일반논문 (Regular Paper)
방송공학회논문지 제27권 제4호, 2022년 7월 (JBE Vol.27, No.4, July 2022)
https://doi.org/10.5909/JBE.2022.27.4.581
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

메타데이터 기반 순위 알고리즘을 활용한 데이터셋 검색 시스템

최 우 영[a)], 전 종 훈[a)‡]

Dataset Search System Using Metadata-Based Ranking Algorithm

Wooyoung Choi[a)] and Jonghoon Chun[a)‡]

요 약

최근 빅데이터 활용에 대한 요구사항이 증대됨에 따라 데이터 분석에 필요한 데이터셋 검색 기술에 관한 관심 또한 늘어나고 있다. 데이터셋 검색을 위해서는 일반 문서 검색과는 달리 데이터셋에 대한 메타데이터에 대한 활용도를 높여야 함에도 불구하고 이를 적극적으로 활용하는 검색 시스템에 관한 연구는 미미한 실정이다. 본 논문에서는 데이터셋의 메타데이터를 색인하고 이를 기반으로 데이터셋 검색을 수행하는 새로운 데이터셋 전용 검색 시스템을 제안한다. 데이터셋 검색결과에 부여하는 순위는 데이터셋 고유의 특성을 반영한 알고리즘을 새로이 고안하여 적용하며, 분석에 필요한 융합 가능한 데이터셋 여러 건을 한꺼번에 검색할 수 있도록 원천 질의에 의해 검색된 데이터셋과 연관 관계에 있는 추가 데이터셋을 검색하는 기능을 제공한다.

Abstract

Recently, as the requirements for using big data have increased, interest in dataset search technology needed for data analysis is also growing. Although it is necessary to proactively utilize metadata, unlike conventional text search, research on such dataset search systems has not been actively carried out. In this paper, we propose a new dataset-tailored search system that indexes metadata of datasets and performs dataset search based on metadata indices. The ranking given to the dataset search results from a newly devised algorithm that reflects the unique characteristics of the dataset. The system provides the capability to search for additional datasets which correlate with the dataset searched by the user-submitted query so that multiple datasets needed for analysis can be found at once.

Keyword : Dataset, Search, Metadata, Ranking, Big data

a) 명지대학교 ICT 융합대학 융합소프트웨어학부(School of Software Convergence, College of ICT Convergence, Myongji University)
‡ Corresponding Author : 전종훈(Jonghoon Chun)
E-mail: jchun@mju.ac.kr
Tel: +82-2-300-0647
ORCID:https://orcid.org/0000-0003-3396-4239
· Manuscript July 5, 2022; Revised July 18, 2022; Accepted July 18, 2022.

Ⅰ. 데이터셋 검색 시스템의 필요성

최근 빅데이터 활용에 대한 요구사항이 증대됨에 따라 데이터 분석에 필요한 데이터셋 검색 기술에 대한 관심 또한 늘어나고 있다. 데이터셋은 필연적으로 데이터셋에 대한 부가적인 설명을 담고 있는 메타데이터와 쌍으로 존재하며, 데이터셋의 검색을 위해서는 메타데이터를 색인하여 이를 활용하여 검색 시스템을 구축하는 것이 일반적이다.

데이터셋에 대한 메타데이터는 DCAT[1]과 같은 표준이 존재하며 표준기관에서 공표한 표준은 아니지만 웹상에 게시되어 있는 데이터셋 메타데이터의 실질적인 표준처럼 사용되고 있는 Schema.org[2]가 있다. 둘 다 공통으로 제목(title), 설명(description), 키워드(keyword), url(uniform resource locator), 수정 일자 등을 포함하는 다수의 항목으로 이루어져 있다. 따라서 각각의 항목에 포함된 텍스트를 인덱싱하여 이를 검색에 활용하는 것이 실제 데이터셋 자체를 인덱싱하는 것보다 검색 측면에서 훨씬 합리적인 선택임을 쉽게 알 수 있다. 다만, 현업에서 실질적으로 사용되는 메타데이터 항목들은 데이터셋의 제목, 설명, 키워드 등의 극소수에 불과하며 많은 다른 메타데이터 항목들은 잘 사용되지 않거나 검색에 기여도가 높지 않은 내용으로 채워지는 경우가 많다. 따라서 우리는 메타데이터에 포함되는 항목 중 제목, 설명, 키워드 등의 몇몇 항목이 기타 다른 버전, url 등의 항목에 비해서 상대적으로 중요도가 높고 검색에 기여하는 정도가 더 높을 수 있다고 가정하고, 각 메타데이터 항목별로 각기 다른 가중치를 부여하여 검색 질의 요구사항에 부합하는 정도를 계산할 수 있는 새로운 방식의 알고리즘이 필요하다는 점에 주목한다.

