Journal of Korea Multimedia Society Vol. 21, No. 2, February 2018(pp. 199-209)
https://doi.org/10.9717/kmms.2018.21.2.199

정형 및 비정형 데이터 수집을 위한 웹 크롤러 시스템 설계 및 구현

배성원[†], 이현동[††], 조대수[†††]

Design and Implementation of a Web Crawler System for Collection of Structured and Unstructured Data

Seong Won. Bae[†], Hyun Dong Lee[††], DaeSoo Cho[†††]

ABSTRACT

Recently, services provided to consumers are increasingly being combined with big data such as low-priced shopping, customized advertisement, and product recommendation. With the increasing importance of big data, the web crawler that collects data from the web has also become important. However, there are two problems with existing web crawlers. First, if the URL is hidden from the link, it can not be accessed by the URL. The second is the inefficiency of fetching more data than the user wants. Therefore, in this paper, through the Casper.js which can control the DOM in the headless brwoser, DOM event is generated by accessing the URL to the hidden link. We also propose an intelligent web crawler system that allows users to make steps to fine-tune both Structured and unstructured data to bring only the data they want. Finally, we show the superiority of the proposed crawler system through the performance evaluation results of the existing web crawler and the proposed web crawler.

Key words: Web Crawler System, Cluster, Headless Browser Testing Framework, Unstructured Data

1. 서 론

최근 빅 데이터를 활용한 최저가 쇼핑, 맞춤형 광고, 상품 추천 등 빅 데이터를 기반으로 다양한 분야의 서비스와 접목되면서 빅 데이터에 대한 중요성은 더욱 커지고 있다. 이로 인해 웹상에서 데이터를 자동으로 수집하는 웹크롤러에 대한 중요성 또한 높아지고 있다. 하이퍼링크로 웹 사이트들이 거미줄처럼 연결되어 있는 웹 환경에서 텍스트, 이미지, 동영상 등 웹 문서에 내포하고 있는 자료를 수집하고 하이퍼링크를 통해 다른 웹 사이트에 접근하여 다시 자료 수집을 하는 자동화 프로그램을 웹크롤러라고 한다.

여기서 하이퍼링크를 통해 다른 웹 사이트에 접근하는 방식에서 첫 번째 문제점이 도출된다. HTML 안에서 하이퍼링크는 A태그이다. A태그는 href 속성을 가지고 있는데 이 속성은 다른 웹사이트의 URL 주소를 가지고 있다. 이 주소를 통해서 웹 크롤러는 웹 사이트에 접근을 할 수 있다. 문제는 자바스크립트 함수를 사용하면 URL 주소를 은닉할 수 있기 때문에 은닉된 URL에 대해서는 기존의 웹크롤러는 접근하여 수집할 수 없다. 두 번째 문제점은 기존의 크롤러가 태그 및 텍스트와 같은 정형 데이터와

※ Corresponding Author : DaeSoo Cho, Address: (47011) Jurye-ro 47, Sasang-gu, Busan, Korea, TEL : +82-51-320-1964, E-mail : dscho@dongseo.ac.kr
Receipt date : Jan. 12, 2018, Approval date : Jan. 23, 2018
[†] Dept. of Division of Computer Engineering. Dongseo University (E-mail : seongwon9179@gmail.com)
[††] Industry Academy Cooperation Foundation, Dongseo University (E-mail : win4class@hanmai.net)
[†††] Dept. of Division of Computer Engineering. Dongseo University (E-mail : dscho@dongseo.ac.kr)
※ Following are results of a study on the "University for Creative Korea-1" Project, supported by the Ministry of Education

비디오, 오디오와 같은 비정형 데이터를 구분해서 수집에는 한계가 존재한다.

본 논문에서는 Casper.js기반의 웹크롤러 시스템을 제안한다. 제안하는 웹 크롤러 시스템은 은닉된 URL에 접근하기 위해 Headless Browser환경에서 Casper.js를 활용하여 은닉된 URL에 접근을 하여 수집할 수 있는 기능과 브라우저의 기능인 확장 프로그램을 통해 사용자가 원하는 데이터 형(텍스트, 비디오, 오디오 등)과 수집하고 싶은 특정영역을 설정할 수 있다. 설정된 값을 바탕으로 수집을 하는 기능을 가진다.

본 논문의 구성은 다음과 같다. 2장에서 기존 웹 크롤러의 기능과 문제점 및 관련 기술을 살펴보고, 3장에서 제안하는 Casper.js를 이용한 웹 크롤러 시스템을 제안한다. 4장에서는 기존의 웹 크롤러와 제안하는 웹 크롤러 시스템간의 성능 비교를 통해 제안하는 웹 크롤러 시스템에 대해 평가를 하고 5장에서 결론 및 향후 연구 방향을 논한다.

2. 관련연구

2.1 일반 웹 크롤러 기술

웹 크롤러란 조직적, 자동화된 방법으로 월드 와이드 웹을 탐색하는 컴퓨터 프로그램이다. 웹 크롤러가 하는 작업을 웹 크롤링(web crawling) 혹은 스파이더링(spidering)이라 부르며 봇이나 소프트웨어 에이전트의 한 형태이다. 웹 크롤러는 크게 일반 웹 크롤러와 분산 웹 크롤러가 있다. Fig. 1는 웹 크롤러의 전체적 흐름을 나타낸다.

