웹 데이터 크롤링 및 DB 구축

구분	연구
기간	@2022년 7월 4일 → 2022년 12월 31일

연구 과제명

디지털 인문사회 교육 및 연구 상호 피드백 모델 / KAIST 디지털인문사회과학부 김하나 교수

기획 의도

- 데이터 수집 자동화 모델 개발
- N포털 사이트 뉴스 기사 중, '에너지 전환' 키워드를 포함한 기사 데이터 및 댓글을 수집합니다. (2021.7.1 ~ 2022.7.31 기간 내)
- 댓글 작성 유저가 과거 다른 뉴스 기사에 작성한 댓글 히스토리를 모두 수집합니다. 추후 유저 특성 분석에 활용될 DB를 생성합니다.

팀 구성 및 역할

1인 프로젝트

- 데이터 수집 및 정제
- DB 스키마 설계

개발 환경

- 언어 : Python
- Tools : ERD
- 활용 라이브러리 : Pandas Selenium BeautifulSoup4 Requests

성과

- 2,016개 기사에 대한 댓글, 댓글 히스토리 데이터 셋 총량 121.2GB
- 웹 데이터 크롤링 알고리즘 개선 통해 기존 소요 시간 대비 **1,130%** 수집 속도 향상 (아래 '데이터 수집 속도 향상' 방법 활용)

데이터 수집

- Python환경에서 Selenium와 BeautifulSoup4 라이브러리를 활용하였습니다.
- requests 라이브러리를 통해 데이터를 파싱하였습니다.
- 반복적으로 웹사이트에 방문하여 데이터를 불러오는 과정에서, Browser Memory Overflow 문제로 인해 반복적으로 'Out of Memory' 오류가 발생했습니다.
  이는 다음과 같은 방법을 고안하여 해결했습니다.

  Selenium 'Out of Memory' 해결법

데이터 수집 속도 향상

- Selenium의 webdriver 설정 시 아래 옵션을 설정함으로써 크롤링 속도를 더욱 향상 시킵니다.

▼ headless

브라우저 창을 열지 않고 selenium을 실행함으로써 속도를 향상시키기 위해 사용하였습니다.

▼ disable-gpu

windows 환경에서 google-chrome-headless를 활성화하기 위해 사용하였습니다.

▼ disable-extension

추가 확장 프로그램을 사용하지 않도록 함으로써 속도를 향상시키기 위해 사용하였습니다.

▼ blink-settings=imagesEnabled=false

웹 페이지 상에 존재하는 이미지 파일을 로딩하지 않음으로써 속도를 향상시키기 위해 사용하였습니다.

- Selenium 사용 시 Element Load가 완료되는 시점에 바로 작업이 수행될 수 있도록(최적화) 다음 세 가지를 함수를 활용합니다.

  ▼ WebDriverWait

  특정 Element가 Load되는 순간까지만 대기하는 함수로써 가장 최적화된 방법이므로 주로 사용하였습니다.

  ▼ time.sleep()

  Element의 id 값이 중복되어 가장 마지막 Element까지 Load되는 시점을 명확히 할 수 없을 때 명시적으로 대기하기 위해 사용하였습니다. (실험을 통해 페이지 로드가 정상적으로 완료되는 Optimal 값을 설정)

  ▼ driver.implicitly_wait()

  찾고자 하는 Element가 페이지 중간에 있을 경우 불필요한 대기를 수행하게 되는 단점이 있으나, time.sleep과 달리 페이지 로드가 완료되는 때까지만 대기하므로 더 효율적입니다.

- API Request 이용해 Data Load 소요 시간을 기존 대비 **45%** 단축합니다.
  - Selenium 환경에서 Button Element를 클릭하고 Data Load가 완료되기까지는 오랜 시간이 걸리기 때문에 이때는 Selenium을 활용하는 대신 API를 직접 호출하여 시간을 단축합니다.

