

고급 SW 빅데이터 과정

웹 크롤링

핵심 소개

프로젝트 명 : 고급 SW 빅 데이터 과정

프로젝트 기간 : 2018.7.16 ~ 2018.7.27

팀 인원 : 개인 프로젝트

담당 역할 :

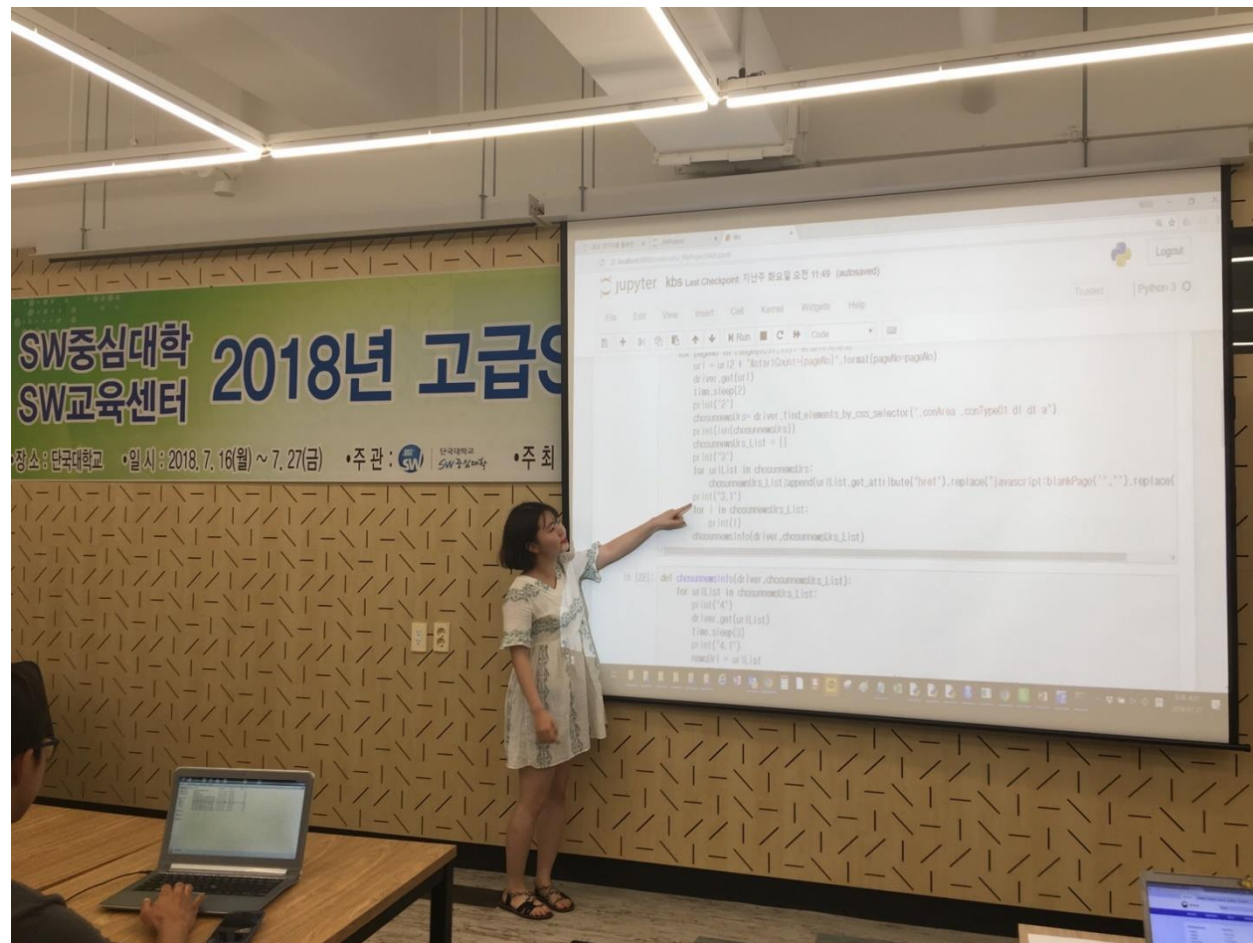
- 파이썬을 이용한 뉴스 기사 텍스트 마이닝(웹 크롤링)
- 텍스트 마이닝 후 해당 내용 DB에 저장.

