9장. 웹 로봇

웹로봇

- 사람과의 상호작용 없이 연속된 웹 트랜잭션들을 자동으로 수행하는 소프트웨어 프로그램
- 예시
  - 주식시장 서버에 매 분 GET 요청 보내고 얻은 데이터를 활용해 주가 추이 그래프 생성하는 봇
  - www 규모와 진화에 대한 통계 정보 수집 및 기록 하는 봇
  - 검색 db 를 만들기 위한 문서 수집 검색엔진 봇
  - 상품 가격 db 만들기 위한 온라인 쇼핑몰의 카탈로그에서 웹페이지 수집

목차

9.1 크롤러와 크롤링

크롤러

- 웹페이지를 재귀적으로 반복하는 방식으로 순회하는 로봇
- 혹은 스파이더라 부르는데 스파이더라 부르는 이유는 웹(web) 을 따라 기어다니기(crawl) 때문 ㅋㅋ

9.1.1 어디에서 시작하는가: 루트집합

- 루트 집합 : 크롤러가 방문을 시작하는 URL 의 초기 집합
- 일반적으로 좋은 루트 집합은 크고 인기 있는 웹사이트, 새로 생성된 페이지들의 목록, 자주 링크되지 않는 잘 알려지지 않은 페이지들의 목록

9.1.2 링크 추출과 상대 링크 정상화

- 크롤러는 웹을 돌다다니며 각 페이지 안에 들어있는 URL 링크들을 파싱, 크롤링할 페이지들의 목록에 추가해야 한다.

9.1.3 순환 피하기

로봇이 웹 크롤링 시, 루프나 순환에 빠지지 않도록 매우 조심해야 한다.

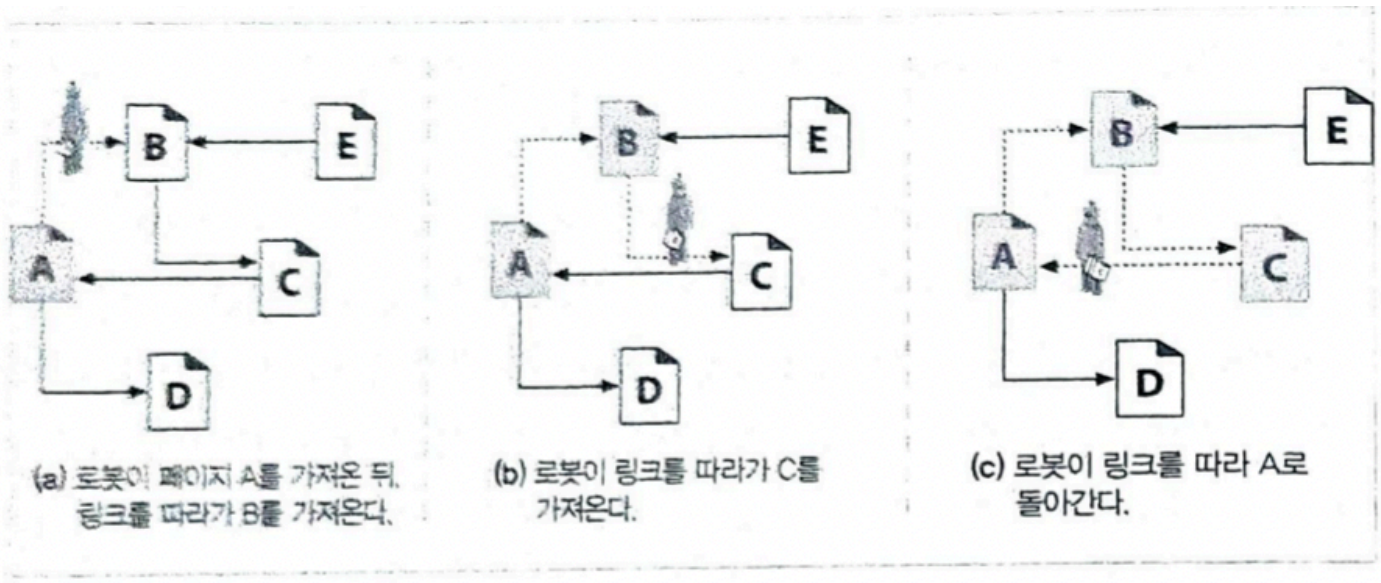

그림 9-2 하이퍼링크의 웹을 크롤링하기

9.1.4 루프와 중복

순환은 최소 다음 세가지 이유로 인해 크롤러에게 해롭다.

