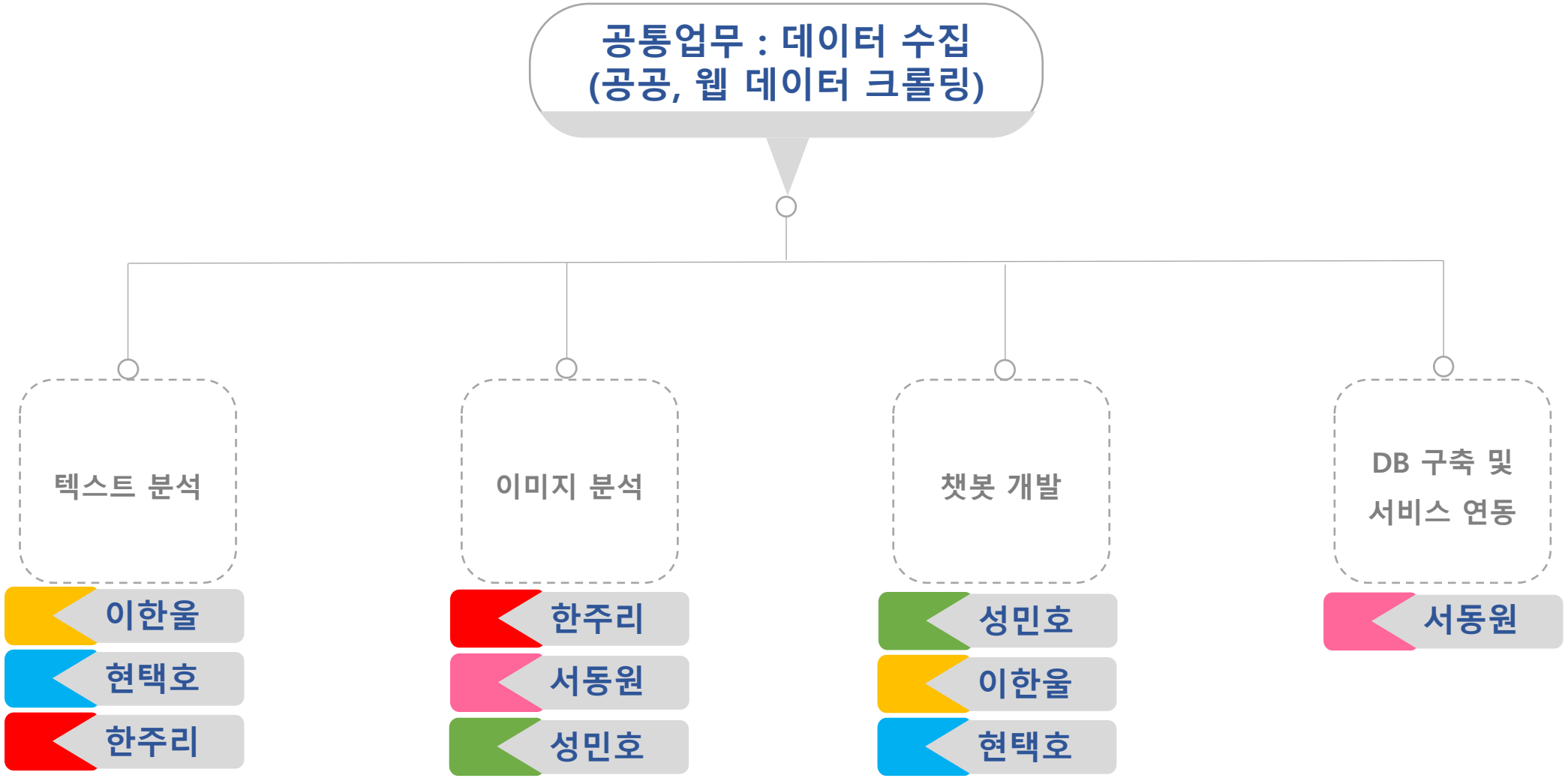
마리톡
데이터 수집 및 전처리 방안

~EatingElmo 자료집과 함께 보면 더 좋은~

서동원

2020.03.26.(목)

