웹 크롤러 (Web Crawler)

전북대학교 컴퓨터공학부 정보검색 1분반(이경순 교수)

2019-09-30

컴퓨터공학부 201514740 이동준

Index

전북대학교 컴퓨터공학부 정보검색 1분반(이경순 교수) 웹 크롤러 조사

Web Crawler ?

정의와 발전과정

조직적, 자동화된 방법으로 **월드와이드 웹(WWW)**을 **탐색**하는 **컴퓨터 프로그램**이다. (위키피디아 웹 크롤러)

크롤링을 하는 이유

- **Content indexing for search engines (검색 엔진의 색인 컨텐츠)**
- **Automated testing and model checking of the web application**
  (웹 어플리케이션에서 자동화된 테스트와 모델 체크를 위함)
- **Automated security testing and vulnerability assessment**
  (자동화된 보안 테스트와 취약점 평가를 위함)

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014) p1

Web Crawler ?

정의와 발전과정

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014) p3

Traditional	Deep	RIA (Rich Internet Application)	Unified Model
Seed : URL list **Set of seed URLs** 단순 하이퍼링크를 통한 URL 접속	**Set of seed URLs , User context data** Form Tag 사용자에 따른 다른 결과를 얻음	**Starting page** 어플리케이션의 DOM 을 이용한 연결 (JavaScript event,HTML5,Ajax) **MS Sliverlight** **Oracle JavaFX**	**A seed URL** URL 변화없이 DOM에 근거함. 클라이언트 측면 이벤트 발생

How to Web Crawler operate ?

작동원리

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014) p8

Traditional

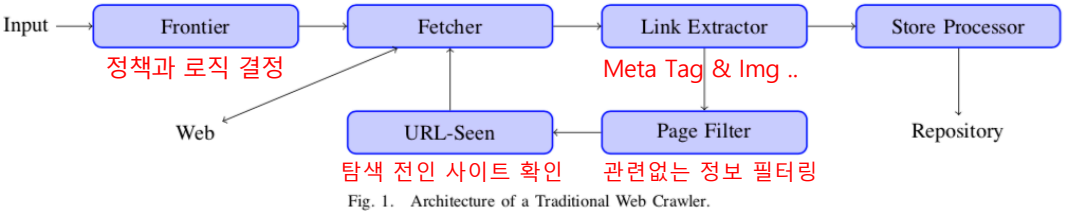


Fig. 1. Architecture of a Traditional Web Crawler.

Study	Component	Method	Goal
WebCrawler MOMspider [4]	Fetcher Frontier Page filter	Parallel downloading of 15 links robots.txt Black-list	Scalability Politeness
Google [12]	Store processor Frontier	Reduce disk access time by compression PageRank	Scalability Coverage Freshness
Mercator [5]	URL-Seen	Batch disk checks and cache	Scalability
WebFountain [13]	Storage processor Frontier Fetch	Local copy of the fetched pages Adaptive download rate Homogenous cluster as hardware	Completeness Freshness Scalability
Polybot [14]	URL-Seen	Red-Black tree to keep the URLs	Scalability
UbiCrawler [15]	URL-Seen	P2P architecture	Scalability
pSearch [16]	Store processor	Distributed Hashing Tables (DHT)	Scalability
Exposto et al. [19]	Frontier	Distributed Hashing Tables (DHT)	Scalability
IRLbotpages [20]	URL-Seen	Access time reduction by disk segmentation	Scalability

TABLE II
TAXONOMY OF TRADITIONAL WEB CRAWLERS

How to Web Crawler operate ?

작동원리

Deep Web

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014) p8-9

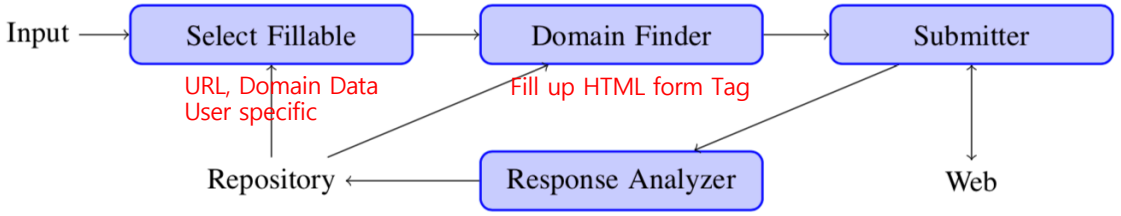


Fig. 2. Architecture of a Deep Web Crawler.