또한, 데이터셋 검색은 데이터베이스 질의 환경과는 상이한 것이, 분석에 필요한 융합 가능한 여러 데이터셋을 한꺼번에 찾고자 하는 경우가 많으므로 원천 질의와 유사하면서 초기에 검색된 데이터셋과 연관 관계에 있는 유사 데이터셋을 추가로 검색하고자 하는 요구사항이 있다. 이를 반영하기 위해서는 품질이 좋고, 최근에 갱신되었으며, 다른 사용자들이 많이 사용했던 데이터셋에 가점을 부여하고, 원천 질의와의 연관성 여부도 고려하여 초기 데이터셋과 융합 가능한 데이터셋의 연관성 점수를 계산하고 이를 순위값으로 반환할 수 있는 새로운 방식의 연관성 점수 계산 방식이 추가로 필요하게 된다.

기존의 문서 검색엔진들이 메타데이터 항목이나 특성을 고려하지 않고, 단순히 본문 텍스트에 출현하는 단어 빈도수 등만을 기반으로 색인을 만들어서 검색을 수행하는 방식과는 근본적으로 다른 데이터셋 검색에 특화된 새로운 방식의 검색 기술이 필요한 것이다.

따라서, 본 논문에서는 종래 문서 검색에 최적화되어 있는 일반적인 검색엔진에서 사용하는 연관성 점수 계산 방식을 지양하고, 데이터셋에 대한 설명정보를 포함하고 있는 메타데이터의 특성과 항목별 중요도를 고려하고 고품질 데이터셋에 높은 점수를 부여하는 방식으로 데이터셋을 검색하여 이를 기반으로 연관성 점수를 계산하고 내림차순으로 순위를 제공할 수 있는 새로운 알고리즘을 제안한다.

본 논문에서 제안한 방식은, 고유한 식별자 및 이와 일대일로 매핑되는 다수의 데이터셋과 각각의 데이터셋과 매칭되는 메타데이터를 포함하여 저장하여 관리하는 데이터베이스가 존재하는 환경에서 각각의 사용자가 필요로 하는 데이터셋을 검색하는 방법에 있어서, 첫째 사용자에 의해 제출된 질의에 부합하는 데이터셋을 검색하기 위해 사용하는 주어진 질의와 데이터셋의 연관성 점수를 계산하여 순위값으로 반환하는 단계, 둘째 사용자가 선택한 특정 데이터셋과 융합 가능한 후속 데이터셋을 순위값을 기반으로 계산하여 추가로 제공하는 단계를 포함한 메타데이터 기반 연관 데이터셋 검색 방법에 관한 것으로 주어진 질의에 대한 데이터셋의 연관성 점수 계산 방법과 선택된 특정 데이터셋과 나머지 데이터셋들간의 연관성 점수 계산 방법을 포함한다.

본 논문의 구성은 다음과 같다. 2장에서는 데이터셋 검색의 배경 및 관련 연구를 조사 분석하고 3장에서는 메타데이터의 특성을 고려한 새로운 메타데이터 기반 순위값 계산 방식을 제안한다. 4장에서는 프로토타입 구현 환경과 결과를 제시하고 5장에서는 결론과 향후 연구과제를 제시한다.

Ⅱ. 데이터셋 검색의 배경 및 관련 현황

데이터셋 검색에 대한 문제를 다루기 위해서는 먼저 데

이터셋의 정의가 필요하다. 특정 커뮤니티에 따른 여러 다양한 정의가 존재하나, 우리는 “데이터셋은 특정 목적을 위해 조직되고 형식화된 수집된 관찰결과의 모음”이라고 정의한 [3]의 정의를 차용하여 사용하기로 한다.