- 크롤러를 루프에 빠뜨려 꼼짝 못하게 만들 수 있다.
- 크롤러가 같은 페이지를 반복해서 가져오면 웹 서버의 부담이 된다. 또한 크롤러로 인해 서비스 방해 행위는 법적인 문제제기의 근거가 될 수 있다.
- 크롤러는 많은 수의 중복페이지들을 가져오게되는데 결국 크롤러 애플리케이션은 중복 컨텐츠로 넘쳐나게 될 것이다.

9.1.5 빵 부스러기의 흔적

대규모 웹 크롤러가 그들이 방문한 곳을 관리하기 위해 사용하는 유용한 기법은 아래와 같다.

트리와 해시 테이블

- 방문한 URL 추적을 위해 검색 트리나 해시 테이블을 사용했을 수도 있다.

느슨한 존재 비트맵

- 공간 사용을 최소화하기 위해 몇몇 대규모 크롤러들은 존재 비트 배열과 같은 느슨한 자료 구조를 사용한다. 각 URL은 해시 함수에 의해 고정된 크기의 숫자로 변환되고 배열 안에 대응하는 '존재 비트(presence bit)'를 갖는다.
- URL이 크롤링 되었을 때 해당하는 존재 비트가 만들어지는데 만약 존재 비트가 이미 존재하면 크롤러는 그 URL 을 이미 크롤링 되었다고 간주한다.

체크포인트

- 로봇 프로그램이 갑작스레 중단될 경우를 대비해, 방문한 URL의 목록이 디스크에 저장되었는지 확인 한다.

파티셔닝

- 한 대의 컴퓨터로 크롤링을 완수하기 어려워지면서 로봇의 역할을 나눠 파티셔닝으로 나눈다.

9.1.6 별칭과 로봇 순환

올바른 자료 구조를 갖추었더라도 URL 이 별칭을 가질 수 있는 이상 어떤 페이지를 이전에 방문했었는지 말해주는게 쉽지 않을 때도 있다.

	첫 번째 URL	두 번째 URL	어떤 경우에 같은 URL을 가리키게 되는가
a	http://www.foo.com/bar.html	http://www.foo.com:80/bar.html	기본 포트가 80번일 때
b	http://www.foo.com/~fred	http://www.foo.com/%7Ffred	%7F이 ~과 같을 때
c	http://www.foo.com/x.html#early	http://www.foo.com/x.html#middle	태그에 따라 페이지가 바뀌지 않을 때
d	http://www.foo.com/readme.htm	http://www.foo.com/README.HTM	서버가 대소문자를 구분하지 않을 때
e	http://www.foo.com/	http://www.foo.com/index.html	기본 페이지가 index.html일 때
f	http://www.foo.com/index.html	http://209.231.87.45/index.html	www.foo.com이 이 아이피 주소를 가질 때

표 9-1 같은 문서를 가리키는 다른 URL들

9.4 로봇 차단하기

robots.txt

- 어떤 웹 서버의 문서 루트에 robots.txt 라고 이름 붙은 선택적 파일을 제공할 수 있다.
- 이 파일은 어떤 로봇이 서버의 어떤 부분에 접근할 수 있는지에 대한 정보가 담겨있다.
- 만약 로봇이 자발적 표준에 따르면 웹 사이트의 리소스에 접근하기 전 우선 사이트의 robots.txt 를 요청할 것이다.