기획 의도 : 뉴스 기사 단어 빈도수 분석

02 개발 환경과 구축

: My Sql, 파이썬 사용

- 파이썬 BeautifulSoup 라이브러리 이용
- My Sql DB에 텍스트 마이닝한 내용 저장
- 신문사별로 뉴스 구성이 다르기 때문에 신문사별 텍스트 마이닝 파일 생성



연합 뉴스

- 검색 페이지 단위로
기사 내용을 가져오는
코드

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import pymysql

def yonhapnewsUrl(search):
    url = "http://www.yonhapnews.co.kr/home09/7091000000.html?query={search}&ctype=A".format(search=search)
    driver = webdriver.Chrome("./chromedriver.exe")
    driver.get(url)

    yonhapnewsUrlList(driver)

def yonhapnewsUrlList(driver):
    url2 = driver.current_url
    yonhapnewsUrs_List = []
    for pageNo in range(1,3): #2페이지까지
        url = url2 + "&page_no={pageNo}".format(pageNo=pageNo)
        driver.get(url)

        yonhapnewsUrs= driver.find_elements_by_css_selector(".cts_atclst li a")

        for urlList in yonhapnewsUrs:
            yonhapnewsUrs_List.append(urlList.get_attribute("href"))
        yonhapnewsInfo(driver, yonhapnewsUrs_List)

def yonhapnewsInfo(driver, yonhapnewsUrs_List):
    for urlList in yonhapnewsUrs_List:
        driver.get(urlList)
        newsTitle = driver.find_element_by_css_selector(".tit-article").text
        newsSubtitles = driver.find_elements_by_css_selector(".stit strong")
        newSubtitle_result = ""
        for newsSubtitle in newsSubtitles:
            newSubtitle_result+=newsSubtitle.text
        newsContents = driver.find_elements_by_css_selector(".article p")
        print("- 기사 제목 - : \n",newsTitle)
        print("- 기사 부제목 - \n:",newSubtitle_result)
        print("- 기사 본문 - \n")
        content_result = ""
        for content in newsContents:
            content_result+=content.text
        print(content_result)

        dbData = [[newsTitle,newSubtitle_result,content_result]]

#
connectDB(dbData)
```

연합 뉴스

- DB 연동 부분

```
def connectDB(dbData):
    DB_HOST = '127.0.0.1'
    DB_USER = 'root'
    DB_PASSWD = 'autoset'
    DB_NAME = 'python'

    conn = pymysql.connect(host=DB_HOST, user=DB_USER, password=DB_PASSWD,
                           db=DB_NAME, charset='utf8')

    curs = conn.cursor()

    sql = """insert into yonhapnews(newsTitle,newSubtitle_result,content)
            values (%s, %s, %s)"""
    curs.executemany(sql,dbData)

    conn.commit()