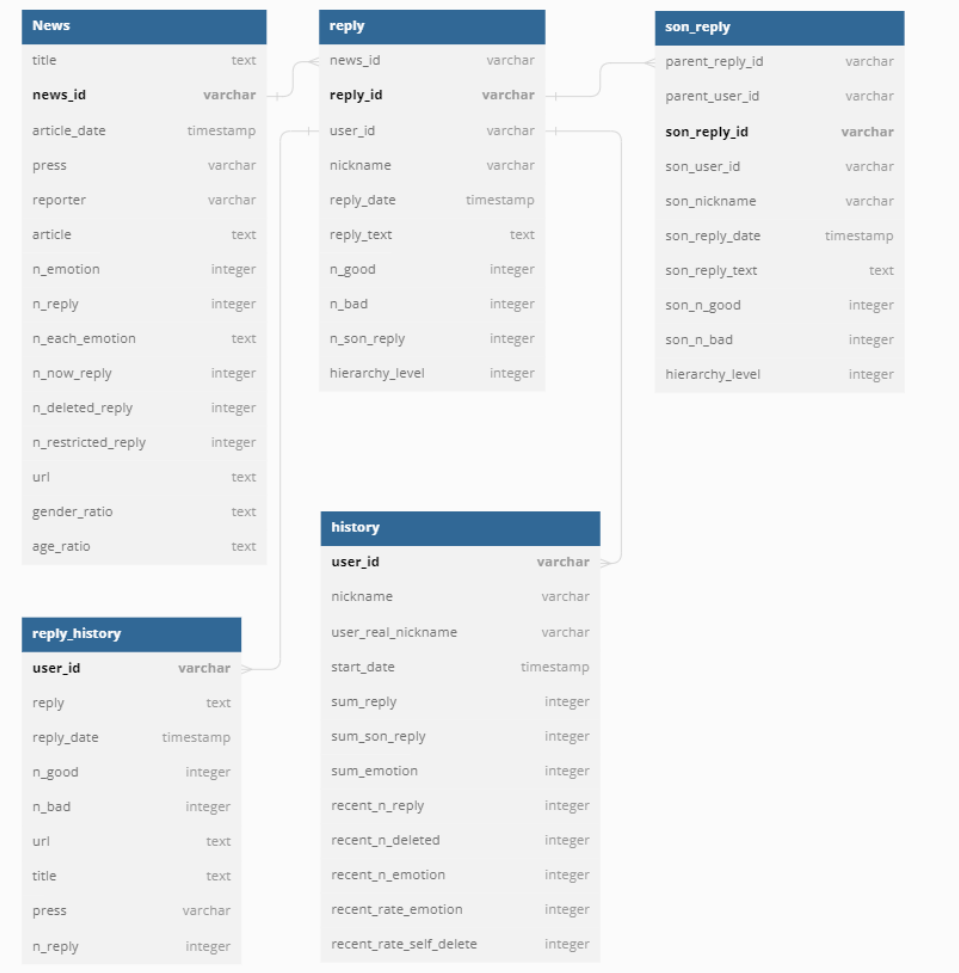
- MultiProcessing 이용해 대용량 데이터를 약 **6.3배** 더 빠르게 처리합니다.
  - Server PC의 CPU Process Capa를 고려하여 과부하가 걸리지 않는 최적 Process 수(8개)를 설정하여 진행합니다.

데이터 수집 과정 및 결과

- 데이터 수집과 동시에 모든 텍스트에 대해 전처리를 수행합니다.
- 데이터는 Pandas 라이브러리를 통해 DataFrame 구조에 저장하여 처리합니다.

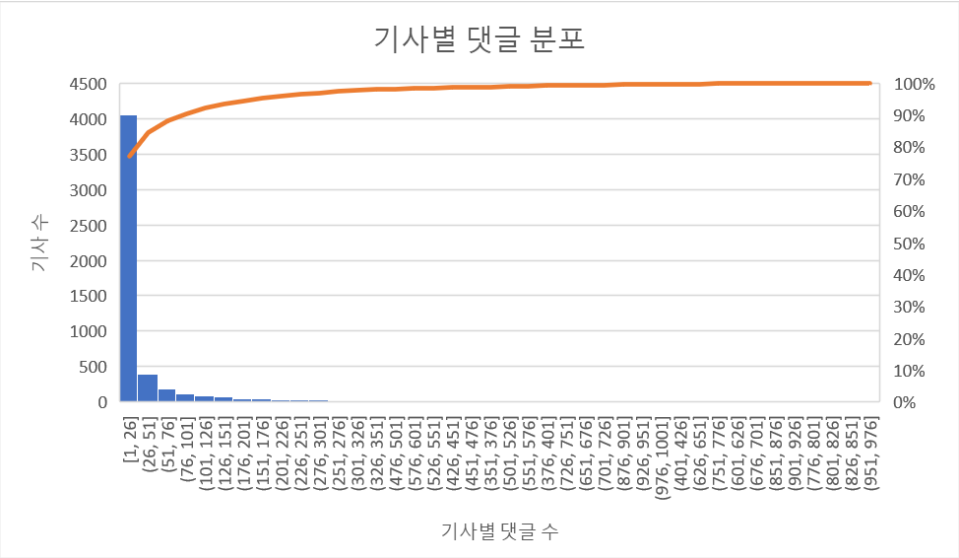
- **단계별 수행 내역**

Step 1. 데이터 수집 목적 정립, 데이터 구조(스키마) 설계



Step 2. 기간(2021.7.1 ~ 2022.7.31)내의 모든 NAVER 뉴스 기사 중, 특정 키워드('에너지 전환')가 포함된 기사들의 '제목, 작성 일자, URL, 댓글 수' 수집

Step 3. 100건 이상의 댓글이 존재하는 뉴스 기사 선정



댓글이 유의미하게 많지 않아서 추후 이루어지는 유저 유형 분석에 활용될 수 없는 기사는 제외합니다.