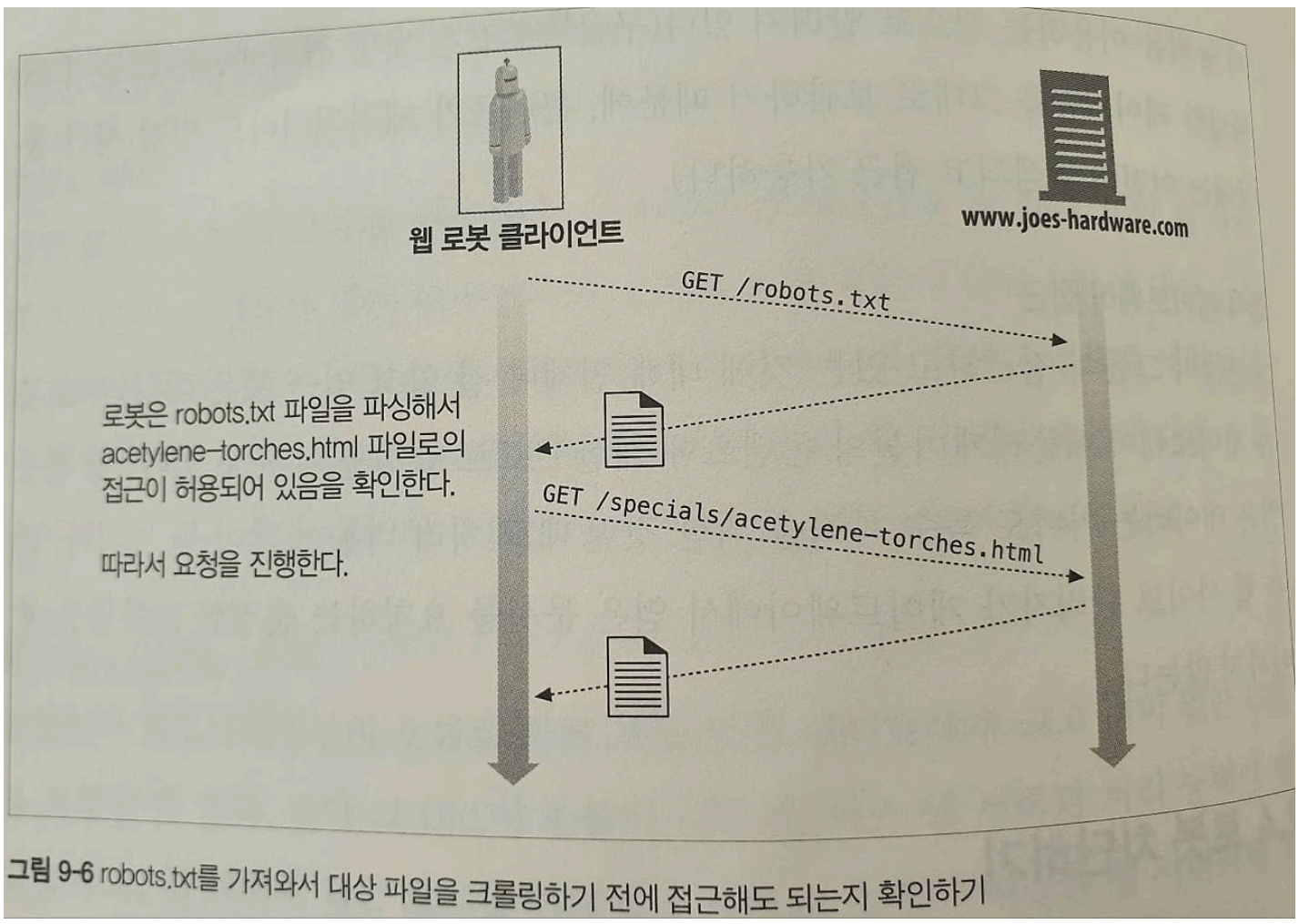

그림 9-6 robots.txt를 가져와서 대상 파일을 크롤링하기 전에 접근해도 되는지 확인하기

9.4.1 로봇 차단 표준

버전	이름과 설명	날짜
0.0	로봇 배제 표준-Disallow 지시자를 지원하는 마틴 코스터(Martijn Koster)의 오리지널 robots.txt 메커니즘	1994년 6월
1.0	웹 로봇 제어 방법-Allow 지시자의 지원이 추가된 마틴 코스터의 IETF 초안	1996년 11월
2.0	로봇 차단을 위한 확장 표준-정규식과 타이밍 정보를 포함한 숀 코너(Sean Conner)의 확장. 널리 지원되지는 않는다.	1996년 11월

오늘날 대부분의 로봇들은 v0.0 이나 v1.0 표준을 채택했다.

