

Week 09 · 소셜네트워크 데이터마이닝과 분석

Social Data Mining 01

Joonhwan Lee

human-computer interaction + design lab.

오늘 다룰 내용

- Crawling from websites

1. Crawling from Websites

웹 데이터 수집

- ◆ RQ: 어떤 사람의 트위터 팔로워 구성을 통해 그 사람의 성향을 유추할 수 있을까?
 - ◆ 예1: A라는 사람의 트위터 팔로워는 모두 500명, 그 중에 30% 정치인, 60%는 연예인 → 연예 정보에 관심이 많은 사람.
 - ◆ 예2: A라는 사람이 팔로우하는 정치인 중, 보수성향 정치인 10%, 진보성향 정치인 90% → 진보적인 성향을 가진 사람.
 - ◆ Q1: 팔로우하는 사람의 속성 (연예인인지, 정치인인지, 보수성향의 정치인인지 등)은 어떻게 수집하나..?

웹 데이터 수집

http://twtkr.com/fpl.php?d=3_1&n=

The screenshot shows the twtkr website interface. At the top, there is a navigation bar with the twtkr logo and a search bar. Below the navigation bar, there is a sidebar on the left with a list of categories. The main content area displays a list of users, each with a profile picture, name, and follower count. Red arrows point to the search bar, the user profile of '유시민' (Yoo Si-min), the user profile of '정봉주' (Jeong Bong-ju), and the user profile of '김용민' (Kim Yong-min).

twtkr 디렉토리

twtkr 디렉토리

홈 | 검색 | 디렉토리 | 이벤트 | 모임 | 광고 | 동네 | 도구 | 설정 | 도움말 | 로그인

twtkr 디렉토리

순위 디렉토리

- 경제
- 전자(문예인 제외)
- 연예인(아이돌)
- 연예인
- 스포츠
- 정치인/공직자
 - 정치인
 - 국무총리/장·차관
 - 광역단체장
 - 기초자치단체장
 - 지방자치단체의원
 - 교육감
 - 공공기관장
 - 기타
- 기업인/CEO
- 전문가
- 미디어
- 기업
- 기관/단체
- 정치단체
- 학교/대학교
- 작가/출판사
- 엔터테인먼트
- 생활/문화서비스
- 개인전문서비스
- 종교/종교인
- 팬클럽/후援회
 - 트위터 서비스
 - 인기/유행 트위터
 - 활동 없는 트위터

powered by cilleh

포인트 팔로워 리스트됨

1 최시원

twtkr 디렉토리

사람찾기

정치인/공직자 : 정치인 순위 (1,076) > 포인트

포인트 팔로워 - 팔로워 팔로워 리스트됨 팔로워 트윗 20명씩 보기

twtkr 경북 관광 알람이 @GB_tour 팔로워

twtkr 마케팅 @twkr_mkt 팔로워

유시민 @u_smin 533,403 #1

533,403 | 서진보정당추진회의운영위원 국가란 무엇인가 저자

팔로워 : 518,664 | 팔로잉 : 40,666 | 트윗 : 1,062 | 리스트됨 : 17,381

정봉주 @BBK_Super 390,192 #2

390,192 | 신나는 진보의 가치를 꿈꾸는 국민이 주인공인 미래광역들

팔로워 : 400,938 | 팔로잉 : 1,962 | 트윗 : 1,253 | 리스트됨 : 9,514

김용민 @funny 378,190 #3

378,190 | 영국1서... 서식. 나는 음식에서 일하고 양식을 지칭한다. 여기서 양지는... 부위.

팔로워 : 364,652 | 팔로잉 : 20,471 | 트윗 : 9,172 | 리스트됨 : 10,3...

문성근 (민주당,배우) @moonmkn 242,760 #4

242,760 | 2010.8 on+off결합 네티의장당으로 다경대통합 국민의영연 차인자. 2012.1 민주당합당 건설, 최고위원 당선 2012.4총선 부산 북.광서(을) 당군 낙선! 우회하...그래도 마침내 2012.12 정권교체! 주소창-moonparty.kr

twtkr스폰서 이벤트

줄리엣성형외과 분당점 @juletic

줄리엣성형네트워킹분당(서현)점입니다~

+ 팔로워 바오기

@twtkr_dir

+ 팔로워 문의하기

디렉토리 등록/추천

디렉토리 등록 사용자 8,992 명

유리신용 7분 > 디렉토리 등록정책 >

twtkr 프리미엄 서비스

웹 데이터 수집

◆ 실습 1: 소스코드 분석

- ◆ 수집하려는 웹 페이지의 소스를 분석하여, 필요한 데이터가 담긴 반복되는 패턴블럭을 찾아낸다.
- ◆ 반복되는 패턴블럭의 계층 구조를 찾아내 각각의 요소를 정리한다.
- ◆ 계층 구조 내에서 필요한 요소를 따로 찾아 정리한다.
- ◆ `twtkr_example.html`을 열고 주요한 데이터의 반복되는 패턴블럭을 찾고, 내부 데이터를 구조화 하시오.

