
  - `join` 메소드로 불필요한 공백 제거
- **requirements**
  - `\/`를 `,`로 치환
  - `\/`를 `,`로 치환
  - `\n` 및 `온점 및 반점을 제외한 특수기호`를 `공백`으로 치환
  - `join` 메소드로 불필요한 공백 제거
- **prefer**
  - `\/`를 `,`로 치환
  - `\/`를 `,`로 치환
  - `\n` 및 `온점 및 반점을 제외한 특수기호`를 `공백`으로 치환
  - `join` 메소드로 불필요한 공백 제거
- **due_date**
  - `null`이라면 pass
  - 날짜라면 `datetime`을 이용해 `timestamp`로 변경
- 그 외
  - `wanted soybol` key 값 추가

Programmers

- **id Key** 생성
  - `programmers symbol-companyname-jobcode`로 key 값 생성
  - 생성한 key는 `id`로 지정
- **title**
  - 단순 전처리만 필요: `특수기호`를 `공백`으로 치환
  - `split`, `join` 메소드로 불필요한 공백 제거
- **description**
  - `\r\n`을 `;`로 치환
  - `\n`, `\r`을 `공백`으로 치환
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `;`, `공백`을 제외한 나머지 특수기호를 `공백`으로 치환
  - `split`으로 분리 후, `-`, `+`만 있는 객체를 `list`에서 제외한 후, `join` 메소드로 `공백`을 구분자로 병합
- **requirement**
  - `\r\n`을 `;`로 치환
  - `\n`, `\r`을 `공백`으로 치환
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `;`, `공백`을 제외한 나머지 특수기호를 `공백`으로 치환
  - `split`으로 분리 후, `-`, `+`만 있는 객체를 `list`에서 제외한 후, `join` 메소드로 `공백`을 구분자로 병합
- **preferredExperience**
  - `\r\n`을 `;`로 치환
  - `\n`, `\r`을 `공백`으로 치환
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `;`, `공백`을 제외한 나머지 특수기호를 `공백`으로 치환
  - `split`으로 분리 후, `-`, `+`만 있는 객체를 `list`에서 제외한 후, `join` 메소드로 `공백`을 구분자로 병합
- **jobCategoryIds**
  - `programmers` 내 자체 id 코드 리스트 이용해서 `id`값에 맞는 직무 유형을 `list` 형식으로 기입
- **updatedAt**
  - `년-월-일`값만 추출하여 `datetime`으로 `timestamp`로 변경
- **endAt**
  - `null`이라면 pass
  - 날짜라면, `년-월-일`값만 추출하여 `datetime`으로 `timestamp`로 변경
- **careerRange**
  - `null`이라면 신입이므로, FALSE 값 지정
  - `null`이 아니라면 경력이므로, TRUE로 값 지정
- **resumeRequired**
  - true 면, true 값 지정
  - false면, false 값 지정

- **isAppliable**
  - true 면, true 값 지정
  - false면, false 값 지정
- 그 외
  - `programmers symbol` 값 추가

Rocket-punch

- **id Key** 생성
  - `rocket-punch symbol-company_name-job_id`로 key 값 생성
  - 생성한 key는 `id`로 지정
- **job_task**
  - 문자열 길이가 0이거나 null이면 pass
  - 아닐 경우 아래를 따름:
    - `\/`를 `/`로 치환
    - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `(`, `)`, 공백을 제외한 나머지 특수기호를 공백으로 치환
    - `split`으로 분리 후, `-`, `+`만 있는 객체를 `list`에서 제외한 후, `join` 메소드로 공백을 구분자로 병합
- **job_specialties**
  - `\/`를 `/`로 치환
- **job_detail**
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `(`, `)`, 공백을 제외한 나머지 특수기호를 공백으로 치환
  - `split`으로 분리 후, `-`, `+`만 있는 객체를 `list`에서 제외한 후, `join` 메소드로 공백을 구분자로 병합
- **job_industry**
  - `\/`를 `/`로 치환
- **date_start**
  - `datetime`을 사용하여 `timestamp`로 변경
- **date_end**
  - `null`이면 pass
  - `null`이 아니라면, `datetime`을 사용하여 `timestamp`로 변경
- **job_career**
  - `list`안에 "경력"이라는 문자열이 있다면 true
  - "경력" 문자열이 없거나, "신입" 문자열이 존재할 경우, false
- 그 외
  - `rocket-punch symbol` 값 추가

Jobkorea

- **id Key** 생성
  - `jobkorea symbol-company-job_id`로 key 값 생성
  - 생성한 key는 `id`로 지정
- **title**
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `(`, `)`, `공백`을 제외한 나머지 특수기호를 `공백`으로 치환
  - `split`으로 분리 후, `join` 메소드로 `공백`을 구분자로 병합
- 모집분야
  - `\/`를 `/`로 치환
  - 온점 `.`, 반점 `,`, `/`, `-`, `+`, `(`, `)`, `공백`을 제외한 나머지 특수기호를 `공백`으로 치환
  - `split`으로 분리 후, `join` 메소드로 `공백`을 구분자로 병합
- 스킬
  - `null`이면 pass
  - `null`이 아닐 경우, `\/`를 `/`로 치환
- 산업
  - `\/`를 `/`로 치환
- 주요사업
  - `\/`를 `/`로 치환
- 시작
  - 문자열을 파싱하여 `datetime`으로 `timestamp`로 변환할 수 있는 형태로 정제한 뒤, `timestamp`로 변경
- 마감
  - 문자열을 파싱하여 `datetime`으로 `timestamp`로 변환할 수 있는 형태로 정제한 뒤, `timestamp`로 변경
- 경력
  - 문자열에 `신입`이 포함되어 있을 경우 false
  - 문자열에 `경력`이 있을 경우 true
- 이력서
  - `null`이 아니라면 true, `null`이라면 false
- **target_url**
  - `\/`을 `/`으로 치환
- 그외
  - `jobkorea symbol` 추가