데이터셋은 이미지, 동영상, 그래프 또는 문서일 수 있으며 전통적인 테이블 데이터일 수도 있다. 그러므로 데이터셋 검색은 데이터셋의 발견, 탐색 및 최종 사용자에게 데이터셋을 찾아서 반환하는 프로세스를 포함한다. 실제 데이터셋이 이미지 또는 그래프 등을 포함하고 있음에도 불구하고 최종 사용자가 본인이 원하는 데이터셋에 대한 질의의 표현을 텍스트로 하는 한, 데이터셋 검색에서 우리가 집중해야 할 부분은 텍스트에 대한 인덱싱과 이에 기반한 검색이다. 이는 각 커뮤니티별로 구축하여 운영하는 데이터셋 저장소(예, Figshare[4], Dataverse[5], Elsevier Data Search[6])나 오픈 데이터 포털(예, data.gov[7], data.go.kr[8], kaggle[9], data.europa.eu[10]), 구글 데이터셋 검색엔진[11]에서도 확인할 수 있는 일반적인 패턴이기도 하다.

데이터셋 검색은 크게 사용자가 제출한 질의를 처리하는 단계, 데이터셋을 생산하는 생산자가 데이터셋에 대한 메타데이터를 기술하고 데이터셋을 특정 데이터 저장소나 포털에 게재하는 단계, 데이터셋 검색엔진에 의해 반환되는 검색결과를 표출하는 단계를 포함한다. 질의 처리 단계에서는 사용자가 제출한 질의로 데이터셋에 대하여 게시된 메타데이터를 검색하는 데 사용되며, 제출한 질의에 포함된 질의어가 메타데이터와 얼마나 유사한지에 따라 결과가 생성된다. 데이터셋 생산자가 데이터 저장소나 포털에 데이터셋을 게재하는 단계에서는 데이터셋에 대한 제목, 설명, 언어, 시간이나 공간적 적용 범위 등을 포함한 데이터셋에 대한 메타데이터를 기술한다. 이 단계는 대부분 수동으로 이루어지는 노동 집약적인 성격을 띠며, 따라서 대부분은 데이터셋에 대한 설명이 불완전하거나 세부적인 내용을 충분히 포함하지 않는다는 문제점이 존재한다. 빈약한 메타데이터는 검색어의 출현 빈도를 직접 사용하여 질의를 처리하는 방식의 전통적인 질의 처리 알고리즘에서의 성능을 저하하는 요인으로 작용한다. 데이터셋 검색엔진에 의해 반환되는 결과는 다양한 방식으로 표출될 수 있으며, 검색결과는 보편적으로 메타데이터로 이루어진 리스트 형식으로 반환되거나, 문장 형식의 요약본이나 데이터셋의 부분 샘플, 혹은 결과 데이터셋을 다양한 방식으로 시각화하기도 한다. 다만, 일반적으로 데이터 분석가나 데이터 과학자가 분석을 위해서 단건의 데이터셋을 검색하는 것이 아니라 융합 가능한 여러 데이터셋을 한꺼번에 검색하거나 추가로 검색해야 하는 요구사항을 고려할 때, 기존의 텍스트 검색엔진의 결과 표출 방식을 그대로 답습하는 것은 개선이 필요한 부분이다.

데이터셋은 이미지, 동영상, 그래프 또는 문서일 수 있으며 전통적인 관계형 데이터베이스의 테이블 데이터의 형태로 존재할 수 있다. 그러므로 현재 사용되고 있는 데이터셋 검색 시스템은 다양한 형태로 존재한다. 개방형 데이터 포털[12,13,14,15]들은 가용한 데이터셋의 메타데이터에 대한 검색 시스템을 사용자들에게 포털 웹사이트를 통해 개방한다. 개방 데이터 포털에서 가장 많이 사용되고 있는 포털 소프트웨어는 CKAN[16]이며, CKAN은 Lucene[17]을 통해 인덱싱하는 Apache Solr[18]를 검색엔진으로 쓴다. 구글은 2018년도에 데이터셋 검색을 위한 전용 검색엔진을 발표했으며, 구글 데이터셋 검색엔진은 Schema.org와 DCAT을 메타데이터 기술을 위한 체계로 수용하여 이에 대한 인덱싱과 검색이 가능하도록 구현되어있다. 데이터셋 검색에 관한 대표적인 연구로는 [19]가 있으며, 이 연구에서는 웹과 같은 대규모의 분산된 환경에서 데이터 분석가나 과학자가 원하는 데이터셋을 수평적, 수직적으로 융합 가능한지 여부를 확인할 수 있는 유사도 측정방식을 제안하고 이를 활용하여 데이터셋 검색을 수행할 수 있는 이론을 제시하고 이의 확장성과 효용성을 실험을 통하여 입증하였다.