웹 크롤러의 기본 동작은 다음과 같다. 크롤러는 URL Frontier 모듈에서 URL을 가져와 http 프로토콜을 사용해 해당 URL의 웹 페이지를 가져오는 것으로 시작한다. 그런 다음 Fetch 모듈에서 임시 저장소에 웹 페이지를 저장한다. Parse 모듈에서 텍스트와 링크를 추출을 하고 텍스트는 Indexer에 보내진다. 링크의 경우는 URL Frontier에 추가되어야 하는지에 대해 Content Seen, URL Filter, Duplication URL Element 모듈들을 거치면서 판단하게 된다[1]. 하지만 문제는 링크에서 은닉된 URL의 경우에는 URL Frontier에 추가대상인 URL임에도 불구하고 은닉되어 있어 추가되지 못하는 한계가 있다.

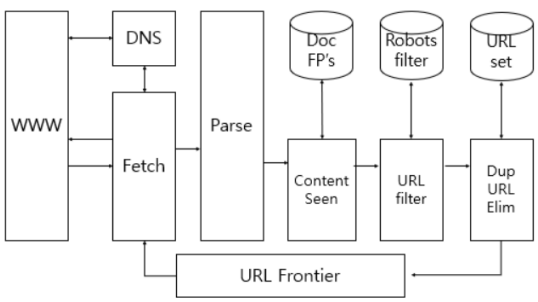

Fig. 1. Conventional Basic Web Crawler Architecture.

2.2 분산 웹 크롤러 기술

전 세계의 웹 문서를 전부를 일반 웹 크롤러로 크롤링 한다는 것은 사실상 불가능하기 때문에 분산 웹 크롤러를 해야만 한다. 분산 웹 크롤러는 크게 2가지로 나누어지는데 그중 하나가 구글에서 사용한 중앙 집중식(Centralized) 방식이고 다른 하나는 Mercator나 다른 곳에서 사용한 P2P(or Fully-Distributed) 방식이다[2,3,4,5]. Fig. 2는 중앙 집중식 분산 웹 크롤러 구조를 나타낸다.

중앙 집중식 분산 웹 크롤러는 URL Manager가 서버와 같은 역을 하며 크롤러가 클라이언트 역을 하는 구조이다. 크롤러에서 문서를 다운로드 받고 OutLink URL을 추출하여 URL Manager에게 넘겨주면 URL Manager는 다운로드 받은 문서의 URL인지 검사하여 URL 중복을 제거를 한다. 즉 일반 웹 크롤러에서 URL 중복과 URL 관리를 하는 부분을 URL Manager가 대신 해 주는 것이다. P2P 방식은 Fig. 3과 같이 각 Crawler가 완전 독립 적인 구조를 가진다.

P2P(peer to peer, or fully distributed)방식은 각각의 크롤러가 일반 웹 크롤러처럼 동작을 한다. 각각의 크롤러는 문서를 다운로드 받고 OutLink URL을

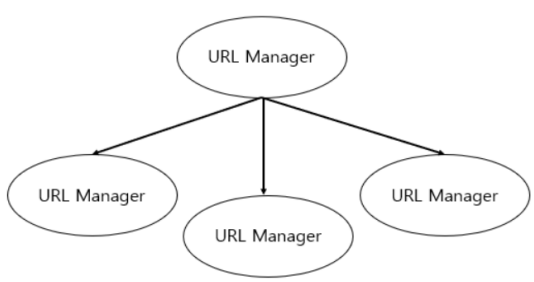

Fig. 2. Centralized Distributed Web Crawler Structure diagram.

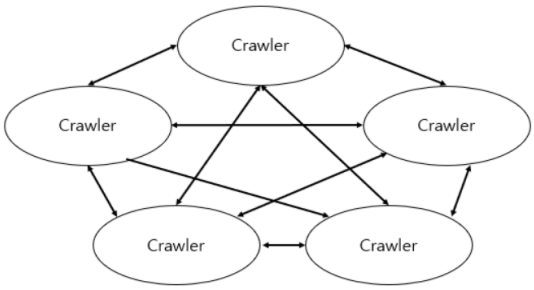

Fig. 3. P2P distributed Web Crawler Structure diagram.

추출하고 URL 중복제거까지 모두 각각의 Crawler가 독립적으로 동작한다. 이렇게 하기 위해서는 각각의 Crawler에서 관리하는 다운로드 받은 URL 목록은 서로 배타적이어야 한다. 그렇지 않으면 서로 다른 Crawler에서 같은 문서를 다운로드 받는 현상이 발생할 것이다. 이것을 해결하는 방법으로 각각의 Crawler는 다운로드 받을 URL Domain을 서로 배타적으로 나눠서 관리한다. 즉, 자신이 다운로드 domain에 속하는 것만 관리하고 나머지 URL은 다른 Crawler에게 넘기는 것이다. 이렇게 하면 각각의 Crawler가 독립적으로 동작할 수 있게 된다[6].

