링크 분석을 통한 비동기 웹 페이지 크롤링 알고리즘

원동현 · 박혁규 · 강윤정 · 이민혜*

원광대학교

Asynchronous Web Crawling Algorithm

Dong-Hyun Won · Hyuk-Gyu Park · Yun-Jeong Kang · Min-Hye Lee*

Wonkwang University

E-mail : dhwon79@wku.ac.kr / hgpark7@wku.ac.kr / yjkang66@wku.ac.kr / lmh3322@wku.ac.kr

요 약

웹은 처리 속도가 다른 다양한 정보들을 함께 제공하기 위해 비동기식 웹 기술을 이용한다. 비동기 방식에서는 작업 완료 전에도 다른 이벤트에 응답할 수 있다는 장점이 있으나 일반적인 크롤러는 웹페이지의 방문 시점 정보를 수집함으로 비동기 방식으로 제공되는 정보를 수집하는 데 어려움이 있다. 또한 비동기식 웹 페이지는 페이지 내용이 변경되어도 웹 주소가 변하지 않는 경우도 많아 크롤링하는 데 어려움이 있다. 본 논문에서는 웹의 링크를 분석하여 비동기 방식 페이지 이동을 고려한 웹 크롤링 알고리즘을 제안한다. 제안한 알고리즘으로 비동기 방식으로 정보를 제공하는 TTA의 정보통신용어사전 정보를 수집할 수 있었다.

ABSTRACT

The web uses an asynchronous web method to provide various information having different processing speeds together. The asynchronous method has the advantage of being able to respond to other events even before the task is completed, but a typical crawler has difficulty collecting information provided asynchronously by collecting point-of-visit information on a web page. In addition, asynchronous web pages often do not change their web address even if the page content is changed, making it difficult to crawl. In this paper, we propose a web crawling algorithm considering asynchronous page movement by analyzing links in the web. With the proposed algorithm, it was possible to collect dictionary information on TTA terms that provide information asynchronously.

키워드

crawler, asynchronous web, archiving, TTA

Ⅰ. 서 론

웹 문서의 형태가 다양한 멀티미디어를 제공함과 동시에 사용자 맞춤형 환경을 제공하면서 웹은 전통적인 정적인 형태에서 동적인 형태로 변화하고 있다[1]. 기술적으로는 기존 html으로만 문서를 작성하는 방식에서 javascript를 활용한 AJAX[2] 비동기 방식으로 변화하며 페이지의 이동 없이도 변경된 정보를 즉시 제공하는 것도 가능해졌다. 하지만 이러한 변화는 기존 방식의 웹 크롤러가 정보를 수집하는데는 어려움으로 작용한다. 기존 크롤러는 웹 사이트에 특정 시점에 접속하여 정보를 수집하는데 비동기방식의 웹은 특정 시점에 따라 정보가 변경되기 때문이다. 예를 들면 TTA(한국정보통신기술협회)에서 제공하는 용어 사전은 사용자가 질의 하기 전까지는 정보를 제공하지 않고, 버튼을 클릭하거나 질의어를 입력할 때 페이지 내용이 변경되어 정보를 제공한다. 이처럼 폼에 특정 정보를 입력해야 하는 경우 검색어로 사용할 질의어를 관리해야 하는데 이 경우 방대한 질의어와 중복된 결과를 처리해야 하는 어려움이 있다. 이러한 문제를 해결하기 위해 본 논문에서는 웹페이지의 링크를 분석하여 동적으로 변경되는 정보를 수집할 수 있는 크롤링 알고리즘을 제안한다.

* corresponding author

II. 관련 연구

크롤러는 사이트 방문이나 크롤러 실행 방식에 따라 집중 크롤러(Focused Crawler), 증분 크롤러(Incremental Crawler), 분산 크롤러(Distributed Crawler), 병렬 크롤러(Parallel Crawler)로 구분지어 지며 [3] 심층웹 크롤러로는 Deepbot, HiWE, Incremental Web Crawler가 있다.[4]

증분 크롤러 (Incremental Crawler) 방식은 크롤링 중인 콘텐츠 원본에 지정되어 있는 웹문서를 크롤링하여 마지막 크롤링 이후 수집되지 않은 정보를 수집하여 기존 크롤링 정보에 변경 내용만 추가하는 방법이다.