9.4.2 웹 사이트와 robots.txt 파일들

robots.txt 가져오기

- 로봇은 웹 서버의 여느 파일들과 마찬가지로 HTTP GET 메서드를 이용해 robots.txt 리소스를 가져온다. 그 robots.txt 가 존재하면 서버는 그 파일을 text/plain 본문으로 반환
- 서버가 404 Not Found 로 응답 시 그 서버는 로봇의 접근을 제한하지 않는 것으로 간주하고 로봇은 어떤 파일이든 요청하게 될 것이다.
- 로봇은 사이트 관리자가 로봇의 접근을 추적할 수 있도록 From 이나 User-Agent 헤더를 통해 신원 정보를 넘기고, 사이트 관리자가 로봇에 대해 문의나 불만사항이 있을 경우를 위해 연락처를 제공 해야 한다.

```
GET /robots.txt HTTP/1.0
Host: www.joes-hardware.com
User-Agent: Slurp/2.0
Date: Wed Oct 3 20:22:48 EST 2001
```

응답코드

- 많은 사이트가 robots.txt 를 갖고 있지 않지만, 로봇은 그 사실을 모른다. 로봇은 어떤 웹사이트든 반드시 robots.txt 를 찾아보고 검색 결과에 따라 다르게 동작한다.

9.4.3 robots.txt 파일 포맷

robots.txt 파일은 매우 단순한 줄 기반 문법을 가진다.

```
샘플
User-Agent: slurp
User-Agent: webcrwaler
Disallow: /private

User-Agent: *
Disallow:
```

User-Agent

- 각 로봇의 레코드는 하나 이상의 User-Agent 줄로 시작되며 형식은 아래와 같다.

```
User-Agent: <robot-name>

or

User-Agent: *
```

만약 로봇이 자신의 이름에 대응하는 에이전트 줄을 못찾으면 접근제한이 없다는 의미이다.

Disallow , Allow

- 어떤 URL 경로가 명시적으로 금지되어 있고 허용되는지를 기술한다.

9.4.4 그 외에 알아둘 점

- robots.txt 파일은 명세가 발전함에 따라 User-Agent, Disallow, Allow 외의 다른 필드를 포함할 수 있으며 로봇은 자신이 이해하지 못하는 필드는 무시해야 한다.
- 주석은 파일 어디든지 허용된다.
- 하위 호환성을 위해, 한 줄을 여러 줄로 나누어 적는 것은 허용되지 않는다.

9.4.5 robots.txt 캐싱과 만료

로봇 명세 초안은 Cache-Control 지시자가 존재할 경우 7일간 캐싱하도록 하고 있다.

...

9.6 검색 엔진

인터넷 검색엔진은 사용자가 전 세계의 어떤 주제에 대한 문서라도 찾을 수 있게 해 준다.