2.3 웹 콘텐츠 추출 기술

웹 콘텐츠 추출 기술은 웹 문서로부터 정보 분석에 활용될 콘텐츠인 제목, 작성자, 게시일, 본문을 자동으로 추출하는 기능을 제공한다. 웹 콘텐츠 추출 시스템은 콘텐츠를 추출하는 규칙을 자동 생산해 콘텐츠만을 추출하는 장치로 콘텐츠 추출 규칙을 자동 생성하는 규칙 생성기(Rule Generator), 주어진 웹 문서에서 내비게이션 콘텐츠를 제거하는 내비게이션 콘텐츠 제거기(Navigation Content Eliminator), 콘텐츠 추출 규칙 키워드 유사도 비교를 통해 콘텐츠를 추출하는 콘텐츠 추출기(Core Context Extractor)로 구성되어 있다[7]. Fig. 4는 웹 콘텐츠 추출 시스템의 구성을 나타낸다.

기존의 연구된 웹 크롤러 기술과 본 논문에서 제안하는 정형 및 비정형 데이터 수집을 위한 웹 크롤러 시스템의 차이는 은닉된 URL의 데이터를 수집하기 위해 기존의 웹 크롤러와 Headless Browser를 함께 구성한다는 점에서 차이가 있다. 그리고 이를 효율적으로 운용하기 위해 멀티미디어 데이터 수집과 은닉된 URL을 수집하기 위한 롤을 정의한다.

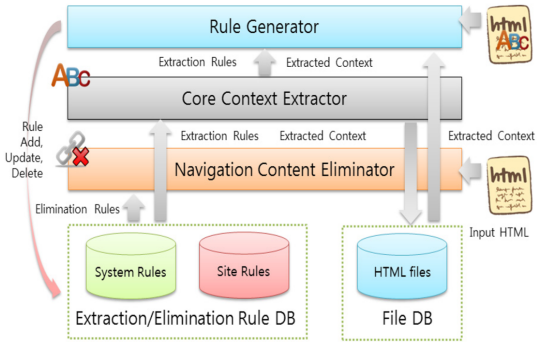

Fig. 4. System composition diagram of web contents extractor.

3. 정형 및 비정형 데이터 수집을 위한 웹 크롤러 시스템

3.1 정형 및 비정형 데이터를 가져오기 위한 웹크롤러 시스템 구성

제안하는 웹 크롤러는 정형 및 비정형 데이터를 효율적으로 수집할 수 있는 기능[8]과 링크에서 은닉된 URL의 웹 페이지도 수집할 수 있는 크롤링 시스템이다. 서버는 비동기로 동작하는 Node.js를 기반으로 기존 URL을 통해 크롤링하는 Crawler Service와 은닉된 URL을 크롤링하기 위한 Casper Service를 포함한다. 클라이언트는 사용자가 크롤링을 하고 싶은 특정 영역과 수집하고 싶은 데이터 타입을 설정할 수 있는 확장프로그램과 크롤링된 결과를 볼 수 있는 모니터링을 포함한다. 그림 Fig. 5는 제안하는 웹 크롤러 시스템의 구성을 나타내고, Table 1은 제안하는 웹 크롤러 시스템의 상세 설명을 나타낸다.

제안하는 웹크롤러 웹 서버인 Node.js는 Single Thread 기반으로 Non-blocking IO를 지원하는 고성능 서버이다. 하지만 Single Thread이기 때문에 다수의 웹 페이지를 크롤링하기에는 한계가 존재한다. 이런 문제를 해결하기 위해서 Cluster를 활용하였다. Cluster란 Node.js의 모듈로서 서버의 포트를 공유하는 다수의 프로세스를 생성하여 처리를 할 수 있다. 이를 통하여 Cluster Crawler Service Worker인 쓰레드를 생성하고 크롤링을 함으로써 속도의 한계를 해결하였다.

Web Server Service의 Collection Rules 모듈은 클라이언트로부터 받은 크롤링 규칙과 URL 혹은 XPath로 이루어진 목록인 시드를 받는다. XPath(XML

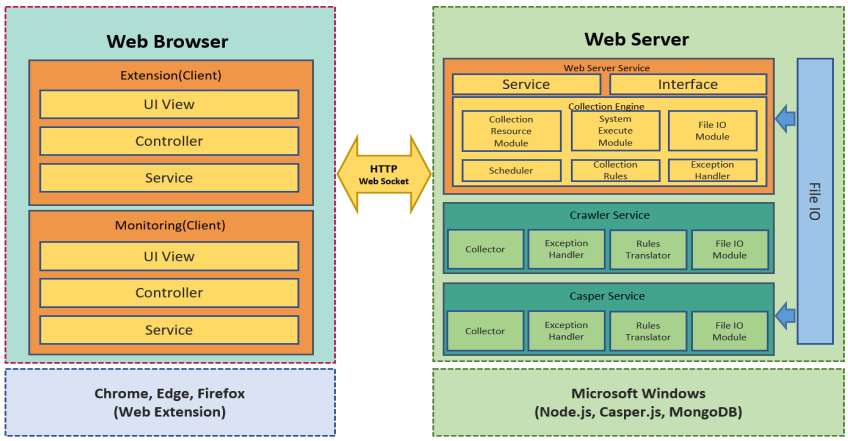

Fig. 5. Web Crawler System Configuration Diagram.

Path Language)란 W3C의 표준으로 확장 생성 언어의 문서의 구조를 통해 경로 위에 지정한 구문을 사용하여 항목을 배치하고 처리하는 방법을 기술하는 언어이다. 크롤링 규칙에는 크롤러가 크롤링할 데이터 타입이 정의한다.