Study	Component	Method	Goal
HiWe [7]	Select fillable Domain Finder Submitter Response Analyst	Partial page layout and visual adjacency Normalization by stemming etc Approximation matching Manual domain Ignore submitting small or incomplete forms Hash of visually important parts of the page to detect errors	Lenient submission efficiency Submission efficiency
Liddle et al [24]	Select fillable Domain Finder	Fields with finite set of values, ignores automatic filling of text field Stratified Sampling Method (avoid queries biased toward certain fields) Detection of new forms inside result page, Removal of repeated form Concatenation of pages connected through navigational elements Stop queries by observing pages with repetitive partial results Detect record boundaries and computes hash values for each sentence	Lenient submission efficiency Submission efficiency
Barbosa and Freire [23]	Select fillable Domain Finder Response Analysis	Single keyword-based queries Based on collection data associate weights to keywords and uses greedy algorithms to retrieve as much contents with minimum number of queries. Considers adding stop-words to maximize coverage Issue queries using dummy words to detect error pages	Lenient submission efficiency Submission efficiency
Ntoulas et al [25]	Select fillable Domain Finder	Single-term keyword-based queries Three policies: random, based on the frequency of keyword in a corpus, and an Adaptive policy that learn from the downloaded pages. maximizing the unique returns of each query	Lenient submission efficiency Submission efficiency
Lu et al [26]	Select fillable Domain Finder	querying textual data sources, Works on sample that represents the original data source. Maximizing the coverage per number of requests to the problem of set-covering problem	Lenient submission efficiency Scalability Submission efficiency

TABLE III
TAXONOMY OF DEEP WEB CRAWLERS

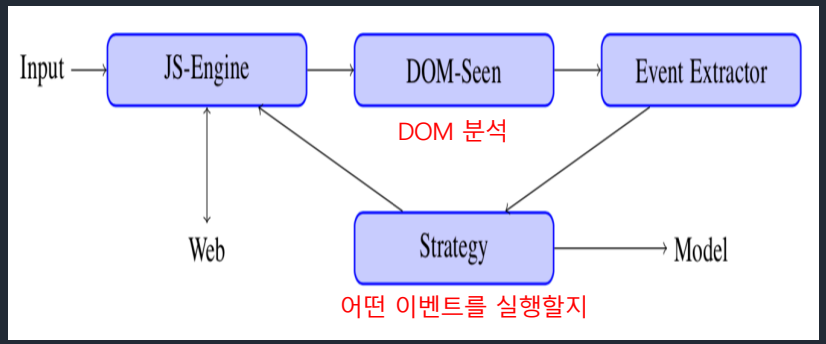
How to Web Crawler operate ?

작동원리

RIA

Start Virtual browser
& run javascript engine

JavaScript Event 추출



DOM : 문서 객체 모델
특징 : HTML, SVG, 또는 XML 객체를 문서로 모델링하는 것

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014) p8-9

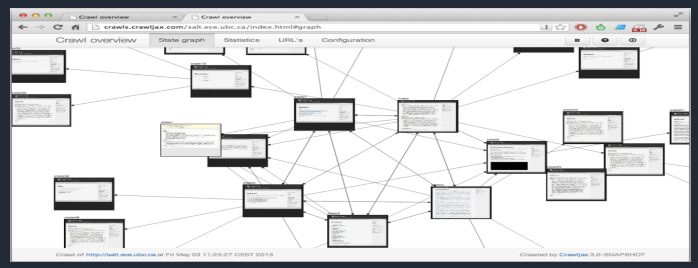
Study	Component	Method	Goal
Duda et al [29]–[31]	Strategy JS-Engine DOM-Seen	Breadth-First-Search Caching the JavaScript function calls and results Comparing Hash value of the full serialized DOM	Completeness Efficiency
Mesbah et al [32], [33]	Strategy DOM-Seen	Depth-First-Search Explores an event only once New threads are initiated for unexplored events Comparing Edit distance with all previous states	Completeness State Coverage Efficiency Scalability
CrawlRIA [34]–[37]	Strategy DOM-Seen	Depth-First strategy (Automatically generated using execution traces) Comparing the set of elements, event types, event handlers in two DOMs	Completeness
Kamara et al [8], [38]	Strategy	Assuming hypercube model for the application. Using Minimum Chain Decomposition and Minimum Transition Coverage	State Coverage Efficiency
M-Crawler [50]	Strategy	Menu strategy which categorizes events after first two runs Events which always lead to the same/current state has less priority Using Rural-Postman solver to explore unexecuted events efficiently	State Coverage Efficiency Completeness
Peng et al. [41]	Strategy	Choose an event from current state then from the closest state	State Coverage Efficiency
AjaxRank [31]	Strategy DOM-Seen	The initial state of the URL is given more importance Similar to PageRank, connectivity-based but instead of hyperlinks the transitions are considered hash value of the content and structure of the DOM	State Coverage Efficiency
Dincturk et al. [51]	Strategy	Considers probability of discovering new 'state' by an event and cost of following the path to events state	State Coverage Efficiency
Dist-RIA Crawler [44]	Strategy	Uses JavaScript events to partition the search space and run the crawl in parallel on multiple nodes	Scalability
Feedex [42]	Strategy	Prioritize events based on their possible impact of the DOM. Considers factors like code coverage, navigational and page structural diversity	State Coverage Efficiency