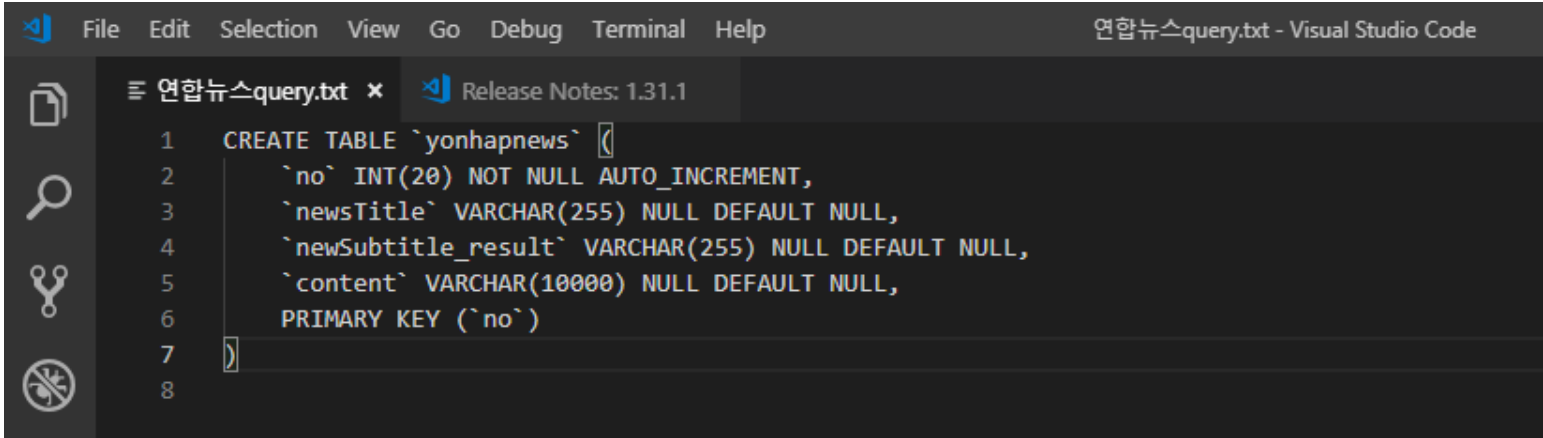
    conn.close()

search = input("검색 : ")
yonhapnewsUrl(search)

# In[ ]:
```

연합 뉴스

- 테이블 생성 쿼리



The screenshot shows the Visual Studio Code interface with a file named '연합뉴스query.txt' open. The editor displays a SQL query to create a table named 'yonhapnews'. The query is as follows:

```
1 CREATE TABLE `yonhapnews` (  
2   `no` INT(20) NOT NULL AUTO_INCREMENT,  
3   `newsTitle` VARCHAR(255) NULL DEFAULT NULL,  
4   `newSubtitle_result` VARCHAR(255) NULL DEFAULT NULL,  
5   `content` VARCHAR(10000) NULL DEFAULT NULL,  
6   PRIMARY KEY (`no`)  
7 )  
8
```

조선 일보

- 검색 페이지 단위로
기사 내용을 가져오는
코드

```
import pymysql
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import csv
import requests
from bs4 import BeautifulSoup

resultURLs = []
chosuns_result = []

def news_content(driver):
    pars = len(driver.find_elements_by_css_selector("#news_body_id .par"))[:]
    pars_test=[]
    for i in range(pars) :
        pars_test.append(driver.find_elements_by_css_selector("#news_body_id .par")[i].text)
    return ",".join(pars_test)

def chosunURL(driver):
    URL = driver.current_url
    html = requests.get(URL)
    bs = BeautifulSoup(html.text, "html.parser")
    chosun = bs.select(".result.news dl dt ")

    i = len(chosun)
    int(i)

    for name in chosun[0:i] :
        resultURLs.append(name.select_one("a").attrs.get("href"))

    return resultURLs

def chosunLinks(resultURLs,driver):
    for content in resultURLs:
        driver.get(content)
        a = driver.find_element_by_css_selector(".news_title_text #news_title_text_id").text
        b = driver.find_elements_by_css_selector("#news_body_id .news_subtitle")[0].text
        c = news_content(driver)
        d = [[a,b,c]]
        connectDB(d)
```

조선일보

- DB 연동 부분

```
def connectDB(d):
    DB_HOST = '127.0.0.1'
    DB_USER = 'root'
    DB_PASSWD = 'autoset'
    DB_NAME = 'python'

    conn = pymysql.connect(host=DB_HOST, user=DB_USER, password=DB_PASSWD,
                           db=DB_NAME, charset='utf8')

    curs = conn.cursor()

    sql = """insert into pythonTest(title,subject,content)
            values (%s, %s, %s)"""
    curs.executemany(sql,d)
    conn.commit()

    conn.close()

def chosun(search):
    URL = "http://search.chosun.com/search/total.search?query={search}&pageconf=total".format(search=search)
    driver = webdriver.Chrome("./chromedriver")
    driver.get(URL)

    chosun_search = driver.find_elements_by_css_selector(".main_menu li")
    chosun_search[1].click()

