데이터셋 생성을 위한 이미지 URI 및 메타데이터 수집 크롤러

박준홍*, 김석진*, 정연욱*, 이동욱*, 정영주*, 서동만*
*대구가톨릭대학교 IT 공학부
e-mail : sarum@cu.ac.kr

For creating a Dataset Image URI and Metadata Collection Web Crawler

June-Hong Park*, Seok-Jin Kim*, Yeon-Uk Jung*, Dong-Uk Lee*, YoungJu Jeong*, Dong-Mahn Seo*
*School of Information Technology Engineering, Daegu Catholic University

요 약

인공지능 학습에 대한 관심이 증가하면서 학습에 필요한 데이터셋 구축에 필요한 많은 양의 데이터가 필요하다. 데이터셋 구축에 필요한 데이터들을 효과적으로 수집하기 위한 키워드 기반 웹 크롤러를 제안한다. 구글 검색 API 를 기반으로 웹 크롤러를 설계하였으며 사용자가 입력한 키워드를 바탕으로 이미지의 URI 와 메타데이터를 지속적으로 수집하는 크롤러이다. 수집한 URI 와 메타데이터는 데이터베이스를 통해 관리한다. 향후 다른 검색 API 에서도 동작하고 다중 쓰레드를 활용하여 크롤링하는 속도를 높일 예정이다.

1. 서론

빅데이터 기술의 발달과 딥러닝 기반 인공지능 학습의 개선으로 인하여 인공지능 학습이 재조명되고 있다. 이와 같은 인공지능 학습을 진행하기 위해서는 표본 데이터셋이 필요하다

인공지능 학습을 위한 데이터셋을 구축하기 위한 방법으로 사용자가 데이터를 직접 수집하는 방법과 크롤러를 통해 이미지를 수집하는 방법이 있다. 사용자가 직접 데이터를 수집하는 방법은 프로그램을 통해 자동으로 이미지 수집을 진행하는 방법에 비해서 시간, 인적 자원의 낭비가 심하다.

이러한 낭비를 줄이기 위한 방법으로 데이터를 자동화된 방법으로 수집하는 프로그램인 크롤러가 있다. 크롤러란 특정 문서 또는 정보를 수집하는 것이다. 그 중 웹 크롤러는 인터넷 상에 존재하는 문서들을 조직적, 자동화된 방법으로 탐색하는 프로그램이다[1].

본 논문에서는 인공지능 학습을 위한 데이터셋을 자동으로 구축하는 크롤러를 제안한다. 데이터셋을 자동으로 구축하기 위해 제안하는 크롤러는 사용자로부터 입력받은 키워드를 바탕으로 지속적으로 이미지 URI 를 수집한다. 수집한 이미지 URI 와 메타데이터는 데이터베이스를 통하여 관리하므로 데이터셋 구축에 필요한 데이터를 키워드를 통한 요청으로 가져올 수 있다.

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1H1A1079764).

2. 에이전트 개발도구의 요구사항

2.1 Python

제안하는 크롤러는 파이썬(Python)을 사용하여 구현하였다. 파이썬은 간결한 문법으로 구성되어 있고, 사용자에게 다양한 라이브러리를 제공한다[2]. 웹 크롤러를 개발할 때 Selenium, BeautifulSoup, Requests 등의 라이브러리들을 제공하기 때문에 Python 을 사용하여 웹 크롤러 개발을 진행하였다.

2.2 BeautifulSoup

BeautifulSoup 를 사용하여 웹페이지 HTML 코드에서 원본 이미지 URI 수집을 진행하였다. BeautifulSoup 는 HTML, XML 문서의 태그에 대한 파싱을 진행할 때 유용한 HTML parser 이다. 구현한 웹 크롤러에서는 검색 페이지 내 이미지의 원본 URI 와 해당하는 이미지의 메타데이터를 파싱하기 위해 사용하였다[3].

2.3 Requests

수집한 이미지 URI 의 유효성을 검증하기 위해 Requests 라이브러리 내 get 메소드를 사용하여 해당 URI 의 실제 URI 를 확인하였다. 이후 urllib 의 urlopen 함수를 통해 해당 이미지의 실제 URI 를 응답받아 수집, 업로드 날짜와 함께 데이터베이스에 저장하였다[4].

3. 본론

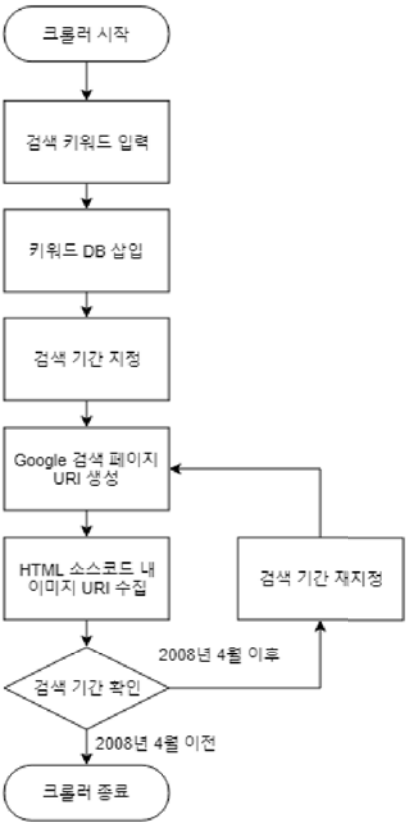

그림 1. 이미지 크롤러 동작 순서도

제안하는 웹 크롤러는 구글 검색 API[5]를 기반으로 사용자로부터 입력받은 키워드를 통해 검색 페이지 내 이미지들을 지속적으로 크롤링하는 것을 목표로 한다. 웹 크롤러의 동작 순서는 그림 1 과 같다.