TABLE IV
TAXONOMY OF RIA WEB CRAWLERS

How to Web Crawler operate ?

전략

- **DOM Equivalence and Comparison** — DOM의 Hash 값을 통해서 비교를 함
- **Parallel Crawling** — 반복되는 페이지 다운로드를 피하고 오버헤드를 최소화 하면서 다운로드율을 높이기 위함
- **Automated Testing**



crawljax

- **Ranking** — 기존의 PageRank와는 달리 AjaxRank 등 하이퍼링크 중심을 벗어나려 함

Let's Practice Web Crawler !

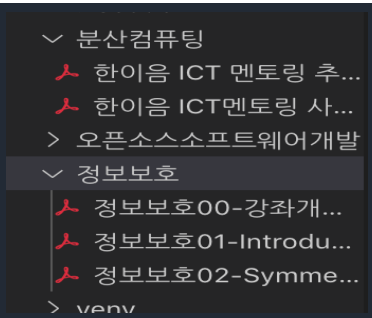
Python을 이용한 실습

Request, BeautifulSoup 사용한 학교 ieilms 정보 가져오기

```
# HTML 소스 가져오기
html = req.text
# HTTP Header 가져오기
header = req.headers
# HTTP Status 가져오기 (200: 정상)
status = req.status_code
# HTTP가 정상적으로 되었는지 (True/False)
is_ok = req.ok

LOGIN_INFO = {
    'username': [redacted]
    'password':
}

login_req = s.post('https://ieilms.jbnu.ac.kr/login/index.php', data=LOGIN_INFO)
    # 어떤 결과가 나올까요?
print(login_req.text)
```




```
URL : http://ieilms.jbnu.ac.kr/course/view.php?id=11824
교수 : 이형태
학기 : (2학기)
과목명 : 정보보호
과목코드 :[0000116953_1]
공지사항 목록 :
                번호 : 4
                제목 : [강의슬라이드] 02. Symmetric Encryption
                첨부파일 :
                        정보보호02-Symmetric Encryption.pdf
                작성자 : 이형태
                일자 : 2019-09-11


                번호 : 3
                제목 : [강의슬라이드-수정] 01. Introduction to Information Security and Cryptography-
                첨부파일 :
                        정보보호01-IntroductiontoInfoSecCrypto_수정.pdf
                작성자 : 이형태
                일자 : 2019-09-11


                번호 : 2
                제목 : [강의슬라이드] 00. Course Overview
                첨부파일 :
                        정보보호00-강좌개요.pdf
                작성자 : 이형태
                일자 : 2019-09-11


                번호 : 1
                제목 : 대학원 진학 설명회 - 9월 5일 수업 대체
                작성자 : 이형태
                일자 : 2019-09-03




URL : http://ieilms.jbnu.ac.kr/course/view.php?id=12412
교수 : 박현찬
학기 : (2학기)
과목명 : 분산컴퓨팅
과목코드 :[0000121757_1]
공지사항 목록 :
                번호 : 1
                제목 : 참고 : ICT멘토링 사업 추경 과정 안내
                첨부파일 :
                        한이음 ICT멘토링 사업소개_추경.pdf
                        한이음 ICT 멘토링 추경 사업 참여 안내(요약).pdf
                작성자 : 박현찬
                일자 : 2019-09-10
```

Reference

Web Crawler

Seyed M. Mirtaheri 외 4명. "A Brief History of Web Crawlers". (2014)

Parallel Crawling

DIVAKAR YADAV . "Parallel Crawler Architecture and Web Page Change Detection".(2008).ISSN: 1109-2750

기타 웹 크롤러에 대한 검색

https://ko.wikipedia.org/wiki/리치_인터넷_어플리케이션

https://ko.wikipedia.org/wiki/웹_크롤러

http://crawljax.com/

https://frontera.readthedocs.io/en/v0.2.0/topics/what-is-a-crawl-frontier.html

https://developer.mozilla.org/ko/docs/Web/API/Document_Object_Model