SSL

2022년 1학기

SEMINAR

서준혁

CONTENTS

01 Crawler란?

02 구현하기

03 기타 사항

01

Crawler란?

01. Crawler란 무엇인가?
02. Crawler의 종류
03. 필요 기술

01 Crawler란?

01. Crawler란 무엇인가?

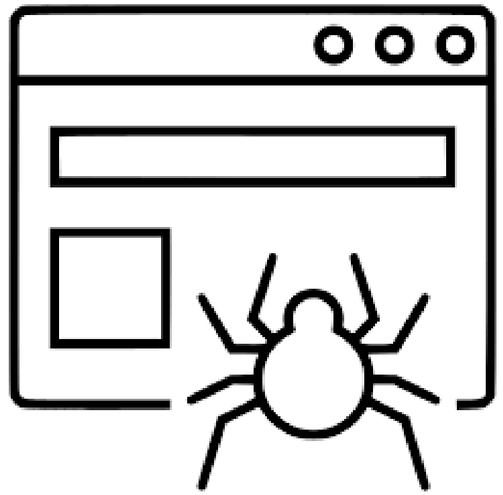
02. Crawler의 종류

03. 필요 기술

Crawler

Crawler란 무엇인가?

Crawler (이하 크롤러)란, 직역하면 '기어다니는 것' 이라는 뜻으로
일반적으로 웹페이지를 자동으로 반복하여 두루 방문하며 목표한 정보를
자동적으로 수집해오는 프로그램을 의미한다.



인공지능의 학습 데이터 표본이 되는 Big Data 구성,
특정 정보에 대한 통계를 내기 위한 데이터 수집 등
Data Science 분야 뿐만 아니라 다양한 분야에서 사용된다.

크롤러를 이용해 데이터를 수집하는 과정을 Crawling (이하 크롤링)
이라고 부르며, 크롤링을 통해 수집한 데이터는 좋은 Big Data가 될 수 있지만
불법 또는 상업적으로 허가 없이 사용한다면 처벌 대상이 될 수 있다.

01 Crawler란?

01. Crawler란 무엇인가?

02. Crawler의 종류

03. 필요 기술

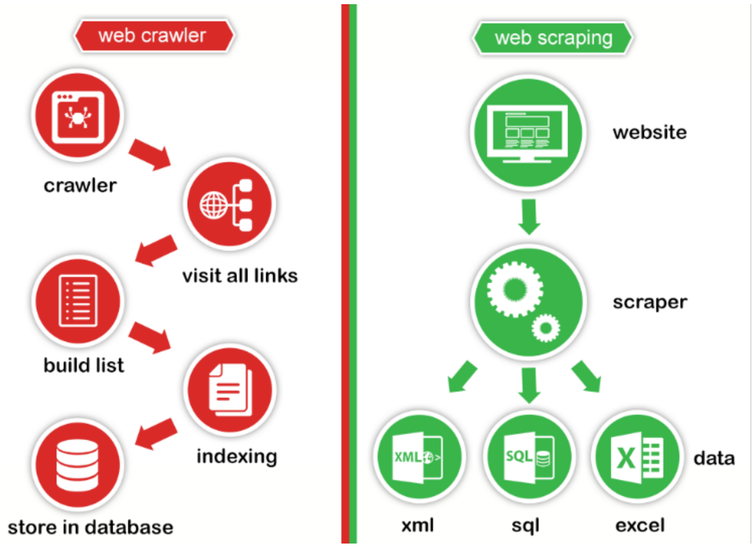
Crawler

Crawler의 종류

Crawler는 정적 크롤러와 동적 크롤러로 나뉜다.

정적 크롤러는 흔히 스크랩퍼 (Scraper)라고 불리며,
입력한 하나의 URL에서 특정 웹 문서 데이터를 추출한다.

동적 크롤러는 크롤러가 일정 패턴에 따라 웹페이지를 스스로 브라우징하며
사이트 또는 네트워크가 제공하는 정보를 끝없이 탐색한다.
입력한 URL을 Base URL (기준 URL)로 잡고 해당 페이지에 존재하는
특정 콘텐츠를 추출하고 해당 페이지의 모든 콘텐츠가 추출되면
페이지 내의 다른 링크에 뿌리를 내리듯이 스스로 접속하며 반복한다.