웹 데이터 수집

◆ 실습 1

```
<div class="total_ranking">
```

```
<div class="stream">
```

```
<div class="avatar">
```

```
<div class="article">
```

```
<div class="header">
```

```
<cite>
```

```
...
```

```
<div class="stream">
```

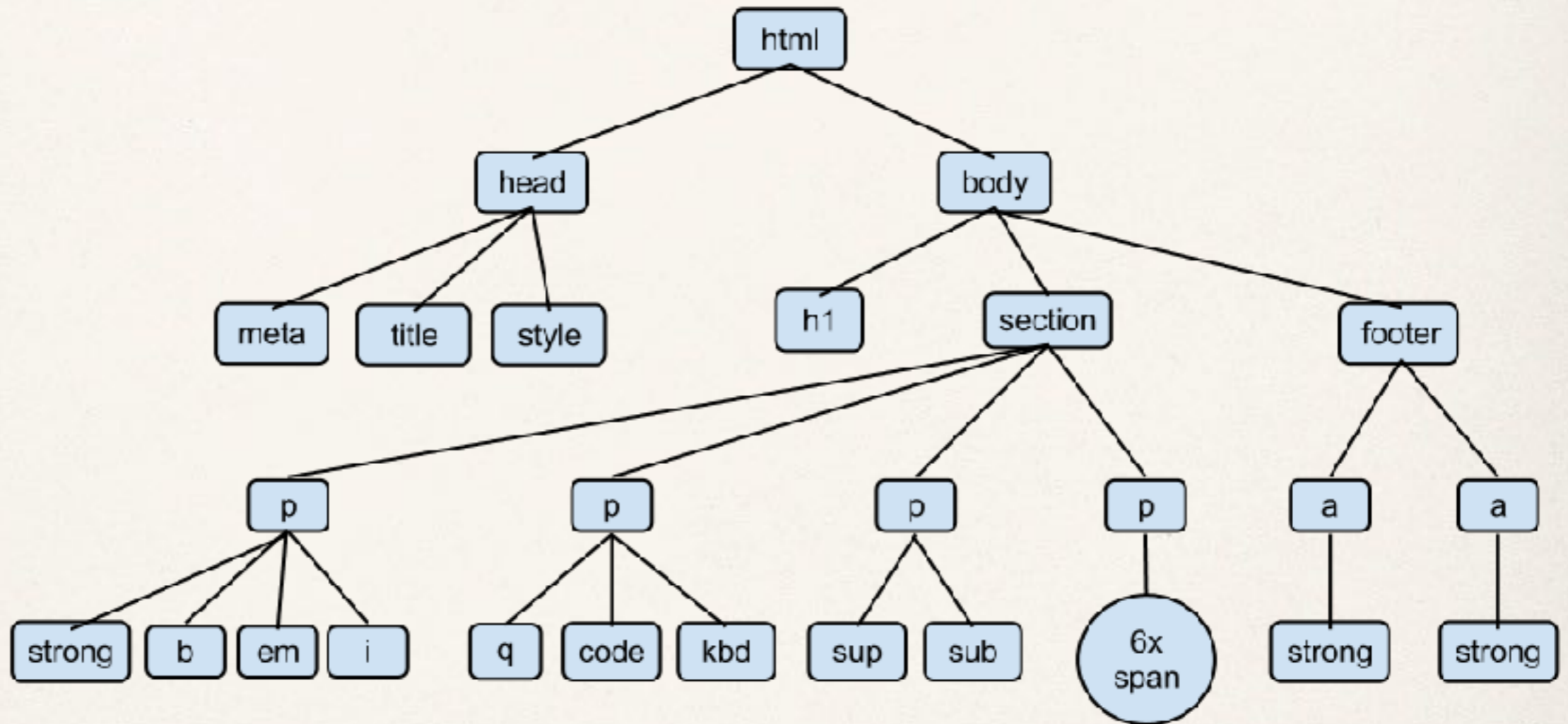
```
<div class="stream">
```

```
...
```

BeautifulSoup을 이용한 웹페이지 수집 및 분석

- ◆ 웹 문서로부터 특정한 데이터를 추출하기 위해서는 HTML 문서를 읽고 구조를 해석할 수 있는 소프트웨어가 필요.
- ◆ BeautifulSoup은 HTML, XML 등을 읽고 해석할 수 있는 소프트웨어 (parser)
 - ◆ BS4는 문서를 파싱한 후 DOM Tree 를 만든다.
- ◆ BeautifulSoup 설치
 - ◆ `pip install beautifulsoup4`