업무 분장 및 팀 구성도


공통업무 : 데이터 수집
(공공, 웹 데이터 크롤링)
텍스트 분석
이한울
현택호
한주리
이미지 분석
한주리
서동원
성민호
챗봇 개발
성민호
이한울
현택호
DB 구축 및
서비스 연동
서동원

목차

- 활용 데이터
- 수집 과정
- 전처리 과정

활용 데이터

1. 공공 데이터
   - 소스 : 서울 열린데이터 광장
   - 데이터 : 서울시 25개 구별 미용실 등록현황

2. 크롤링 데이터
   - 소스 : 네이버, 다음, 티스토리, 페이스북, 인스타그램, 트위터
   - 데이터 :
     1) 블로그 및 SNS에 올라온 리뷰
     2) 텍스트 및 헤어이미지

수집과정 - 공공 데이터

- ~~현재는 직접 다운로드 받은 csv 파일로 전처리 작업을 하되~~
- 어느정도 가닥이 잡히면 API로 데이터를 받아와 처리할 것

- 현재 API로 데이터를 받아오는 코드 구현 완료된 상태
  - API로 전체 데이터 호출 코드 실행하는데 3~5분정도 걸린다.

수집과정 - 크롤링 데이터(공통사항)

- 공통사항
  - 텍스트/이미지 분석을 위한 데이터 수집
  - BeautifulSoup / Selenium 모듈 활용

  - BeautifulSoup :

    python 코드 상에서 html, css, content(text/image) 등 정적 컨텐츠 크롤링 담당
  - Selenium :

    python 코드 상에서 웹브라우저 접속, 검색, 링크이동 등 동적 처리 담당

수집과정 - 크롤링 데이터(텍스트 데이터)

- 수집방안
  - BeautifulSoup / Selenium 모듈 활용

  1) 크롤링할 플랫폼 정하기
     - 블로그 : 네이버, 다음, 티스토리
     - SNS : 페이스북, 인스타그램, 트위터
  2) 선정한 플랫폼에서 리뷰 데이터 url(링크) 수집
  3) url 및 글 제목 바탕으로 1차 필터링
  4) 1차 필터링된 링크의 개수로 크롤링할 데이터 양 파악
  5) 정제된 링크로 실질적인 리뷰 본문(텍스트) 수집

  * 이 과정에서 리뷰에 삽입된 이미지 데이터도 같이 크롤링 하면 어떨까?

수집과정 - 크롤링 데이터(이미지 데이터)

- 수집방안1
  - BeautifulSoup / Selenium 모듈 활용
  1) 크롤링할 플랫폼 정하기
     - 블로그 : 네이버, 다음, 티스토리
     - SNS : 페이스북, 인스타그램, 트위터
  2) 이전 페이지에서 확보한 리뷰 링크에서 이미지 데이터 수집

- 수집방안2
  1) 사람들이 자주 검색할만한 특정 헤어스타일 선택하기
     - ex) 투블럭, 박새로이컷, 허쉬컷, C컬펌, 씨스루뱅, 애쉬브라운 등등...
  2) 해당 헤어스타일로 검색 -> 검색결과 이미지 수집
  3) 수집방안1과 수집방안2의 데이터를 함께 이미지 분석에 활용

전처리 과정 – 공공 데이터
- 완료된 작업

1. API 방식으로 공공 데이터 조회
2. 25개 데이터를 pandas DataFrame(이하 DF로 표기)으로 변환
3. 25개 DF의 컬럼이 길이 및 컬럼명이 같은지 확인
4. 컬럼명을 일치시킨 후 DF 병합
5. drop_duplicated()함수로 중복된 데이터 제거
6. ‘폐업일자’ 컬럼에 값이 있는 데이터를 폐업점으로 판단.
   폐업점 데이터 제거