분산 크롤러 (Distributed Crawler) 방식은 웹문서를 수집하는 과정에 시간이 많이 소요되는 문제를 해결하기 위한 방안으로 분산시스템 기반 크롤러로 크롤러가 동작하는 다수의 서버가 동시에 웹을 수집하고 중심역할을 하는 서버를 두고 각 크롤러 서버를 관리하여 웹문서를 수집한다.

병렬 크롤러 (Parallel Crawler) 방식은 웹의 규모가 커짐에 따라 하나의 프로세스 또는 스레드만으로 전체 웹 페이지를 수집하기에는 어려움을 해결하기 위한 방안으로 개발되었다. 여러 개의 프로세스 및 스레드를 이용하여 대량의 웹 페이지들을 빠르게 수집한다.

Deepbot은 미니웹브라우저라는 클라이언트 스크립트 실행 도구를 내장하여 서버와의 세션 유지 및 클라이언트의 스크립트 실행이 가능하도록 하여 데이터를 수집한다.

HiWE은 웹 질의어 인터페이스에 숨겨진 데이터를 추출하기 위해 입력폼을 분석하고 입력폼에 값을 전달하는 방식으로 심층웹을 수집한다.

Incremental Web Crawler는 심층웹이 제공하는 정보의 변화에 즉시 반영된 결과를 저장하기 위한 크롤러로 크롤러가 웹페이지를 방문하는 방문주기를 확률적으로 정하고 웹페이지 변화 주기를 계산하여 재방문하여 정보를 수집한다.

III. 링크 분석 기반 크롤링 알고리즘

본 논문에서는 스크립트 명령어를 링크로 관리하여 크롤러가 가능한 많은 정보를 수집할 수 있도록 링크 주소와 스크립트 주소를 함께 관리한다. 먼저 정보 수집을 위한 seedURL로 정적인 주소를 수집하고 해당 seedURL에서 실행되는 스크립트들의 실행 결과를 분석하여 역할에 따라 관리한다. 웹 페이지에 있는 링크의 실행 결과에 따라 표 1과 같이 관리한다. 링크 클릭시 페이지도 이동하고 내용도 변경되는 경우 동적으로 웹 페이지가 변경되는 경우로 정적 주소로 처리하기 위해 page URL로 저장한다. 페이지 이동은 했으나 데이터 변경이 없는 경우는 특정 페이지 내 새로운 정보가 없는 경우로 탐색을 종료하고 다음 Page URL로 이동한다. 페이지 이동이 없이 내용만 변경된 경우, 비동기 방식으로 웹 페이지가 변경된 것이므로 해당 스크립트를 계속 실행하여 정보를 수집한다. 링크를 클릭했을 때 페이지 이동도 없고 내용 변경도 없는 경우는 데이터 수집이 종료된 것으로 판단한다.

표 1. 링크 분석 및 관리

페이지 이동	내용 변경	관리
예	예	Page URL로 저장
예	아니오	데이터 없음, 다음 Page URL로 이동
아니오	예	스크립트 실행 및 데이터 수집
아니오	아니오	데이터 수집 종료

IV. 알고리즘 실험 및 평가

본 논문에서는 실험을 위해 정보통신 용어사전 (http://terms.tta.or.kr/dictionary/searchFirstList.do) URL링크를 분석하여 정보를 수집하였다.