Ⅲ. 메타데이터 기반 순위값 계산 방식

3장에서는 메타데이터를 활용하여 데이터셋을 검색하기 위한 검색 시스템에서 적용 가능한 새로운 순위값 계산 방식을 제안한다. 제안하는 순위값 계산 방식은 데이터셋 메타데이터 항목의 특성을 고려하여 차별화된 가중치를 부여하는 방식으로 순위값을 산출함으로써, 기존의 문서 검색엔진에서 메타데이터 항목이나 특성을 고려하지 않고 본문 텍스트에 출현하는 단어 빈도수 등만을 기

반으로 검색을 수행하는 방식과는 근본적으로 다른 접근 방식을 취한다.

<그림 1>은 데이터셋, <그림 2>는 매칭되는 데이터셋을 설명하는 메타데이터의 전형적인 예이다. 데이터셋에 대한 메타데이터는 제목, 설명, 키워드, url, 수정 일자 등의 항목으로 이루어져 있으며, 각각의 항목에 포함된 텍스트를 인덱싱하여 이를 검색에 활용하는 것이 실제 데이터셋 자체를 인덱싱하는 것보다 검색 측면에서 훨씬 합리적인 선택임을 쉽게 알 수 있다. 예를 들어, <그림 1>의 코로나 바이러스 데이터셋의 CountyFIPS 컬럼의 '20'이라는 값을 인덱싱하는 것은 사용자가 미국의 일일 코로나바이러스 데이터셋을 검색 시스템에서 검색하여 찾는 데 도움이 크게 되지 않는다. 그러나 <그림 2>의 미국 일일 코로나 테스트 데이터셋(COVID test dataset)에 대한 메타데이터에서 설명(description) 컬럼을 인덱싱하면 이 데이터셋이 미국의 일일 코로나 테스트 데이터셋이라는 것을 알 수 있으므로 이 정보를 활용하여 검색이 가능해질 수 있다. 따라서 데이터셋 검색에서는 데이터셋 그 자체를 인덱싱하는 것도 의미가 있을 수 있지만, 데이터셋을 이용하여 분석 등의 작업을 수행하고자 하는 사용자의 관점에서 필요한 메타데이터를 인덱싱하는 것이 더 의미가 있을 수 있다 하겠다. 데이터셋 검색 관련 연구는 현재 활발히 진행되고 있으며, 대부분의 연구가 데이터셋 자체보다는 메타데이터를 인덱싱하려는 시도가 많다는 점[20,21,22,23]에서 본 논문에서 제안하는 순위값 계산 방식과 유사성을 띤다.

앞서 언급한 대로 메타데이터는 표준이 존재하며, 어떤 메타데이터 표준을 준용하는지와 상관없이 제목, 설명, 키워드 등을 포함하여 적게는 수개에서 수백 수천 개까지의 항목을 포함한다. 예를 들어, 메타데이터의 표준 중의 하나인 DCAT의 경우에는 91개, 표준은 아니지만, 실질적으로 표준이나 다름없이 메타데이터의 기술 체계로 활용되고 있는 Schema.org의 경우에는 1,447개에 달하는 메타데이터 항목을 사용하도록 정해놓고 있다. 다만, 현업에서 실질적으로 사용되는 메타데이터 항목들은 제목, 설명, 키워드 등의 극소수에 불과하며 꽤 많은 다른 메타데이터 항목들은 잘 사용되지 않거나 검색에 기여도가 높지 않은 내용으로 채워지는 경우가 많다. 따라서 우리는 메타데이터에 포함되는 항목 중 데이터셋의 제목, 설명, 키워드 등의 몇몇 항목이 기타 다른 버전, url 등의 항목에 비해서 상대적으로 중요도가 높고 검색에 이바지하는 정도가 더 높을 수 있다