전처리 과정 – 공공 데이터
- 필요한 작업1 : 쓰레기 데이터 제거

- 전체값이 중복되진 않지만 제거해야할 데이터를 판단해야 함

1. 폐업했으나 미신고하여 남아있는 경우
   - Selenium, BeautifulSoup으로 공공데이터 '업소명"을 네이버에 검색
   - 업소명, 전화번호, 주소 등을 비교하여 일치하는 정보만 남긴다.
   - 나머지 불일치 데이터 모두 삭제

2. 매장이전 또는 상호변경 후 이전 데이터가 삭제되지 않은 경우
   - "소재지시작일" 을 비교하여 최신 데이터만 남기기
   - Selenium, BeautifulSoup으로 공공데이터 '업소명'을 네이버에 검색
   - 업소명, 전화번호, 주소 등을 비교하여 일치하는 정보만 남긴다.
   - 나머지 불일치 데이터 모두 삭제

3. 같은 매장인데 층 수를 따로따로 신고한 경우
   - Selenium, BeautifulSoup으로 공공데이터 '업소명'을 네이버에 검색
   - 면적이 넓은 곳만 남기기? 소재지 시작일로 비교?
   - 그래도 판단하기 애매하면 어떻게 처리하지????? 아이디어 내주세요

전처리 과정 – 공공 데이터
- 필요한 작업2 : 누락값 채우기

- 누락데이터(2020.03.18. 기준)
  1. 전화번호
     - 6871개
  2. 행정동명
     - 47개
  3. 소재지도로명
     - 65개
  4. 소재재지번
     - 48개

- 해결방안
  - BeautifulSoup/Selenium을 활용하여 네이버에 업소명 검색
  - 네이버 지도에 등록된 정보를 크롤링하여 누락값 채우기

```
In [305]: len(df_all[df_all['업소명'].isnull()])
Out[305]: 0

In [295]: len(df_all[df_all['전화번호'].isnull()])
Out[295]: 6871

In [297]: len(df_all[df_all['영업자시작일'].isnull()])
Out[297]: 0

In [298]: len(df_all[df_all['소재지시작일'].isnull()])
Out[298]: 0

In [299]: len(df_all[df_all['행정동명'].isnull()])
Out[299]: 47

In [300]: len(df_all[df_all['업태명'].isnull()])
Out[300]: 0

In [302]: len(df_all[df_all['허가(신고)번호'].isnull()])
Out[302]: 0

In [303]: len(df_all[df_all['소재지도로명'].isnull()])
Out[303]: 65

In [304]: len(df_all[df_all['소재지지번'].isnull()])
Out[304]: 48
```

전처리 과정 – 리뷰 포스팅 개수 데이터

(네이버 기준)
EatingElmo(32p) 참고

- 포스팅 개수 파악 후 데이터프레임에 포스팅수 컬럼 추가
- EatingElmo 팀의 자료에서 검색창에 1,000개를 초과하는 블로그 검색결과가 안나오는 문제를 확인
- EatingElmo 팀은 포스팅수가 1,000개가 넘으면 1,000개까지만 숫자를 맞추고 그 이상 넘어서는 포스팅(링크)는 수집하지 않음

  -> 추후 확인 필요

전처리 과정 – 리뷰 포스팅 개수 데이터(2)

(네이버 기준)
EatingElmo(34p) 참고

- EatingElmo피셜
- 데이터들을 직접 살펴보니 광고성 게시물들이 대체로 뒤쪽에 배치 -> 아마 네이버에서 자체적인 필터링이 있다고 판단
- 따라서 전체 데이터의 75% 정도만 크롤링에 반영하기로함
- 리뷰 포스팅 개수 = 리뷰 포스팅 개수 * 0.75