01 Crawler란?

01. Crawler란 무엇인가?

02. Crawler의 종류

03. 필요 기술

Crawler

필요 기술

스크랩퍼의 경우 원하는 하나의 HTML을 파싱하고
CSS 선택자를 이용해 추출할 데이터를 뽑아내면되므로
스크랩핑할 하나의 페이지를 개발자 도구를 이용해 잘 분석해서
Axios (Parser) + Cheerio (Selector) 을 사용해 직접 파싱부터 선택자까지
정의해줄 수 있으며 최근에는 BeautifulSoup4 등의 모듈을 이용해
비교적 과거보다 간편하게 스크랩핑이 가능하다

동적 크롤러의 경우 스크랩퍼의 기능 뿐만 아니라 스스로 반복 탐색하는 기능이
필요하기 때문에 Selenium, Scrapy 등을 이용해 구현이 가능하다.

AXIOS

cheerio

Scrapy

Se

02

구현하기

01. 정적 크롤러 구현하기

02. 동적 크롤러 구현하기

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

정적 크롤러 구현하기

SBS 뉴스 기사 제목을 입력받고 검색하여 데이터 프레임을

구성한 뒤 엑셀 파일로 저장하는 정적 크롤러를 구현해보자.

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from datetime import datetime
import time

# 검색할 기사 내용 입력
_search = input("검색어 입력 : ")
url = "https://news.sbs.co.kr/news/search/main.do?query=" + _search

# 헤더 정의
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75 Safari/537.36'}
webpage = requests.get(url, headers=headers).content
html = BeautifulSoup(webpage, 'html.parser')

# url을 넣기 위한 탐색
urls = html.find_all('a', {'class': 'psil_link'})
# 제목을 넣기 위한 탐색
titles = html.find_all('strong', {'class': 'psil_tit'})

URL = []
TITLE = []
CONTENT = []
DATE = []
NAME = []
```

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

```
21  URL = []
22  TITLE = []
23  CONTENT = []
24  DATE = []
25  NAME = []
26
27  for url in urls:
28      URL.append(url.attrs['href']) # List에 URL 담기
29
30  for title in titles:
31      TITLE.append(title.get_text()) # List에 제목 담기
32
33  # 기사 내용 리스트에 담기
34  for url in URL:
35      webpage2 = requests.get(url, headers=headers).content
36      html2 = BeautifulSoup(webpage2, 'html.parser')
37
38      # 기사 날짜
39      date_source = html2.find('span', {'class': 'date'}).get_text()
40      year = date_source[4:8]
41      month = date_source[9:11]
42      day = date_source[12:14]
43      hh = date_source[15:17]
44      mm = date_source[18:20]
45      date = year + "년" + month + "월" + day + "일" + hh + "시" + mm + "분 발행"
```

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

```
47      # 기사 본문
48      news_content = html2.find('div', {'class': 'text_area'}).get_text()
49      news_content = news_content.replace('\n', ' ')
50
51      # 작성자
52      author = html2.find('p', {'itemprop': 'author creator'}).get_text()
53      name = author[0:7] + author[22:41]
54
55      DATE.append(date)
56      CONTENT.append(news_content)
57      NAME.append(name)
58
59  # DataFrame 정의
60  df = pd.DataFrame({'제목': TITLE, '발행일': DATE, '내용': CONTENT, '작성자': NAME, 'URL': URL})
61  df.to_excel('%s.xlsx'%(_search), index=False, encoding='utf-8')
```