COVID_Testing

CollectedDate	CompletedDate	County	CountyFIPS	FID	Lab	Region	RegionCode	Results	the_geom
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20583	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20582	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20581	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20580	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20579	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20578	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20577	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20576	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20575	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20574	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20573	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20572	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20571	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20570	ASPHL	Anchorage	7	Negative	
06/07/2020 12:00:00 PM	06/09/2020 12:00:00 PM	Anchorage Municipality	20	20569	ASPHL	Anchorage	7	Negative	

그림 1. 미국의 일일 코로나 테스트 데이터셋
Fig. 1. Daily COVID Test Dataset USA

```
{
  "@type": "dcat:Dataset",
  "accessLevel": "public",
  "contactPoint": {
    "@type": "vcard:Contact",
    "fn": "Mark Meyer",
    "hasEmail": "mailto:no-reply@data.muni.org"
  },
  "description": "A public data set of daily COVID testing data. Managed by the State of Alaska Department of Health and Social Services.",
  "distribution": [
    {
      "@type": "dcat:Distribution",
      "downloadURL": "https://data.muni.org/api/views/hndz-yyym/rows.csv?accessType=DOWNLOAD",
      "mediaType": "text/csv"
    },
    {
      "@type": "dcat:Distribution",
      "describedBy": "https://data.muni.org/api/views/hndz-yyym/columns.rdf",
      "describedByType": "application/rdf+xml",
      "downloadURL": "https://data.muni.org/api/views/hndz-yyym/rows.rdf?accessType=DOWNLOAD",
      "mediaType": "application/rdf+xml"
    },
    {
      "@type": "dcat:Distribution",
      "describedBy": "https://data.muni.org/api/views/hndz-yyym/columns.json",
      "describedByType": "application/json",
      "downloadURL": "https://data.muni.org/api/views/hndz-yyym/rows.json?accessType=DOWNLOAD",
      "mediaType": "application/json"
    },
    {
      "@type": "dcat:Distribution",
      "describedBy": "https://data.muni.org/api/views/hndz-yyym/columns.xml",
      "describedByType": "application/xml",
      "downloadURL": "https://data.muni.org/api/views/hndz-yyym/rows.xml?accessType=DOWNLOAD",
      "mediaType": "application/xml"
    }
  ],
  "identifier": "https://data.muni.org/api/views/hndz-yyym",
  "issued": "2020-04-10",
  "landingPage": "https://data.muni.org/d/hndz-yyym",
  "modified": "2020-08-04",
  "publisher": {
    "@type": "org:Organization",
    "name": "data.muni.org"
  },
  "theme": [
    "Public Health"
  ],
  "title": "COVID Testing"
}
```


그림 2. 미국 일일 코로나 테스트 데이터셋의 메타데이터
Fig. 2. Metadata of daily COVID test dataset USA

는 가정에 착안하여, 각 메타데이터 항목별로 각기 다른 가중치를 부여하여 순위값을 계산할 수 있는 새로운 방식을 제안한다.

예를 들어, 메타데이터 항목 중 제목 항목의 중요도가 가장 높고, 설명 항목의 중요도가 2번째로 높고, 키워드 항목의 중요도가 3번째로 높다고 가정하면 각각의 가중치를 0.5, 0.3, 0.2 순으로 부여하여 동일한 단어라도 어떤 메타데이터 항목에서 출현하는지에 따라 가중치가 차별화되어 부여받도록 하는 방식이다. 즉 중요도가 높은 메타데이터 항목에 출현하는 단어에는 높은 가중치를 추가로 부여하고 중요도가 상대적으로 떨어지는 메타데이터 항목에 출현하는 단어에는 낮은 가중치를 부여하는 방식을 채택함으로써, 본문 텍스트 전체를 대상으로 단어들의 출현 빈도수를 모두 동등하게 취급하여 인덱싱하는 일반 텍스트 검색엔진과는 차별화를 꾀한다.