크롤러는 규칙에 정의된 데이터 타입에 따라 수집한 DOM의 태그를 판별한다. HTML 태그는 멀티미디어 데이터에 따라 다르게 기술하기 때문에 영상의 경우는 video 태그를 이미지의 경우는 img 태그를 음악의 경우는 audio 태그로 구분지어 태그에 기술된 멀티미디어 URL을 통해 서버에 다운을 받아서 Collection Resource Module을 통해 관리를 한다.

3.2 웹크롤러 크롤링 규칙 정의 스케쥴 관리 기능

Scheduler 모듈은 Collection Rules 모듈로부터 크롤링 규칙과 시드를 인자로 받아서 시드의 값이 URL일 경우 Crawler Service Worker로 전달하고 크롤링을 수행한다. XPath일 경우 Web Server Service의 File IO 모듈로 호스트 URL, XPath, 크롤링 규칙을 JSON 파일로 저장한다. 저장이 완료되면 System Execute 모듈로 Casper Service를 실행 시킨다. Casper Service는 실행 즉시 File IO 모듈로 저장된 JSON 파일을 읽어 들이고 Rules Translator 모듈로 JSON 파일을 해석하여 웹 페이지를 크롤링 한다. Crawler Service Worker 또는 Casper Service는 크롤링 도중 수집된 비정형 데이터를 각각의 서비스에 포함된 File IO 모듈을 통해 서버에 저장한다. Crawler Service Worker는 크롤링이 완료되면 수집한 HTML과 규칙에 따른 정형 및 비정형 데이터 목록을 따로 만들고 Cluster send 메서드를 호출해 Web Server Service에게 수집된 결과 값을 보낸다. Casper Service의 경우 크롤링이 완료되면 Crawler Service와 똑같이 수집된 결과 값인 HTML과 정형 및 비정형 데이터의 목록을 File IO 모듈을 통해 저장하고 Callback 함수를 실행하여 System Execute 모듈에 알린 후 Scheduler 모듈이 File IO 모듈로 결과 값을 읽어 드린다. Scheduler 모듈은 1개의 웹 페이지가 크롤링이 완료 될 때 마다 Collection Resource 모듈을 실행시켜 저장된 비정형 데이터의 Path를 작성하고 완료되면 Schedule 모듈에 알린다. Schedule 모듈은 HTML, 정형 데이터, 비정형 데이터의 Path 값을 Interface 모듈로 보내고 Interface 모듈은 DB에 저장한다.

Scheduler 모듈은 각각의 웹 크롤러가 수집한 결과물 중 HTML에서 링크를 다시 추출하여 링크의 URL 패턴을 보고 URL인지 아닌지 구분을 하고 URL이 아닐 경우 링크의 XPath값을 추출하고 반대의 경우이면 URL값을 추출하여 Collection Rules의 기존의 시드를 업데이트 하며 자체적으로 크롤링 스케줄을 만든다. 아래 Fig. 6은 Collection Rules의 시드 값이 URL일 경우의 시퀀스 다이어그램이고 Fig. 7은 시드 값이 XPath일 경우의 시퀀스 다이어그램을 나타낸다.

3.3 중복 데이터를 고려한 데이터 저장 기능

웹 크롤러는 거미줄처럼 엉킨 웹의 환경에서 크롤

Table 1. Component Description Table

Division	Component name			Description
Web Server	Web Server Service	Service		A module that handles client requests and responses by HTTP or Websocket.
		Interface		Save data collected by each crawler to DB
		Collection Engine	Collection Resource Module	Module for creating path of unstructured data collected by crawler
			System Execute Module	Window command prompt connection module to run Casper Service
			File IO Module	Module for writing the Collection Rules file to the Casper Service or reading the crawling result data
			Scheduler	Based on URL seed defined in Collection Rules Crawler Service Walker schedule management and Casper service schedule management module
			Collection Rules	A module that manages the collection rules received from the Service and passes the rules to the Scheduler
			Exception Handler	Modules that manage exceptions that occur in Web Server Service modules
	Crawler Service	Collector		Conduct rules and URL-based web crawls interpreted by Rules Translator
		Exception Handler		Modules that manage exceptions that occur in Web Server Service modules
		Rules Translator		A module that interprets the Collection Rules received from the Scheduler so that they can be applied to the Collector.
		File IO Module		A module that stores unstructured data collected by the Collector on the Web server.
	Casper Service	Collector		Conduct rules and XPath-based web crawls interpreted by Rules Translator
		Exception Handler		Modules that manage exceptions that occur in Web Server Service modules
		Rules Translator		A module that interprets the Collection Rules received from the Scheduler so that they can be applied to the Collector.
		File IO Module		A module that reads the rules saved by the Scheduler or stores collected data that have been completed by the Collector
Client	Extension	UI View		Screen for defining collection rules and specifying specific areas
		Controller		Logic module for handling events that occur in the UI View
		Service		Module for communicating with Web Server Service
	Monitoring	UI View		Screen for the administrator to check the results that the crawler has collected
		Controller		Logic module for handling events that occur in the UI View
		Service		Module for communicating with Web Server Service