```
PS C:\Users\ssam2\Desktop\3학년 1학기\연구실\2회차> python .\staticCrawler.py
검색어 입력 : 금오공대
```

제목	발행일	내용	작성자	URL
'n번방 방지	2021년12월	민주당 (	한세현 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1006565533&plink=SEARCH&cooper=SBSNEWSSEARC
이재명, '험	2021년12월	<앵커>	한세현 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1006565432&plink=SEARCH&cooper=SBSNEWSSEARC
이재명 "문	2021년12월	<앵커>	김기태 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1006565231&plink=SEARCH&cooper=SBSNEWSSEARC
"2010년부	2020년07월	<앵커>	송인호 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1005881358&plink=SEARCH&cooper=SBSNEWSSEARC
삼성전자 -	2018년07월	삼성전자	SBS 뉴스	https://news.sbs.co.kr/news/endPage.do?news_id=N1004863980&plink=SEARCH&cooper=SBSNEWSSEARC
전국 4년제	2017년08월	전국 4년	홍지영 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1004344040&plink=SEARCH&cooper=SBSNEWSSEARC
금오공대 (	2017년03월	지난달 2	한지연 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1004105646&plink=SEARCH&cooper=SBSNEWSSEARC
[주영진의	2017년03월	인터뷰를	SBS 뉴스	https://news.sbs.co.kr/news/endPage.do?news_id=N1004078558&plink=SEARCH&cooper=SBSNEWSSEARC
금오공대 '	2017년03월	신입생 :	홍순준 기자	https://news.sbs.co.kr/news/endPage.do?news_id=N1004078254&plink=SEARCH&cooper=SBSNEWSSEARC
소주 8천병	2017년03월	신입생 :	SBS 뉴스	https://news.sbs.co.kr/news/endPage.do?news_id=N1004075733&plink=SEARCH&cooper=SBSNEWSSEARC

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

동적 크롤러 구현하기

일본 논문 게재 사이트 keizaireport.com 사이트 내에 존재하는 pdf 파일의 다운로드 링크를 수집해보자.

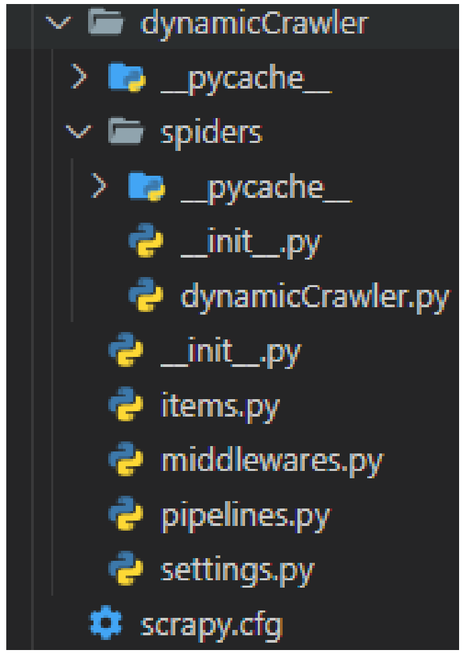
scrapy 모듈 프로젝트 디렉터리를 만들기 위해 'start project dynamicCrawler' 터미널에 입력

그럼 오른쪽 사진과 같은 프로젝트 폴더가 생성됨.

각 파일마다 크롤러를 구성하고 각각의 역할이 다르지만

모두 살펴보기에는 양이 너무 방대하기 때문에

직접적으로 크롤러 자체 코드를 작성하는 dynamicCrawler.py만 수정



02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

동적 크롤러 구현하기

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

동적 크롤러 구현하기

```python
import scrapy

class dynamicCrawler(scrapy.Spider):
    name = "dynamicCrawler"
    start_urls = [
        'http://www3.keizaireport.com/',
    ]