9.6.2 현대적인 검색엔진의 아키텍처

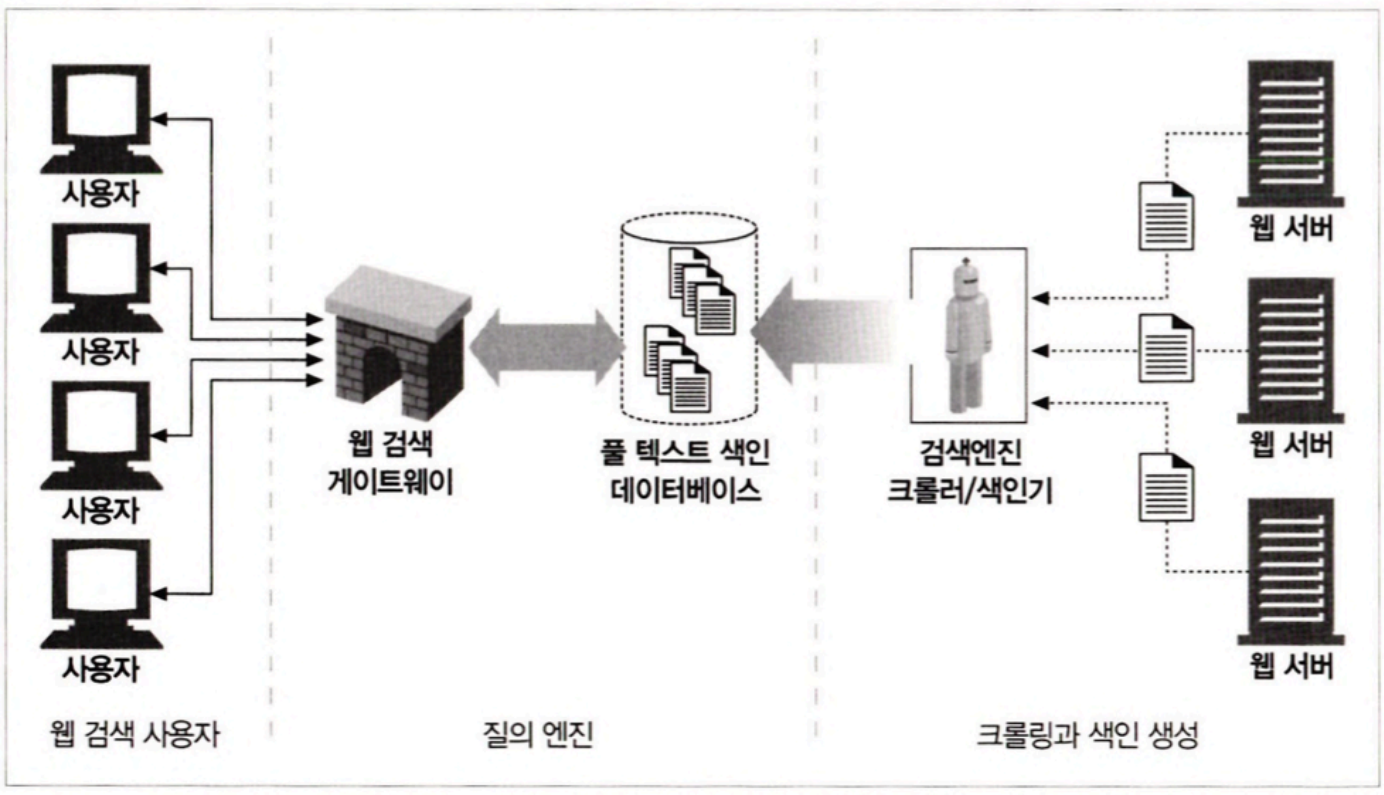

그림 9-7 크롤러와 질의 게이트웨이의 협업을 포함한 상용 검색엔진

풀 텍스트 색인

- 웹의 모든 문서에 대한 일종의 카드 카탈로그처럼 동작
- 실제 fast response 를 위해 종종 사용되며 Redis, KeyDB 등이 쓰인다.
- 단어 하나를 입력받아 그 단어를 포함하고 있는 문서를 즉각 알려주는 DB