HTML Document 와 DOM Tree



HTML Document 와 DOM Tree

The Document

```
<html>
<body>
<h1>Title</h1>
<p>A <em>word</em></p>
</body>
</html>
```

The DOM Tree

```
DOCUMENT
├── ELEMENT: html
│   ├── TEXT: '\n'
│   ├── ELEMENT: body
│   │   ├── TEXT: '\n'
│   │   ├── ELEMENT: h1
│   │   │   └── TEXT: 'Title'
│   │   ├── TEXT: '\n'
│   │   ├── ELEMENT: p
│   │   │   ├── TEXT: 'A'
│   │   │   └── ELEMENT: em
│   │   │       └── TEXT: word
│   └── TEXT: '\n'
└── TEXT: '\n'
```

BS4를 이용한 HTML Parsing

- ◆ BeautifulSoup의 사용

```
> from bs4 import BeautifulSoup
> html_doc = "<html><body><h1>Mr. Belvedere Fan
Club</h1></body></html>"

> soup = BeautifulSoup(html_doc, "html.parser")
> soup
=> <html><body><h1>Mr. Belvedere Fan Club</h1></
body></html>

> print(soup.prettify())

> heading = soup.find_all("h1")
=> [<h1>Mr. Belvedere Fan Club</h1>]

> heading[0].get_text()
=> 'Mr. Belvedere Fan Club'
```

BS4를 이용한 HTML Parsing

◆ find_all 의 사용법

- ◆ `find_all("h1")`

- ◆ `<h1>~</h1>` 태그 안의 내용

- ◆ `find_all("div")`

- ◆ `<div>~</div>` 태그 안의 내용

- ◆ `find_all("div", class_="footer")`

- ◆ `<div class="footer">~</div>` 태그 안의 내용

- ◆ `find_all("div", id="footer")`

- ◆ `<div id="nav">~</div>` 태그 안의 내용

- ◆ `divs = soup.find_all("div", class_="header")`

- `for div in divs:`

- `if div.a["href"] == "twitter_anywhere":`

- ◆ `<div class="header">~</div>` 태그 안의 내용

BS4를 이용한 HTML Parsing

◆ find_all의 사용법

- ◆ find_all이 반환하는 값은 array (한 페이지에 같은 요소가 여럿 있을 것을 가정하므로...)
- ◆ 따라서 find_all이 수집한 데이터를 처리하기 위해서는 for-loop 등의 iterator 를 사용한다.

```
id_list = []
divs = soup.find_all("div", class_="header")
for div in divs:
    if div.a["href"] == "twitter_anywhere":
        id_list.append(div.a.text)
```

실습 2: twitter 아이디와 사용자 이름 수집

- ✦ twtkr_example.html 파일을 읽어 트위터 아이디와 사용자 이름을 수집해 보자. 수집된 id 에서 @ 기호를 삭제하여 출력한다.

- ✦ 예: u_simin, 유시민

- ✦ (참고) HTML 파일 불러오는 방법

```
with open("data/twtkr_example.html") as  
file:
```

```
    html_doc = file.read()
```

웹에서 직접 데이터 수집

- ◆ 항상 저장된 페이지에서 파일을 수집할 수 없음.
- ◆ 실시간으로 웹페이지에 접속해서 저장된 페이지를 수집해야 함.
- ◆ 인터넷에 접속하여 페이지의 소스코드를 받아 처리하기 위해서는 다음과 같은 명령어를 사용.

```
◆ import urllib.request
  with urllib.request.urlopen("http://
    twtkr.com/fpl.php?d=3&n=20") as url:
      doc = url.read()
```

Page Iterator

- ◆ 앞선 예제의 페이지 버튼을 눌러 다음 페이지로 이동.



- ◆ page1: http://twtkr.com/fpl.php?d=3_1&n=
- ◆ page2: http://twtkr.com/fpl.php?d=3_1&s=&p=2&n=20
- ◆ page3: http://twtkr.com/fpl.php?d=3_1&s=&p=3&n=20
- ◆ page4: http://twtkr.com/fpl.php?d=3_1&s=&p=4&n=20

- ◆ 공통점?

Page Iterator

- ◆ http://twtkr.com/fpl.php?d=3_1&s=&p=3&n=20
 - ◆ <http://twtkr.com/fpl.php> 뒤에 붙어 있는 문자와 숫자는 패러미터.
 - ◆ p=3: page number
 - ◆ n=20: page당 표시할 데이터의 수
 - ◆ d=3_1: ??????
 - ◆ s=: ??????

