Java로 구현하는 웹 크롤러를 통한 고효율의 자료검색과 여러 생산적인 응용

고려대학교 컴퓨터학과

2018320196 유지훈

차 례

제1장 서론

1.1 문제 제기

무언가 필요한 자료가 생겼을 때, 사람들은 어떻게 자료를 구할까? 도서관에서 책을 찾아보거나 해당 분야의 권위자와의 면담 혹은 현장에서 직접 취득할 수 있겠지만 인터넷서의 검색은 불가피하다. 대중적으로 잘 알려진 개념 혹은 주제라면 쉽게 원하는 정보를 얻을 수 있겠지만, 찾고자 하는 내용의 전문성이 높아질수록 정보의 취득에 난항을 겪는다. 여러 사이트에서 검색을 반복하고, 검색 결과를 일일이 훑는 일을 반복해야 한다. 또는 어떤 분야에 대해 최근 동향을 살펴보고자 한다면, 일정 기간의 방대한 내용을 살펴보아야 하는데, 이 또한 많은 시간을 공들여야 하는 부분이다.

자료조사는 수행하려는 일의 시작부분임에도 불구하고 많은 시간과 노력이 들어간다. 자료를 조사하는 수행 시간을 단축하고, 수행 과정을 단순화하는 것은 전반적인 일의 완전성을 높인다. 이번 개인프로젝트 과제의 조건인 프로그래밍 언어 java로 웹 크롤링 구현하고 이를 응용하여 다목적으로 이용될 수 있는 프로그램을 구현한다.

제2장 구상

2.1 활용 구상

웹 크롤러를 이용한 이 프로그램은 두 가지 이용목적에 부합한다.

첫째, 신뢰할 수 있고, 알고자 하는 내용이 담긴 자료의 위치를 알아낸다.

둘째, 특정 내용에 대한 최근 동향 혹은 대략적인 내용을 파악한다.

사용자는 이 프로그램을 이용하면서 첫째, 원하는 결과를 얻을 수 있어야 하고, 이용이 간편해야 한다. 사용자가 프로그램에 원하는 검색어를 입력하면 프로그램은 웹 페이지 이름 혹은 게시물 제목, 날짜, 내용 등의 데이터를 추출한다. 검색을 수행하기 전에 검색 하고자 하는 사이트의 유형, 기간 등의 선택지를 만들어 활용성을 높인다.

대략적인 파악 혹은 최근 동향 등의 목적으로 자료 조사를 실시하는데 방대한 텍스트를 다루어야 하는 경우, 텍스트를 읽어내고 빈출 되는 핵심 단어들을 추려 사용자의 원하는 내용에 대한 이해를 돕는 기능을 넣도록 한다.

2.2 프로그램 구상

프로그램은 검색 기능과 요약 기능에 있어서 여러가지 옵션을 추가하도록 하고, 추출한 데이터를 관리할 수 있는 기능을 지니도록 구상한다. 이 프로그램에서는 다음의 여러 기술을 이용하고자 한다.

1) 웹 크롤링

원하는 사이트, 혹은 여러 사이트에서 검색 항목이 담긴 내용을 추출하기 위해 웹 크롤링을 이용한다. 사용자는 검색 전, 여러 옵션을 통해 추출하고자 하는 내용의 범위와 크기를 조절할 수 있도록 하고, 프로그램은 이 작업들을 원활히 수행할 수 있어야 한다.

2) 데이터 관리

크롤링을 통해 추출한 데이터를 사용자가 삭제 혹은 이동 등의 편집을 자유로이 할 수 있도록 데이터베이스를 구축한다. 데이터를 추출하였으면 사용자가 쉽게 파악할 수 있도록 데이터에 대한 가공이 이루어져야 하며, 가공된 데이터의 관리 및 처리를 수행하여 사용자의 업무 수행의 효율성을 높인다.

제3장 계획 및 실행

각 주차 별 계획은 다음과 같다.