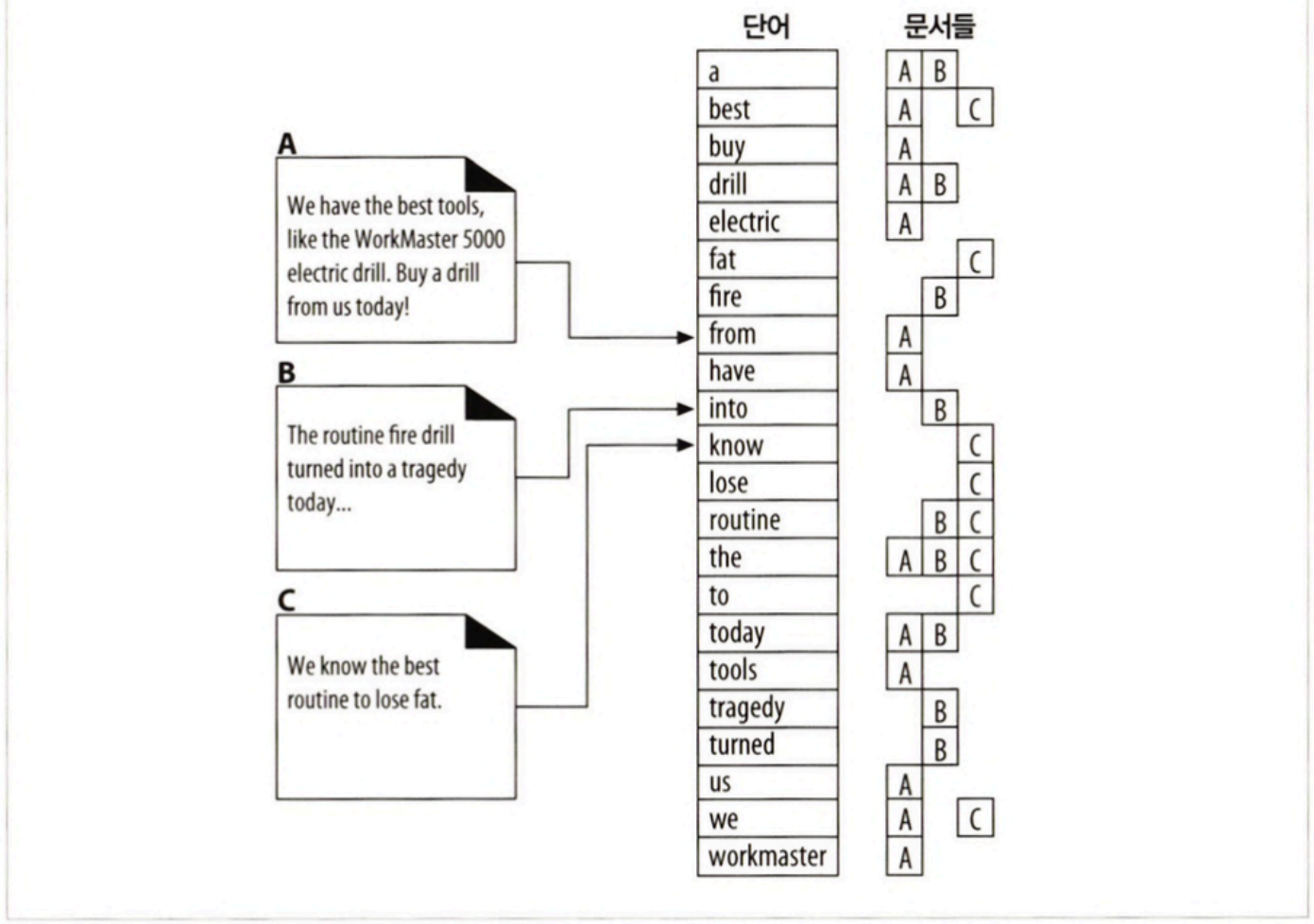

그림 9-8 세 문서와 풀 텍스트 색인

검색 결과를 정렬하고 보여주기

- 검색엔진은 그 문서들이 주어진 단어와 가장 관련이 많은 순서대로 결과 문서에 나타낼 수 있도록 문서들 간 순서를 알 필요가 있다. → 관련도 랭킹
- 많은 검색엔진이 웹을 크롤링하는 과정에서 수집된 통계데이터를 실제로 사용함
  - 검색엔진 = 데이터 장사꾼
  - 크롤링 = 데이터 채굴
  - 색인 DB = 데이터 웨어하우스
  - 랭킹 알고리즘 = 데이터 가공 & 점수화