    def parse(self, response):
        self.browser.get(response.url)
        for href in response.css('a::attr(href)').extract():
            if href.endswith('.pdf'):
                f = open('pdf_links_.csv', 'a')
                f.write(href)
                f.write('\n')
                f.close()
            else:
                next_page = href
                if next_page is not None:
                    next_page = response.urljoin(next_page)
                    yield scrapy.Request(next_page, callback=self.parse)
```

scrapy crawl dynamicCrawler

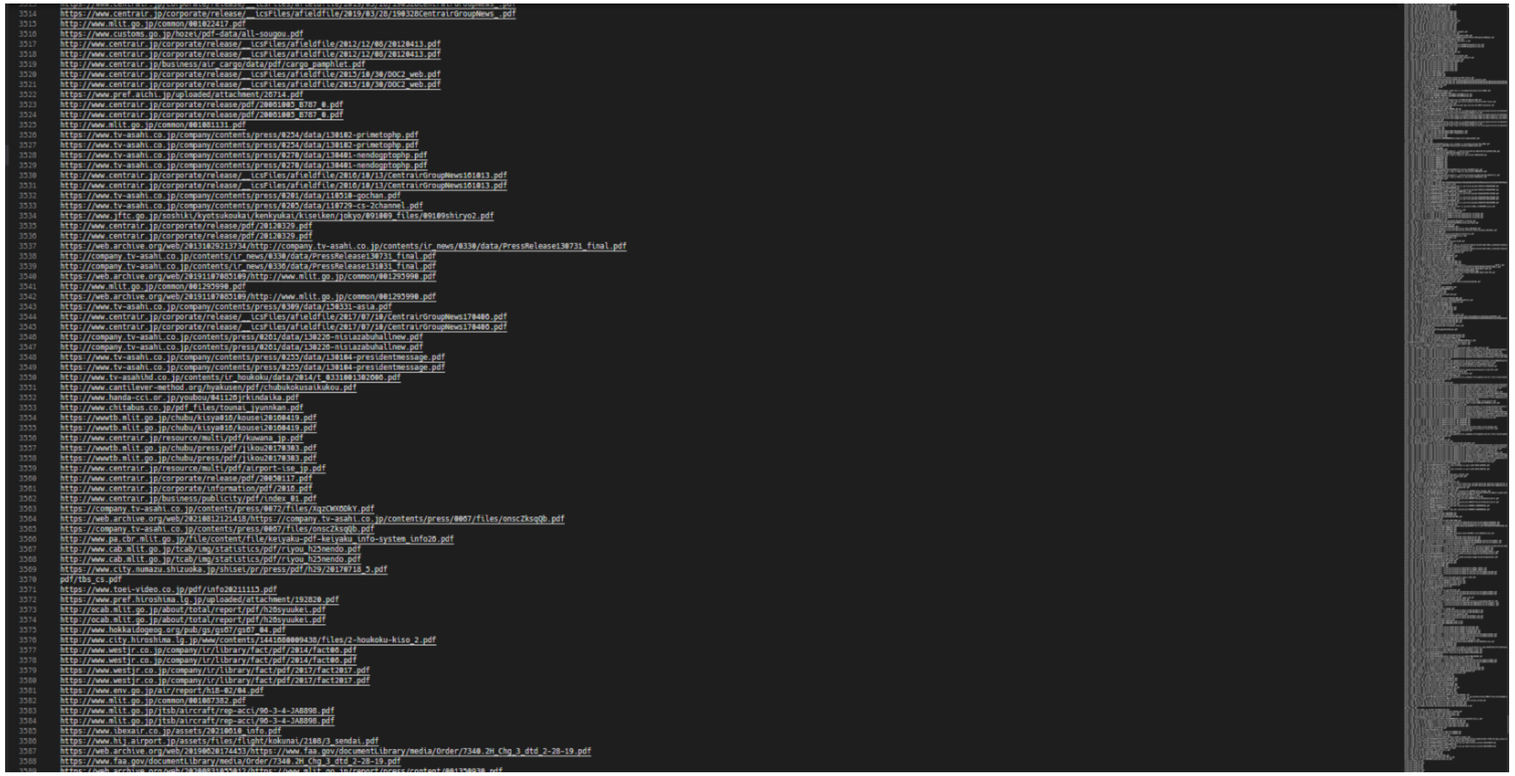
02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

동적 크롤러 구현하기



20분간 실행한 결과, 약 4천개의 PDF 파일의 다운로드 링크가 수집되었음.

02 구현하기

01. 정적 크롤러

02. 동적 크롤러

크롤러 구현

동적 크롤러 구현하기

이번 구현에서는 중단점을 지정해주지 않고 콜백 함수를 재귀적으로 호출해

계속해서 동작하도록 작성했더니 뒤로 가면 갈수록 논문 사이트와는 관계 없는 사이트들의

pdf 링크가 수집되는 것을 확인할 수 있었다.

실제로 동적 크롤러를 올바르게 사용하려면 적당한 기준점과 중단 지점을 설정해주어야

본래 목표한 수집 데이터의 연관성이 올라갈 수 있다.

03

기타 사항

01. 느낀 점

03 기타 사항

01. 느낀 점

Crawler

느낀 점

예전부터 크롤링과 빅데이터에 관심이 많아 천천히 공부했던 분야였는데,

이번 프로젝트를 위해 다시 크롤러를 꺼내들면서 복습할 수 있었던 계기가 된 것 같다.

이번에 인공지능 강의를 수강하면서 데이터 표본 분류를 통한 학습을 배우고 있는데

이를 크롤링과 접목시켜 수집한 데이터를 수치화하고 분류하여

목표한 데이터로부터의 정확성을 표현할 수 있는지 알아보고 싶다는 생각이 들었다.

감사합니다

SSL_서준혁