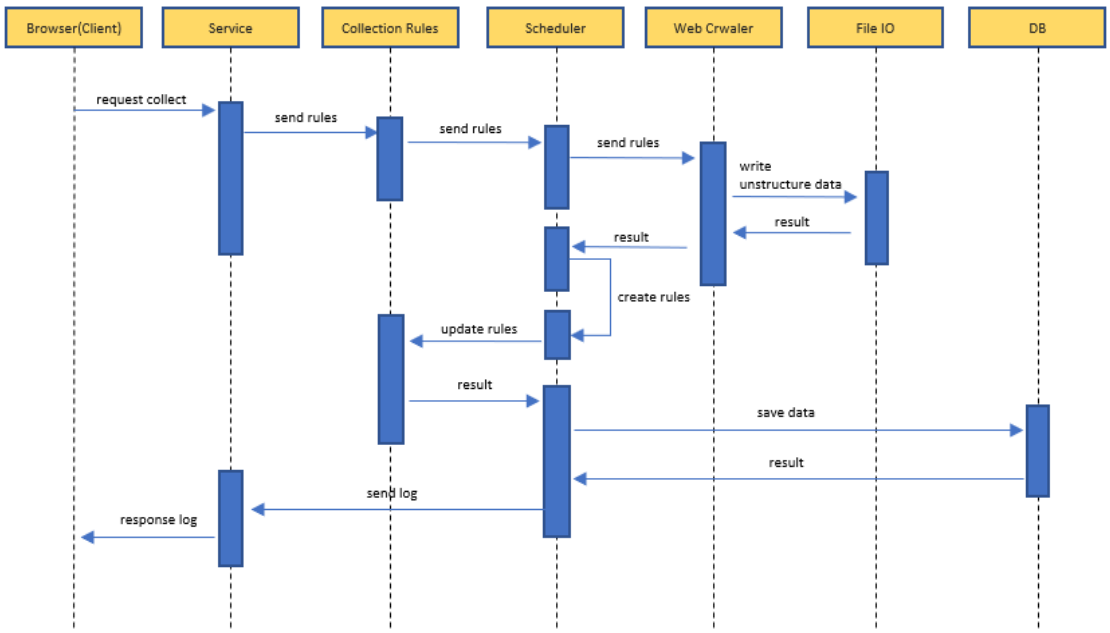

Fig. 6. URL-based web crawler sequence.

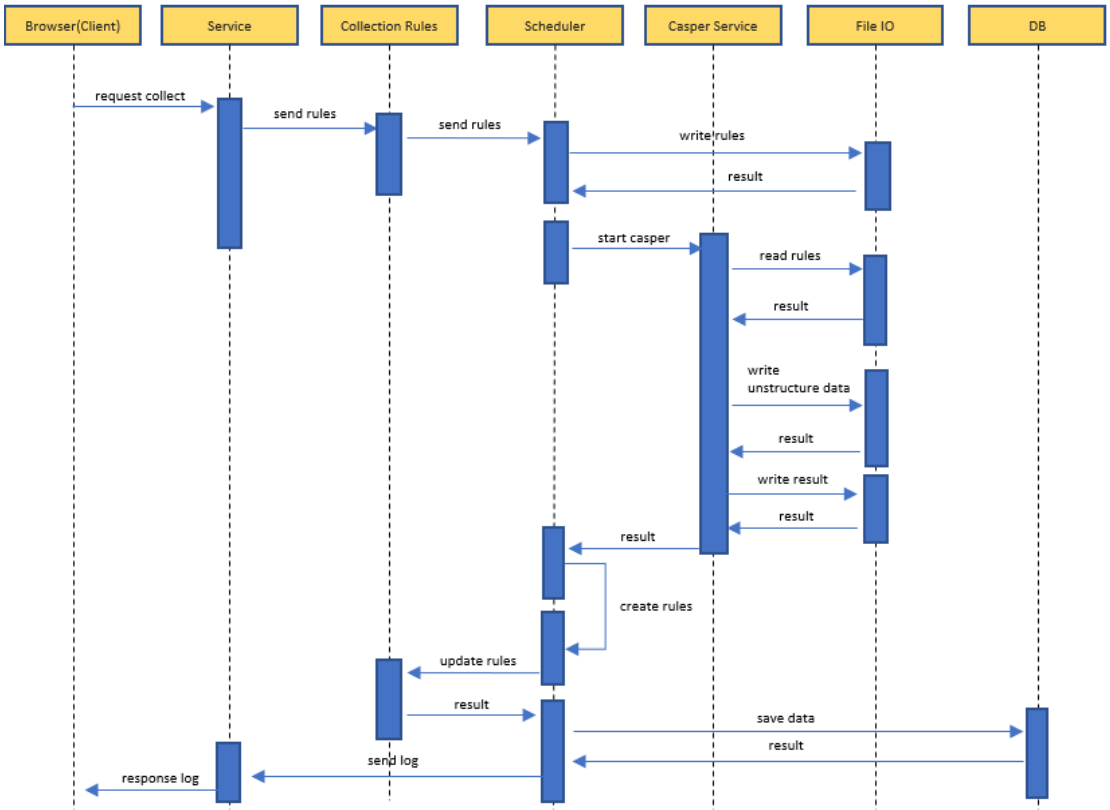

Fig. 7. XPath-based web crawler sequences.