기간	내용
1주차 10/28 ~ 11/3	- 프로그램의 구조 구상 - Java를 이용한 웹 크롤링 구현
2주차 11/4 ~ 11/10	- 웹 크롤링 구현 - 사용자가 원하는 조건을 만족하는 웹 크롤링 설계 및 구현
3주차 11/11 ~ 11/17	- 추출한 데이터의 활용 및 관리 방안 마련 - 데이터 관리 프로그램 구현
4주차 11/18 ~ 11/ 24	- 프로그램 구조 재점검 - 사용자 인터페이스 구상
5주차	- 웹 크롤링 및 데이터 관리 기능 탑재

11/25 ~ 12/1	- 크롤링 성능 테스트
6주차 12/2 ~ 12/8	- 자료 검색의 정확성 개선 - 데이터 관리 향상 방안 검토
7주차 12/9 ~ 12/15	- 테스트 및 디버깅 - 결과물 산출

각 주차 별 실행은 다음과 같다.

기간	**내용**
1주차 10/28 ~ 11/3	- Java를 이용한 웹 크롤러 탐색
2주차 11/4 ~ 11/10	- JSON을 이용한 크롤러 구상 및 구현
3주차 11/11 ~ 11/17	- 기존의 Json에서 Jsoup로 변경 - Jsoup를 이용한 크롤러 프로토타입 구현
4주차 11/18 ~ 11/ 24	- Jsoup를 이용한 크롤러 프로토타입 구현
5주차 11/25 ~ 12/1	- 웹 파싱 점검 및 계획 대폭 수정 - 크롤러 프로그램 구현

6주차 12/2 ~ 12/8	- 크롤러 프로그램 구현 - 테스트 및 디버깅
7주차 12/9 ~ 12/15	- 테스트 및 디버깅 - 결과물 산출 및 커밋

제4장 프로젝트 결과

JSoup를 이용한 논문검색사이트의 검색 결과를 파싱하는 프로그램을 만들었다. 세개의 소스코드로 이루어진 이 프로그램은 URL에 접속하여 특정 데이터를 파싱하는 SearchSite.java, 파싱한 데이터를 텍스트 파일로 관리하는 FileManage.java, 프로그램 실행에 관한 Main.java가 그 요소이다.

SearchSite에는 논문 검색을 위해 이용하는 주요 사이트에 접속하여 사용자가 원하는 검색 키워드의 검색 결과를 파싱한다. 이 과정에서 Jsoup를 이용한 파싱이 이용되었다. Jsoup로 연결해서 얻어온 HTML 전체 문서를 Document 객체에 넣고, select 메소드를 이용하여 사용자에게 필요한 정보가 담긴 HTML의 요소를 탐색한다. 탐색한 정보는 PrintStream을 이용하여 바탕화면의 SearchResult 폴더에 검색 키워드를 이름으로 같은 텍스트 파일로 출력한다. 이로써 이용자는 프로그램을 통해 원하는 키워드를 검색한 후 그 결과를 텍스트 파일에서 한데 모아 살펴보고 원하는 논문을 URL을 통해 접속하여 확인 할 수 있다.

본 계획에서는 프로그램에서 검색을 위한 옵션을 선택하고, 뿐만 아니라 논문의 텍스트 분석을 통한 이용자의 자료수집 능력에 향상을 기여하는 것이 목표였다. 다만 검색 키워

드의 결과를 나타내는 웹의 접속과 처음 접하는 HTML 문서로부터 원하는 정보를 가져오는 과정에 다수의 시간이 소요되었다. 더불어 쿠키데이터를 이용한 웹 접속으로 양질의 검색결과를 가져오고자 시도하였으나, 해결하지 못한 버그의 발생으로 결국 프로그램에서 해당 소스코드를 다수 제거할 수밖에 없었다. 시간관리의 부족으로 텍스트 분석 기능은 탑재하지 못하였으며, 검색기능의 향상의 가능성이 전무한 프로젝트 결과를 낳았다.