크롤러는 사용자가 입력한 키워드를 바탕으로 구글 검색 API 으로부터 구글 검색 페이지의 HTML 코드를 전달받는다. 키워드 입력은 웹 크롤러 내의 코드 수정을 통해 입력한다.

전달받은 HTML 코드 내에서 BeautifulSoup 를 이용하여 페이지 내 모든 이미지의 원본 URI, 메타데이터 정보를 리스트에 저장한다. 이미지의 원본 URI 과 이미지 메타데이터 정보는 HTML 코드 내 json 형태로 명시된 값을 파싱하여 저장한다. 이후 리스트에 저장된 이미지 URI 의 유효성 검사를 진행한다. 이미지의 유효성 검사는 해당 이미지 URI 가 정상적으로 동작하는지 확인하는 과정이다. Requests.get 을 통해 이미지 URI 의 응답을 얻은 후 urlopen 으로 이미지의 실제 URI 를 요청한다. timeout 옵션을 사용하여 5 초간 응답을 대기하며, 5 초가 지나도 응답이 없는 경우 해당 이미지는 삭제된 것으로 간주한다. urlopen 을 사용하여 정상적으로 응답을 받는다면 geturl() 함수를 사용하여 이미지의 실제 URI 를 전달받는다. 하나의 이미지에 대해 유효성 검사가 끝나면 해당 이미지 URI 와 메타데이터를 데이터베이스에 삽입하는 과정을 진행한다. 데이터베이스 내 저장된 동일한 URI 값 입력을 제한하기 위해 동일한 URI 가 존재할 시 제외하여 중복된

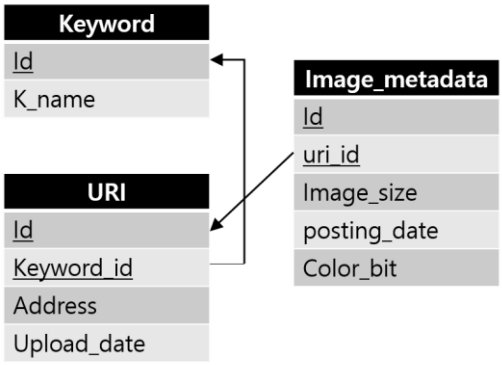

그림 2. 데이터베이스 E-R 다이어그램

그림 3. 생성된 json 데이터셋의 일부

이미지가 저장되지 않도록 한다. 입력하려는 URI 가 이미 데이터베이스 내에 존재한다면, 해당 URI 는 입력하지 않는다. 검색 페이지 HTML 코드에서 모든 이미지 URI 를 저장하였다면, 검색 기간을 재조정한 뒤, 위 과정을 반복한다.

수집한 데이터는 그림 2 와 같은 구조의 데이터베이스를 통해 관리한다. 데이터베이스는 Keyword, URI, Image_metadata 3 개의 테이블로 구성된다. 각 테이블에는 이미지 URI, 이미지 업로드 날짜, 데이터베이스 업로드 날짜, 이미지 메타데이터 정보, 이미지 검색 키워드를 저장한다. 데이터베이스는 수집한 이미지의 중복을 방지하는 역할도 수행한다. 데이터베이스에 저장된 정보들을 json 형태로 Export 한 경우 그림 3 과 같은 형태가 된다. 데이터베이스에 이미지 URI 와 메타데이터를 저장하는 과정을 마치면 원하는 키워드의 이미지를 웹 페이지 검색보다 빠르게 출력할 수 있다. 또, 이미지 URI 가 필요한 다른 시스템의 서브셋으로도 사용할 수 있다.

5. 결론 및 향후 계획

본 논문에서는 이미지 파일의 URI 와 메타데이터를 수집한 뒤 검색 키워드를 기준으로 데이터베이스에 저장하는 웹 크롤러를 구현하였다. 해당 이미지 크롤러를 사용하여 검색 키워드에 대응하는 이미지들을 수집할 수 있었으며, 시간당 약 1500 개의 이미지 URI

를 수집할 수 있었다.

그러나 한 번의 크롤링에서 단일 키워드에 해당하는 이미지를 대상으로 크롤링을 진행하는 제한이 존재하고, 하나의 쓰레드만을 사용하여 크롤링을 진행하기 때문에 이미지 URI 를 수집하는 속도에도 한계가 있다. 그리고 구글 검색 API 의 한계로 인해 2008 년 4 월 이전의 이미지는 수집이 불가능하다는 문제점이 존재한다. 또한 웹 상에 존재하는 원본 이미지 파일이 삭제되어 이미지를 출력할 수 없는 URI 를 데이터베이스에 삭제하는 기능 구현도 필요하다.

향후 연구에서는 크롤러가 구글 검색 API 에 기반하여 발생하는 문제점을 해결하기 위해 크롤러를 구글 검색 외 다른 검색엔진에서도 동작할 수 있도록 수정할 계획이다. 또한 사용자가 초기에 입력한 키워드와 유사한 키워드 또는 리스트에 존재하는 다른 키워드를 가져와 크롤링을 재시작하는 기능과 다중 쓰레드를 사용하여 이미지 URI 수집 속도를 높인 크롤러를 구현할 예정이다.

참고문헌

[1] 김광영, 이원구, 이민호, "웹 자원 아카이빙을 위한 웹 크롤러 연구 개발", 한국콘텐츠학회논문지, 제 11 권, 제 9 호, pp.9-16, 2011.

[2] Guido van Rossum. Python Reference Manual http://www.python.org/doc/current/ref/ ref.html.

[3] Richardson, L., Beautiful Soup. http://www.crummy.com/software/BeautifulSoup

[4] Python Requests https://2.python-requests.org/en/master/

[5] Google Custom Search. https://developers.google.com/custom-search