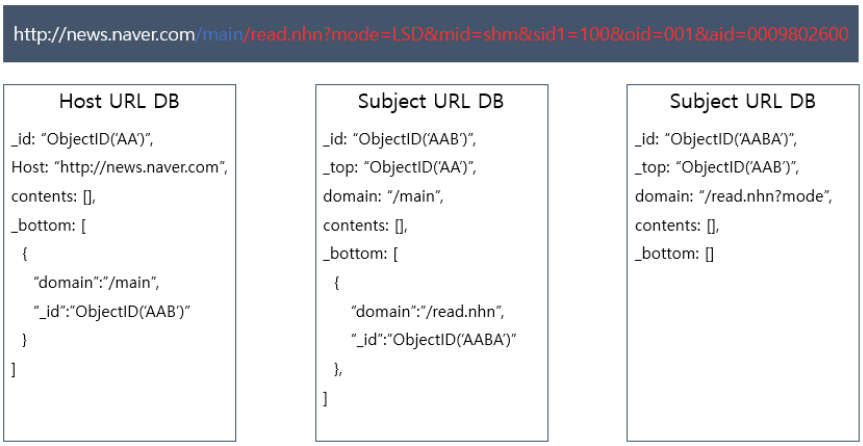

Fig. 8. Database Dataset.

링했던 웹 페이지를 다시 크롤링할 수 있기 때문에 중복 데이터가 발생할 우려가 있다. 이를 방지하기 위해 본 논문에서 제시하는 시스템은 URL을 기반으로 중복된 데이터를 관리한다. DB는 크롤러 특성상 많은 Read & Write가 이루어지고 Schema less한 특성이 있어 어떠한 형태의 데이터라도 저장할 수 있는 특징이 있기 때문에 RDB가 아닌 Mongo DB를 사용하였다. Fig. 8은 URL을 기반으로 DB에 저장되는 형태를 나타낸다.

DB에 저장되는 형태는 Host URL Collection과 Subject URL Collection으로 구분된다. "http://" 문자열 뒤에 붙는 "/" 문자를 기준으로 Host와 Subject를 구분하고 이후의 "/"는 Subject URL로 취급한다. 모든 Document에는 _id 속성이 자동으로 부여가 되는데 이 _id를 통해 크롤러가 탐색하고 있는 URL이 DB에 존재하는 URL인지 찾고 있다면 해당 Document의 Contents 속성을 업데이트 한다. Host URL Document에서 Subject URL Document 또는 Subject URL Document에서 Subject URL Document로 탐색 할 때는 Fig. 8의 내용 중 _bottom, _top 속성 내의 _id 속성을 통해 하위 Document 또는 상위 Document로 이동한다. Fig. 9은 크롤링 과정에서 중복 확인 및 저장하는 하는 과정을 순서도로 나타낸 그림이다.

3.4 Casper.js를 통한 은닉된 URL 크롤링 기능

은닉된 URL로 이루어진 링크는 URL 대신 자바스크립트 함수를 호출하는 형태로 구현되어 있다. 그래서 함수를 구동시키기 위해서는 브라우저 환경에서 접근을 시도하여야 한다. 브라우저 환경 위에서 동작하는 크롤러를 구현해야 하기 때문에 Headless Browser를 이용하였다.

Headless Browser는 GUI가 없이 CLI(Command-Line Interface) 동작하는 웹 브라우저이다. 웹 프로그램의 테스트 자동화, 스크린 샷, 자바스크립트 라이브러리 자동화 테스트, 데이터 스크래핑 등 다양한 목적으로 사용된다. Casper.js는 Headless Browser의 탐색 스크립팅을 제공하는 프레임워크이다. Fig. 10은 Casper.js의 동작 시퀀스를 보여주는 다이어그

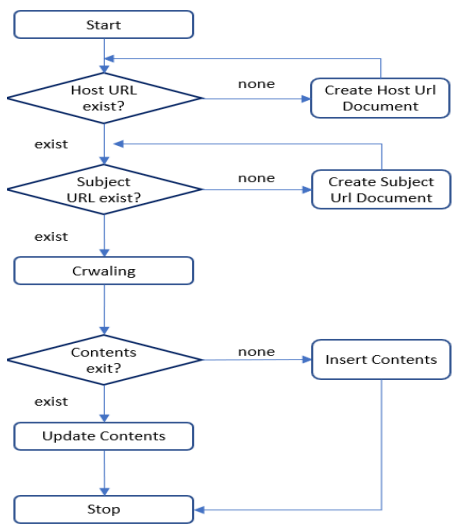

Fig. 9. Duplicate data validation and crawling update flowchart.

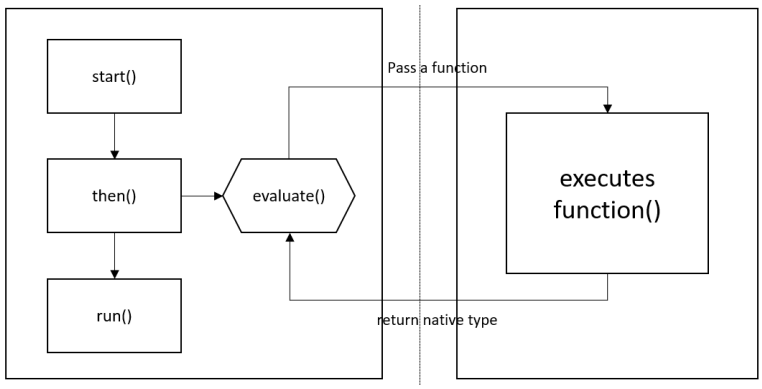

Fig. 10. Casper.js Operation sequence Diagram.

램을 나타낸다.