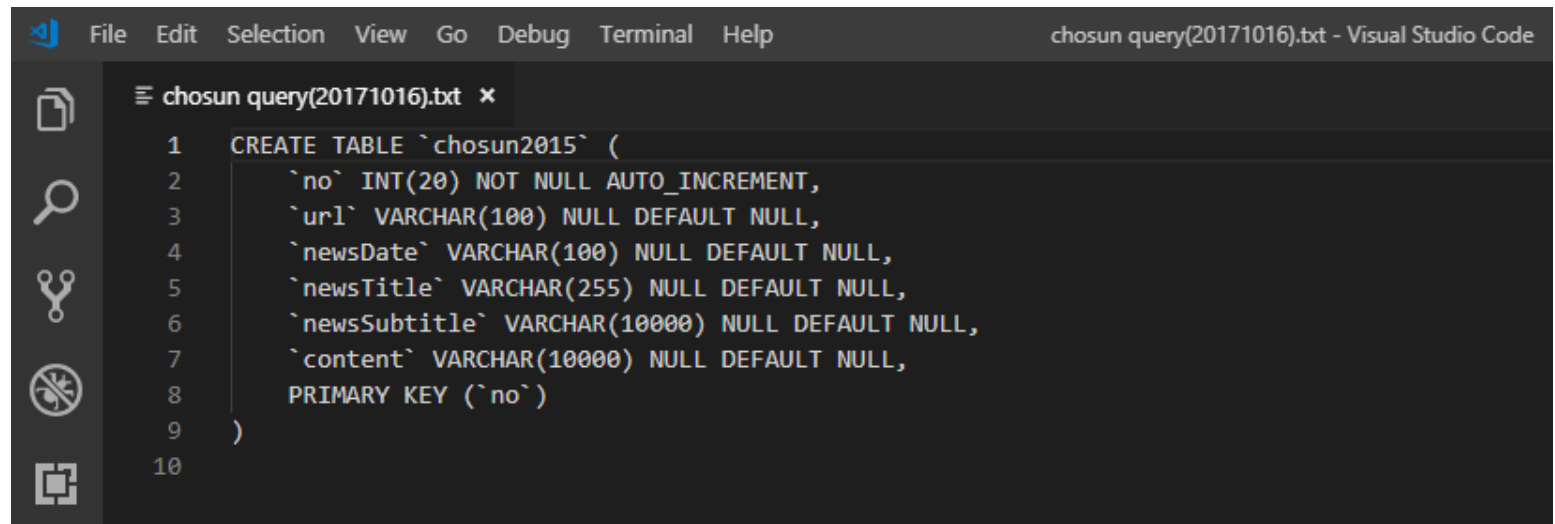
    chosun_search2 = driver.find_elements_by_css_selector("#opt_source dd a")
    chosun_search2[0].click()

    resultURLs = chosunURL(driver)
    chosunLinks(resultURLs,driver)

search = input("검색 : ")
chosun(search)
```


조선 일보

- 테이블 생성 쿼리

A screenshot of the Visual Studio Code editor interface. The title bar at the top reads "chosun query(20171016).txt - Visual Studio Code". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Debug", "Terminal", and "Help". The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The main editor area shows a file named "chosun query(20171016).txt" with the following SQL code:

```
1 CREATE TABLE `chosun2015` (  
2   `no` INT(20) NOT NULL AUTO_INCREMENT,  
3   `url` VARCHAR(100) NULL DEFAULT NULL,  
4   `newsDate` VARCHAR(100) NULL DEFAULT NULL,  
5   `newsTitle` VARCHAR(255) NULL DEFAULT NULL,  
6   `newsSubtitle` VARCHAR(10000) NULL DEFAULT NULL,  
7   `content` VARCHAR(10000) NULL DEFAULT NULL,  
8   PRIMARY KEY (`no`)  
9 )  
10
```