전처리 과정 – 업소별 검색창 링크 수집

(네이버 기준)

- EatingElmo(42p) 참고
- 검색어 예시 : 지역명 + 업소명 + "업소명" + 후기
- 블로그 링크 자체를 수집하는 것이 아니라 "검색을 할 링크"를 수집하는 단계

- 각 업소별 검색할 링크를 수집완료했으면
- 이제 각 블로그 링크를 크롤링할 준비가 된 것

전처리 과정 – 블로그 링크 수집(1)

(네이버 기준)

- EatingElmo(45p) 참고
- 수집한 업소별 검색창 링크를 활용하여 블로그 링크를 수집
- 수집하기 전 크롤링에 걸리는 시간을 미리 추산

- Python 라이브러리 tqdm 또는 Jupyter notebook 자체 명령어 magic method 인 %%time을 이용하여 추산

- 맛집 데이터 기준 1,000개의 가게별 블로그 링크를 수집하는데 4~6시간 소요

-> 데이터를 분할하여 팀원들의 각자의 컴퓨터에서 블로그 링크 크롤링 진행

전처리 과정 – 블로그 링크 수집(2)

(네이버 기준)

- EatingElmo(47p / 49p) 참고
- 블로그 링크를 크롤링을 하여 저장할 데이터
  1) id
  2) 포스트 제목
  3) 작성일자
  4) 작성자
  5) 본문요약(미리보기)
  6) 블로그 링크

전처리 과정 – 블로그 링크 필터링

(네이버 기준)

- EatingElmo(50p ~ 64p) 참고

1) 블로그 작성자로 필터링
2) 블로그 링크로 필터링
3) 블로그 타입(플랫폼)별 필터링
4) 특정 태그 및 키워드로 필터링

전처리 과정 – 블로그 링크 필터링
- 블로그 작성자로 필터링

(네이버 기준)

- EatingElmo(50p) 참고

- 일일이 수작업으로 광고성 작성자를 걸러내는 작업은 한계가 있다.

전처리 과정 – 블로그 링크 필터링

- 블로그 링크로 필터링

(네이버 기준)

- EatingElmo(52p) 참고

- 제각기 다른 음식잠(우리의 경우 미용실)으로 검색했는데 base url이 여러번 중복된다면? -> 어그로/스팸/낚시성 작성자일 확률이 높다.

전처리 과정 – 블로그 링크 필터링
- 블로그 타입(플랫폼)별 필터링

(네이버 기준)

- EatingElmo(58p / 59p) 참고

- Naver, Duam, Tistory 등 네이버에서 검색을 해도 네이버 이외의 플랫폼의 검색결과가 나온다.

- 각 플랫폼별 데이터 수를 확인하여 너무 작은 플랫폼은 과감하게 버릴것인지 판단

전처리 과정 – 블로그 링크 필터링
- 특정 태그 및 키워드로 필터링

(네이버 기준)

- EatingElmo(58p / 59p) 참고

- 특정 스팸 키워드를 지정하여 키워드가 일정 회수이상 포함되는 링크를 버린다.

전처리 과정 – 블로그 링크 필터링
- 주의사항(접속에러)

(네이버 기준)

- EatingElmo(64p) 참고

- EatingElmo 피셜
- 특정 url에 접속이 안되는 에러 발생
- url 중간에 특정 문자열을 제거하고 “/” 문자로 대체하면 접속이 잘 되는 것을 확인

  -> 추후 확인 필요

전처리 과정 – 블로그 본문 수집

(네이버 기준)

- EatingElmo(65p) 참고

- 링크 수집이 완료 -> 데이터를 분할하여 팀원들에게 골고루 분배
- EatingElmo의 경우 링크수집까지 윈도우 환경에서 크롤링하였고
- 본문수집부터 GCP(Google Cloud Platform) 의 VM Linux 환경에서 진행하였다.

  -> 논의 필요