$$Score(q,T) = \sum_i w_i \times Score(q,f_i) \qquad (1)$$
$$\sum_i w_i = 1$$

식(1)은 주어진 질의 q에 대한 특정 데이터셋의 메타데이터 테이블 T의 연관성 점수(relevance score)를 계산하는 식이다. 여기에서 메타데이터 테이블 T는 메타데이터 항목들을 컬럼으로 가지는 형태로 가정하며, 제목, 설명, 키워드 등 각각의 메타데이터 항목에 대응하는 컬럼은 식(1)의 필드(field) f_i에 해당한다. 메타데이터 테이블 T의 컬럼들을 각각 f_i 필드라 하고, 주어진 질의 q와 각 필드 f_i의 연관성 점수를 함수 $Score(q,f_i)$로 계산할 때, 각 필드 f_i의 중요도를 고려하여 차별화된 가중치 w_i를 부여하고 이를 각각의 $Score(q,f_i)$ 에 곱하여 합을 구함으로써 주어진 질의 q와 전체 메타데이터 테이블 T와의 연관성 점수 $Score(q,T)$를 계산한다. 이때 w_i, 즉 메타데이터 각 항목의 가중치에 대한 합은 1로 한다. 즉, 질의 q에 대한 메타데이터 T의 연관성 점수는 각 메타데이터 필드의 중요도를 고려하여 각기 다른 가중치를 부여하고 각 메타데이터 필드와 질의 q와의 연관성 점수를 곱하고 이의 총합을 구함으로써 전체 메타데이터와 질의 q와의 연관성 점수를 구하고, 이를 주어진 질의 q에 대한 각 데이터셋의 순위값으로 반환하게 되는 방식이다.

$$Score(T_i,T_j) = \alpha\sum_k w_k \times Score(f_k^{T_i}, f_k^{T_j}) + \beta\sum_k w_k \times Score(q,\ f_k^{T_j}) + \gamma(r(T_j)) + \delta(uf(T_j)) + \epsilon(Q(d_j)) \qquad (2)$$
$$\sum_k w_k = 1$$
$$\alpha + \beta + \gamma + \delta + \epsilon = 1$$

식 (2)는 주어진 메타데이터 테이블 T_i에 대한 메타데이터 테이블 T_j와의 연관성 점수를 구하는 식이다. 데이터셋 d_i의 메타데이터가 T_i이고 데이터셋 d_j의 메타데이터가 T_j일 때, 두 메타데이터가 충분히 잘 기술되었다고 가정하면, 두 메타데이터 테이블의 연관성 점수를 구하는 것은 곧 두 데이터셋 d_i, d_j의 연관성 점수를 구하는 것과 같다고 볼 수 있다. 다만 주어진 T_i에 대한 T_j의 연관성 점수를 구할 때, 단순히 두 메타데이터 테이블의 매칭 필드별 연관성 점수($\alpha\sum_k w_k \times Score(f_k^{T_i}, f_k^{T_j})$)만을 사용해서 구하는 것이 아니라, 주어진 메타데이터 테이블 T_i를 구하는데 사용된 원천 질의(original query) q와 T_j의 연관성 점수 ($\beta\sum_k w_k \times Score(q,\ f_k^{T_j})$), T_j에 매칭되는 데이터셋 d_j의 최신성(recency) 점수 $\gamma(r)$, 사용성(usage frequency) 점수 $\delta(uf)$, 품질(quality) 점수 $\epsilon(Q(d_j))$를 합하여 구한다. 이때 식(1)과 유사하게 전체 w_k들의 합은 1로 하며($\sum_k w_k = 1$), 전체 매개변수 α, β, γ, δ, ϵ의 합 역시 1로 한다 ($\alpha + \beta + \gamma + \delta + \epsilon = 1$).