1) 깃허브 링크

https://github.com/Upota/JavaProject

2) 동장 데모 영상 링크 주소

https://youtu.be/4G5NPtTdnPs

제5장 회고

크롤러에 대한 흥미를 바탕으로 직접 프로젝트로 시작하기에 이르렀지만, 그 과정은 생각보다 난해하였다. 가장 먼저 Json을 이용한 파싱 프로그램을 만들고자 목표했지만, 프로그램에서 웹에 접속하다는 행위가 일체 처음인지라 굉장한 난항을 겪었다. 인터넷에서 자바를 이용한 웹 크롤러를 살펴보던 중 Jsoup를 이용한 소스코드를 볼 수 있었고, 이에 Jsoup를 채택하여 프로젝트를 다시 구상하기로 하였다.

프로그램을 설계하는 과정에서 첫 어려움은 원하는 정보를 텍스트로 어떻게 가져올 것인지였다. Jsoup는 HTML 문서의 전체를 한번에 가져오고, 원하는 요소를 선택하면 된다는 점에서 큰 이점을 얻었지만, 그럼에도 불구하고 처음 접하는 HTML문서를 어떻게

처리해야 정보를 가공할 수 있을지 고민이었다. 먼저 HTML문서가 작성되는 방법을 알아야 했고, 그와 관련한 문서들을 먼저 공부한 후 프로젝트에 진척을 이룰 수 있었다.

논문검색을 위한 사이트를 선정하고 각 사이트에 알맞은 파싱과정을 각각 구현하던 중, 사이트에서 요구하는 헤더와 쿠키 데이터를 제공해야 파싱이 가능 한 곳들이 있었다. 이를 해결하기 위해 헤더와 쿠키 데이터를 다루는 것을 찾아보고 프로젝트에 적용하여 했으나, 각 사이트에서 무엇을 요구하고 어떤 정보를 넘겨야 원하는 동작이 이루어지는지 파악하는 것에 어려움을 겪었다. 그래서 논문 검색을 위한 사이트의 수를 대거 축소하였고, 양질의 검색 결과를 얻는 데에 있어서 만족스럽지 못한 결과다.

분명 어려움과 실수가 다분했던 프로젝트였다. 다만 부족한 것이 많았기에 더 보충하고자 하는 것들이 있다. 먼저 쿠키와 헤더를 능숙히 이용하는 것은 물론이고 각 사이트에서 제공하는 고급검색의 기능을 프로그램을 통해 사용하는 것이다. 더불어 검색 결과를 텍스트로 저장하는 과정에서 URL에 하이퍼링크를 적용한다면 사용자의 편의가 개선될 것이다. 또한, 검색결과를 출력한 파일들을 관리할 수 있는 기능을 추가하는 것이 추후 이루어질 개선사항이다. 가장 아쉬우면서도 추후에 탑재하고자 하는 기능은 방대한 텍스트 자료로부터 문맥파악을 시도하는 기능이다. 이를 통해 자료수집에 대한 사용자의 수고를 덜고, 사용자가 목표한 일의 만족스러운 성과를 달성하도록 돕는 것이 이 프로그램의 목표이다.

참고자료

[1] 나동빈, (2016, Jun). "Java 프로그래밍을 이용한 사이트의 서버시간 웹 파싱,"
https://www.youtube.com/watch?v=OuPjoiXq9gg [October, 2018]

[2] “jsoup Java HTML Parser 1.11.3 API,”

https://jsoup.org/apidocs/overview-summary.html , [November, 2018]

[3] (2017, May). “Java HTML parser, Jsoup 로 원하는 값 얻어내기 - 기본”

http://partnerjun.tistory.com/42?category=693285, [November, 2018]

[4] 진성소프트, (2016, Nov). “[Android] Jsoup 을 이용한 영어 단어 정보 Parsing,”

http://jinseongsoft.tistory.com/60 [November, 2018]

[5] “jsoup : 자바 HTML 파서(Java HTML Parser)”

http://iwbtbitj.tistory.com/86, [November, 2018]

a. 가능하면 IEEE 양식 사용