Step 4. 선정된 기사에 대한 크롤링은 다음 4단계로 진행

▼ 기사의 기본 정보 수집 (Table['News'])

수집 정보 : 제목, 작성 일자, 신문사, 기자명, 기사 본문, 총 기사 반응 수, 기사 반응 별 수, 총 댓글 수, 현재 댓글 수, 삭제된 댓글 수, 제한된 댓글 수, 기사 URL, 댓글 성비, 댓글 연령비

▼ 기사의 댓글 데이터 수집 (Table['reply'])

기사의 '댓글 더보기' 버튼을 클릭하여 모든 댓글을 펼친 후, 데이터를 수집합니다.

수집 정보 : 해당 댓글의 ID, 작성자 ID, 작성자 닉네임, 작성 일자, 댓글 내용, 좋아요 수, 싫어요 수, 대댓글 수

▼ 댓글을 작성한 모든 유저에 대한 정보 수집

- 유저의 활동 데이터 (Table['history'])

  수집 정보 : 작성자 ID, 활동 시작 일시, 현재 까지 작성한 모든 댓글 수, 작성한 댓글에 달린 답글 수, 공감 수, 최근 작성 댓글 수, 최근 삭제 댓글 수, 최근 공감 수, 최근 받은 공감률

- 과거 다른 기사에 작성한 댓글 히스토리 (Table['reply_history'])

  수집 정보 : 댓글 내용, 일자, 반응 수, 기사 URL, 기사 제목, 신문사, 해당 기사의 총 댓글 수

cf. 댓글이 과도하게 많은 헤비 유저의 경우 과거 댓글 수가 많아 selenium이나 BeautifulSoup로 크롤링하기에 시간이 오래 걸리므로, Naver 뉴스 기사 고유의 oid, aid, commentNo를 파라미터로 설정하여 직접 데이터를 파싱하여 자료를 수집하였습니다.

▼ 모든 댓글들에 대해 댓글별 자식 댓글 데이터 추가 수집

- 자식 댓글 데이터 (Table['son_reply'])

  수집 정보 : 부모 댓글의 ID, 자식 댓글의 ID, 자식 댓글 작성자 ID, 자식 댓글 작성일자, 댓글 내용, 좋아요/싫어요 수

Step 5. 위 Step 3에서 선정된 모든 기사들에 대해 Step 4 수행

결과 예시

대용량 데이터를 빠른 속도로 처리하고, 제약 없는 데이터 추가를 위해 Document-Oriented NoSQL (Json) 형식으로 구조화합니다. 이는 추후 MongoDB등 NoSQL 기반 DBMS를 통해 처리할 수 있습니다.

```
{
  "title": "'RE100' 공방전... 李 \"모른다고 상상하지 못해\", 尹 \"어려운 거 설명해야 예의\"",
  "article_date": "2022.02.04. 오후 9:02",
  "press": "한국일보",
  "reporter": "강은영 기자",
  "article": "대선후보 첫 TV토론 이후 'RE100' 논란 이재명 \"단어 문제 아니라 국가 산업전환 핵심과제\"윤석열 \"대통령 될 사람 모를 수도 있는 거 아닌가\"AI윤석열
  "n_emotion": "774",
  "n_reply": "790",
  "n_each_emotion": {
    "좋아요": "30",
    "훈훈해요": "1",
    "슬퍼요": "2",
    "화나요": "739",
    "후속기사원해요": "2"
  },
  "n_now_reply": "585",
  "n_deleted_reply": "203",
  "n_restricted_reply": "2",
  "url": "https://n.news.naver.com/mnews/article/469/0000656329?sid=100",
  "gender_ratio": {
    "남자": "79",
    "여자": "21"
  },
  "age_ratio": {
    "10대": "0",
    "20대": "4",
    "30대": "9",
    "40대": "35",
    "50대": "31",
    "60대 이상": "20"
  },
  "dic_reply": [
    {
      "reply_id": "*",
      "user_id": "*",
      "nickname": "*",
      "reply_date": "2022.02.06. 09:02:40",
      "reply_text": "이건 알지.!!!",
      "n_good": "0",
      "n_bad": "0",
      "n_son_reply": "0",
      "hierarchy_level": "1",
      "dic_son_reply": "",
      "dic_history": {
        "user_id": "*",
        "nickname": "*",
        "user_real_nickname": "*",
        "start_date": "2018.02.06.",
        "sum_reply": "6,867",
        "sum_son_reply": "32",
        "sum_emotion": "33,238",
        "recent_n_reply": "94",
        "recent_n_deleted": "0",
        "recent_n_emotion": "444",
        "recent_rate_emotion": "83%",
        "recent_rate_self_delete": "0%",
        "dic_reply_history": [
          {
            "text": "*",
            "date": "2022-10-11 01:19:06",
            "good": "4",
            "bad": "1",
            "url": "https://n.news.naver.com/mnews/article/021/0002535100?sid1=001",
            "title": "'나는 독일인입니다'도 문프셀러 등극? “청산하지 못한 역사의 상처 공감”",
            "press": "문화일보",
            "n_reply": "1041"
```


※ 개인을 특정할 수 있는 정보는 임의의 값으로 대체하였습니다.