Casper.js는 start 메서드를 호출하여 Headless Browser를 구동하여 Collection Rules에 정의된 웹 페이지로 접근한다. 그리고 then 메서드에 Callback 함수를 전달하면서 동기적인 크롤러 수행을 한다. Callback 함수 안에 evaluate 메서드를 호출할 수 있다. evaluate 메서드의 인자에 함수가 정의 될 수 있는데 이 함수에는 Headless Browser가 접속한 웹 페이지의 DOM 이벤트를 정의 할 수 있다. 그래서 Collection Rules에 정의된 XPath로 링크태그를 탐색하여 click 이벤트를 발생시키고 링크의 은닉된 URL로 접속을 할 수 있다. 접속이 되면 Collection Rules에 정의된 데이터 유형에 따라 정형 또는 비정형 데이터와 링크를 수집한다.

3.5 크롤링 수집 영역 추출 기능

크롤링 규칙 정의와 수집 영역 추출 기술은 Browser Extension을 통해 구현하였다. Browser Extension은 웹 브라우저의 기존 기능의 동작을 변경하거나 완전히 새로운 기능을 추가하는 웹 브라우저용 프로그램이다. Extension은 Chrome, Firefox, Edge, Opera 브라우저에서 동작 가능하다. Fig. 11은 제안하는 정형 및 비정형 데이터 수집을 위한 웹 크롤러 시스템의 Extension UI를 나타낸다.

Fig. 11의 ①번은 크롤링 규칙을 정하기 위한 메뉴이다. 체크박스에 체크할 수 있는 인터페이스로 사용자가 크롤링 하고 싶은 데이터 타입을 선택할 수 있다. 체크 박스 밑에는 특정영역을 설정할 수 있는 옵션이 있다. 특정영역을 선택하지 않으면 웹 페이지 전부를 크롤링 하지만 특정영역을 선택하면 ②번 UI가 나타나며 UI 상단의 버튼을 누름으로서 특정 영역 추출 기능을 ON / OFF 할 수 있다. 기능이 실행되면 Document 객체의 Mouse Up 이벤트에 마우스 포인터가 가리키는 DOM의 요소를 가져온다.

③번은 마우스 포인터가 가리키는 DOM의 Background-color 속성을 수정하여 사용자가 어떤 DOM을 가리키는지 인식할 수 있게 구현하였다. 마우스를 클릭하면 가져온 DOM 요소를 저장하며 ②번 UI에 DOM 요소를 XPath로 변환하여 리스트로 나타낸다. 마지막으로 ①번의 하단 크롤링 버튼을 클릭하면 선택된 데이터 타입과 특정영역의 XPath 호스트 URL 데이터로 크롤링 규칙을 만들고 웹 서버로 전달한다.

Extension을 통해 만들어진 규칙은 아래 Fig. 12와 같이 Json 구문으로 서술되며 data_type, specific_area, host_url 속성이 존재한다. data_type의 하위 속성은 타입 별 Boolean 값을 가진다. Default 값은 false로 설정 된다. specific_area 속성은 사용자가 추출한 DOM의 XPath 구문으로 변환되어 추가가 된다.

4. 정형 및 비정형 데이터 수집을 위한 웹 크롤러 시스템 평가

성능 비교를 위해 사용한 기존의 크롤러는(Crawler.js)는 Node.js 기반의 웹 크롤러 모듈로서 Node.js에서 사용 가능한 모듈을 설치할 수 있는 사이트인 NPM에서 가장 인기 있는 크롤러이다.

성능 비교를 위한 웹 사이트는 부산시청 홈페이지(http://www.busan.go.kr/netfunnel.jsp)와 고려대학교 홈페이지(http://www.korea.ac.kr)를 비교하였고 게시판의 게시글을 크롤링 하도록 하였다.

Fig. 13은 제안하는 웹 크롤러 시스템과 기존의

Fig. 11. Extension UI.

크롤러의 수집한 웹 페이지의 개수를 나타낸 차트이다. 차트를 보면 웹 페이지를 가져오는 건수의 차이를 확인할 수 있다. 이는 크롤링하려는 페이지의 URL이 은닉되어 있기 때문이다. 기존의 크롤러는 은닉된 URL 링크를 크롤링할 수 없기 때문에 고려대학교에서는 0건 부산시청에서는 14건의 데이터를 수집하였다.

```
{
  "data_type":{
    "text": true,
    "video": true,
    "image": true,
    "audio": false,
    "gif": false
  },
  "specific_area": [
    "//*[@id='PM_ID_themecastBody']/div/div/div/ul"
  ],
  "host_url": "http://www.example.com/main"
}
```

Fig. 12. Collection Rules Json Define.

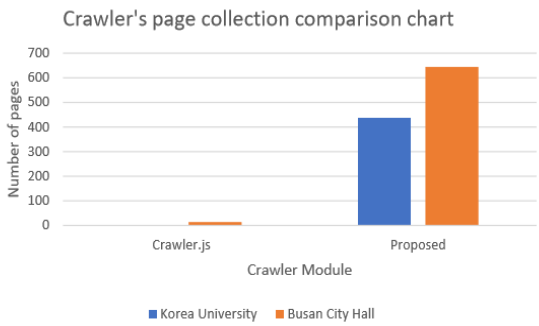

Fig. 13. Crawler's page collection comparison chart.