순위 디렉토리

- 전체
- 전체(연예인 제외)
- 연예인(아이들)
- 연예인
- 스포츠
- 정치인/공직자
 - 정치인
 - 국무총리/장·차관
 - 광역단체장
 - 기초자치단체장
 - 지방자치단체의원
 - 교육감
 - 공공기관장
 - 기타
- 기업인/CEO
- 전문가
- 미디어

Page Iterator

- ✦ http://twtkr.com/fpl.php?d=1_3&n=20&p=3

The screenshot shows the twtkr website interface. The browser address bar displays the URL: twtkr.olleh.com/fpl.php?d=1_3&n=20&p=3. The page title is "twtkr 디렉토리". The main content area shows a search bar and a list of users. The left sidebar contains a "순위 디렉토리" (Ranking Directory) with various categories. The main content area shows a list of users with their profiles, including "박수홍" and "류담".

순위 디렉토리

- 전체
- 전체(연예인 제외)
- 연예인(아이들)
- 연예인
 - 연기/영화
 - 가수/음악
 - 코미디언/개그맨
 - 방송/예능
 - 성우
 - 모델
 - 연예인 기획/관리자
 - 미술사
 - 레디싱모델
 - 기타
- 스포츠
- 정치인/공직자
 - 기업인/CEO
- 전문가
- 미디어
- 기업
- 기관/단체
- 정치단체
- 학교/대학교
- 작가/출판사

연예인 : 코미디언/개그맨 순위 (81) > 포인드

팔로워 - 팔로잉 팔로워 리스트됨 팔로잉 트윗 20명씩 보기

박수홍 @psaongg 9,678 #41
MC 개그맨
팔로워 : 0,097 | 팔로잉 : 5 | 트윗 : 0 | 리스트됨 : 202

류담 @dam1102 8,858 #42
출연중...
팔로워 : 0,173 | 팔로잉 : 1,290 | 트윗 : 190 | 리스트됨 : 276

Page Iterator

- ◆ 전체 페이지를 자동으로 수집하려면..?
 - ◆ twtkr의 정치인 페이지의 경우, 1 페이지 부터 54 페이지까지 존재함.
 - ◆ 불필요한 패러미터를 삭제한 후 loop 를 사용하여 page 자동 전환

```
base_url = "http://twtkr.com/fpl.php?  
d=3_1&n=20&p="
```

```
for i in range(1, 55):  
    print(base_url + str(i))
```

실습 3: Page Iterator 사용하여 모든 아이디 수집

- ◆ twtkr의 디렉토리에서 정치인의 트위터 아이디와 이름을 모두 수집하는 프로그램을 작성하시오.
 - ◆ 주소: http://twtkr.com/fpl.php?d=3_1&s=&p=1&n=20
 - ◆ Step 1: url 정리 및 공통 요소 추출
 - ◆ Step 2: page iterator 작성
 - ◆ Step 3: 실습 2에서 작성한 코드 사용하여 트위터 아이디 및 이름 수집.
 - ◆ Step 4: 화면에 표시. (출력)

Page Iterator 수정

- ◆ 페이지가 끝나는 시점을 자동으로 알고 싶다.
 - ◆ 페이지를 수집할 때, 모든 페이지의 끝 번호를 알수 있을까?
 - ◆ twtkr에서도 카테고리마다 페이지 숫자가 모두 다름
 - ◆ 정치인: 54페이지, 연예인: 29페이지
 - ◆ 끝나는 페이지의 특징을 찾아보자.



화살표가 없다!

Page Iterator 수정

- ◆ 화살표를 그려주는 태그

- ◆ `<li class="btn btn_next">`

- ◆ 맨 마지막 페이지에는 해당 태그가 존재하지 않음. 이는 다음과 같이 표현 가능.

```
len(soup.find_all("li", class_="btn btn_next")) == 0
```

- ◆ 따라서 현재 불러온 페이지에 `<li class="btn btn_next">` 가 존재하지 않고, `<ul class="paging">` 아래 있는 페이지 숫자가 현재 URL에 사용한 페이지와 같으면 현재 페이지가 마지막 페이지.

실습 4: 페이지 자동 종료

- ✦

```
if len(sp.find_all("li", class_="btn btn_next")) == 0:
```
- ✦

```
uls = sp.find_all("ul", class_="paging")
for ul in uls:
    lis = ul.find_all('li')
    last_page_no = int(lis[len(lis)-1].text)
```
- ✦ 위의 코드를 이용하여 `has_no_more_pages()` 함수를 만든 후에,

```
while not_lastpage:
    ...
    if has_no_more_pages(soup, pageno):
        not_lastpage = False
```

를 이용하여 자동으로 페이지 이동하며 데이터 수집.

Questions?