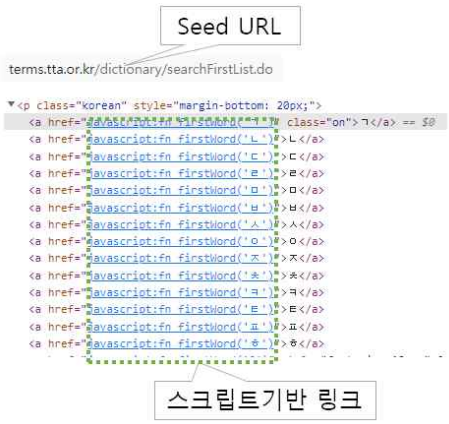

그림 1. URL 링크 스크립트 분석 예시

정보통신용어사전 정보를 수집하기 위해 그림 1과 같이 링크 정보를 분석하였으며 seed URL이 변경되지 않으면서 스크립트 기반의 링크들이 동작하는 것을 확인하였다. a 태그의 href 속성은 웹브라우저에서 페이지 이동 주소를 제공하는 속성이다. 하지만 그림과 같이 href 속성에 javascript 이벤트가 있는 경우는 웹 페이지 이동이 일어나지는 않지만 웹 페이지 내 정보는 변경된다. 이러한 특성을 고려하여 본 논문에서는 그림 2와 같이 알고리즘을 구성하였다.

```
Loop: //모든 SeedURL을 방문할 때 까지 반복

  WebBrowser.Navigate("URL to Crawl")
//크롤링 하기 위한 URL이동
  Delay(WebBroserReady.State.Complete)
//페이지가 로드 대기
  Loop : // 모든 스크립트 실행
  WebBrowser.Invoke.GetNextScript()
  // 로드된 페이지의 스크립트 실행
  Delay(wait Time)  // 스크립트 실행 대기
  Output = WebBrowser.Document // 실행
결과 저장
  EndLoop //스크립트 실행 종료

  foreach(datatype dt in Output) //페이지
분석 및 저장
    if(dt.type = content) SaveContent(dt)
    if(dt.type= url)SaveURL(dt)
    if(dt.type = script)SaveScript()
```

그림 2. 알고리즘 의사코드

URL을 입력 받게 되면 페이지가 관련 정보들을 로드하여 스크립트가 실행 가능한 상태가 될 때까지 기다리며 그 후 스크립트를 실행한다. 스크립트가 실행된 결과를 리턴 받기 위해 잠시 대기하였다가 변경된 정보를 분석하여 스크립트를 실행할 것인지 다음 페이지로 이동할 것인지 판단 결과에 따라 다음 단계로 진행하게 된다.

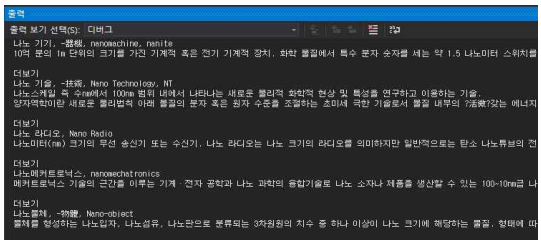

그림 3. 크롤링 실행 화면

알고리즘은 C#으로 구현했으며 VisualStudio 2019 버전의 Webbrowser객체에서 해당 페이지를 로드하여 스크립트를 실행하도록 구현 하였다. Webbrowser가 제공하는 Invoke관련 메서드는 웹페이지내의 스크립트를 실행하고 웹페이지 정보를 업데이트한다. 또한 구조화된 웹 문서를 객체 형태로 전달 하는데, 웹문서 정보에서 사전 용어 및 의미만을 따로 추출하였다. 그림 3은 알고리즘을 적용한 데이터 추출 과정을 보여준다. 크롤링 결과 4821개의 용어를 수집할 수 있었다.

Ⅴ. 결 론

본 논문에서는 웹페이지내 페이지 이동 관련 스크립트를 링크로 분석하여 데이터를 수집하는 크롤링 알고리즘을 제안하였다. 웹페이지의 이동 관련 스크립트에 대한 분석을 사람이 직접 처리해야 하는 단점이 있었지만, 비동기 방식으로 제공되는 정보들을 수집할 수 있었다. 비동기 방식 웹페이지가 점점 늘어나고 있는 상황에서 기존 크롤러가 수집할 수 없는 정보들을 수집하기 위한 다양한 알고리즘이 제안될 것이다.

References

[1] San Murugesan, "Understanding Web 2.0," *IT Professional*, Vol. 9, pp. 34-41, Issue. 4, Aug. 2007.

[2] MDN Web Docs, AJAX[Internet]. Available : https://developer.mozilla.org/ko/docs/Web/Guide/AJAX.

[3] Rahul kumar, Anurag Jain, Chetan Agrawal, "SURVEY OF WEB CRAWLING ALGORITHMS," *Advances in Vision Computing: An International Journal*. Vol. 3, No. 3, Sep. 2016.

[4] Desai Keyur, Devulapalli Virala, Agrawal Smita, Kathiria Preeti, Patel Atul, "Web Crawler : Review of Different Types of Web Crawler, Its Issues, Applications and Research Opportunities," *International Journal of Advanced Research in Computer Science*, Vol. 8, Issue 3, pp. 1199-1202, Mar. 2017.