제안하는 웹 크롤러와 기존의 크롤러의 속도를 평가 하였다. 기존의 크롤러는 위와 동일한 모듈을 사용하였다. 논문에서 제안하는 웹 크롤러와 같은 Node.js 기반이여서 비교에 적합하다고 판단하였다. 비교에 앞서 테스트 환경은 8GM RAM과 2.20GHz의 Intel Core I5, Window 10 운영체제를 갖는 서버와 클라이언트에서 비교 시험을 하였다.

아래 Fig. 14는 총 20개의 무작위 웹 페이지를 선정하였고 10번의 테스트를 진행했다. 처리시간은 웹 페이지를 수집을 시작하는 시점부터 종료하는 시점

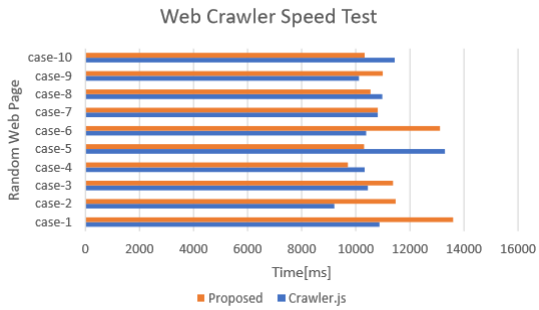

Fig. 14. Web Crawler Speed Test.

까지의 시간을 측정하였다. 결과는 아래와 같이 기존의 크롤러와 비슷한 성능을 보여주었다. 하지만 기존의 크롤러는 은닉된 URL의 링크를 가져오지 못하기 때문에 속도는 비슷하나 수집의 양은 더 우수한 웹 크롤러라는 것을 입증하였다.

5. 결론 및 향후 연구

본 논문에서는 기존의 크롤러가 가진 은닉된 URL 링크 페이지에 접근할 수 없는 문제와 정형 데이터와 비정형 데이터를 구분해서 수집할 수 없는 문제점들을 나열하였고 이를 해결하기 위해 Casper.js를 활용해 Headless Browser에서 직접 DOM 이벤트를 발생하여 접근할 수 있는 웹 크롤러를 제안하였으며, 사용자가 원하는 정형 및 비정형 데이터들인 텍스트, 이미지, 비디오, 오디오 데이터 타입과 크롤링 할 특정 영역을 선택함으로써 더욱 세밀하고 효율적인 수집을 할 수 있는 확장 프로그램 기능을 제시하였다. 또한 기존 웹 크롤러와 제안하는 웹 크롤러의 성능 평가 결과를 보면 기존의 웹크롤러와의 10번의 테스트 케이스에서 평균은 약 437ms 속도차이가 났다. 이는 환경에 따라 변화되는 값이기 때문에 기존의 웹 크롤러와는 속도적인 측면에서 차이가 없지만 수집의 양을 따져보면 매우 큰 차이가 나는 것을 Fig. 13을 보면 알 수 가 있다. 이로써 제안하는 크롤러 시스템의 정형 및 비정형 데이터 수집 기능의 우수성을 입증하였다.

향후에는 기존의 크롤러와 비슷한 속도를 나타내었던 성능을 더욱 발전시켜 속도와 수집하는 양 두 가지 모두 우수한 웹 크롤러 시스템을 개선할 예정이다.

REFERENCE

[1] C.D. Manning, P. Raghavan, and H. Sch¨utze, *Introduction to Information Retrieval,* Cambridge University Press, Cambridge, 2008.

[2] Dustin Boswell, *Distributed High-Performance Web Crawlers: A Survey of the State Of the Art*, 2003.

[3] A. Heydon and M. Najork, “Mercator: A Scalable, Extensible Web Crawler,” *World Wide Web,* Vol. 2, No. 4, pp. 219-229, 1999.

[4] Allan Heydon and Marc Najork, *High-Performance Web Crawling*, COMPAQ SRC Reserch Report 173, 2001.

[5] S. Brin and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” *Proceeding of the Seventh International World Wide Web Conference*, pp. 107-117, 1998.

[6] M.S. Kang and Y.S. Choi, “Design Hadoop Based P2P Distributed Web Crawler,” *Proceeding of Korean Society For Internet Information,* pp. 199-202, 2010.

[7] D.M. Seo and H.M. Jung, “Intelligent Web Crawler for Supporting Big Data Analysis Services,” *Journal of Korea Contents Association,* Vol. 13, No. 12, pp. 575-584, 2013.

[8] Y.H Kim and M.D Chung, “Analysis of Structured and Unstructured Data and Construction of Criminal Profiling System using LSA” *Journal of Korea Multimedia Society,* Vol. 20, No. 1, pp. 66-73, 2017.

배 성 원

2011년~현재 동서대학교 컴퓨터공학부 학사과정
관심분야 : 데이터베이스, 빅데이터, 실시간 이벤트 처리 등

이 현 동

2012년 부경대학교 컴퓨터공학과 졸업(공학박사)
2017년~현재 동서대학교 산학협력단 연구교수
관심분야 : 컴퓨터 보안 응용, 상황 인식 등

조 대 수

1995년 부산대학교 컴퓨터공학과 졸업(공학사)
1997년 부산대학교 컴퓨터공학과 졸업(공학석사)
2001년 부산대학교 컴퓨터공학과 졸업(공학박사)

2001년~2004년 ETRI 텔레매틱스 연구단 선임연구원
2004년~현재 동서대학교 컴퓨터공학부 교수
관심분야 : GIS, 공간데이터베이스. LBS, 빅데이터, 사물인터넷 등