$\alpha\sum_k w_k \times Sim(f_k^{T_i}, f_k^{T_j})$는 서로 다른 메타데이터 테이블 T_i와 T_j 간의 동일한 필드 f_k 쌍의 연관성 점수를 계산하여 필드의 중요도(weight)와 곱한 후, 이를 모두 합하고 매개변수 알파(α)를 곱한 값이다. $f_k^{T_i}$는 메타데이터 테이블 T_i의 k번째 필드를 의미하며, $f_k^{T_j}$는 메타데이터 테이블 T_j의 k번째 필드를 의미한다. $\beta\sum_k w_k \times Score(q,\ f_k^{T_j})$에서 q는 T_i를 검색하였을 때 사용한 원천 질의로써, 그 원천 질의 q와 메타데이터 테이블 T_j의 모든 필드와의 연관성 점수를 쌍으로 계산하고 중요도를 고려하여 설정된 필드 별 가중치를 곱하고 이를 모두 합한 후 나온 값에 매개변수 베타(β)를 곱한 값이다. 데이터셋 d_j의 최신성 점수는 d_j가 얼마나 최신의 데이터셋인지에 대한 값을 지표로 변환한 값

```
"interactionStatistic": [
        {
                "type": "InteractionCounter",
                "interactionType": "http://schema.org/CommentAction",
                "userInteractionCount": 0
        },
        {
                "type": "InteractionCounter",
                "interactionType": "http://schema.org/DownloadAction",
                "userInteractionCount": 55
        },
        {
                "type": "InteractionCounter",
                "interactionType": "http://schema.org/ViewAction",
                "userInteractionCount": 694
        },
        {
                "type": "InteractionCounter",
                "interactionType": "http://schema.org/LikeAction",
                "userInteractionCount": 9
        }
],
```


그림 3. Schema.org를 준용한 데이터셋 메타데이터 중 interactionStatistic 항목 예시
Fig. 3. A sample interactionStatistic metadata of a dataset conforming to the Schema.org standard

이다. 최신성 점수는 예를 들어 Schema.org 메타데이터 항목 중 데이터셋이 최신으로 수정된 날짜를 의미하는 $dateModified$ 항목을 기준으로 식(3)과 같은 방식으로 계산이 가능하다. 다만, $dateModified$ 항목의 값이 존재하지 않으면 $dateCreated$이나 $datePublished$로 대체를 하여 계산을 하고, 위 세 항목의 값이 모두 존재하지 않을 경우 최신성 점수는 0으로 계산하는 방식을 취할 수 있다. 사용성 점수는 식(4)와 같은 방식으로 사용자들이 데이터셋 d_j를 다운받은 횟수나, 데이터셋 d_j의 클릭수 등의 정보를 사용하여 계산할 수 있다. 예를 들어, <그림 3>과 같이 Schema.org를 사용하는 경우에는 interactionStatistic에 중첩되어 있는 댓글수, 다운로드수, 뷰수, 좋아요 개수 정보 중 일부를 사용하거나, 댓글 수와 다운로드 수를 식(4)에서의 dc (download count, 다운로드수)와 뷰수와 좋아요 개수를 vc (view count, 뷰수)로 각각 설정하고 이를 시그모이드[1] 함수에 대입하는 방식으로 구할 수 있다. 데이터셋 품질점수 $\epsilon(Q(d_j))$는 별도의 품질 함수 $Q(x)$에 의해 주어진 데이터셋 d_j의 품질을 산정하고 이를 T_i에 대한 T_j의 연관성 점수에 반영하기 위해 사용한다. 품질 함수 $Q(x)$는 예를 들어, 데이터셋의 품질을 사용자들의 품질평가 결과에 따라 별점 0점에서부터 5점으로 분류하여 부여한 값이 존재한다고 가정할 때, 평균 별점 값 rating에 $\frac{1}{5}$을 곱하여 구할 수 있다.

$$Recency = \sqrt{\frac{1}{(\text{현재연월} - dateModified + 1)}} \text{ (단위 : 일)}$$

* 만약 $dateModified$가 없으면 $dateCreated$ 혹은 $datePublished$로 대체
셋 다 없으면 $Recency = 0$

$$0 \leq Recency \leq 1 \quad (3)$$

1) 시그모이드 함수: $y = \frac{1}{1 + e^{-x}}$

식(2)가 필요한 이유는 메타데이터 테이블 T_i가 원천 질의 q에 대한 결과로써 주어졌을 경우에 한하여, 데이터셋을 추가적으로 검색하기 위한 용도로 사용하기 위함이다. 따라서, $\beta\sum_k w_k \times Score(q, f_k^{T_j})$항은 원천 질의 q와 후속 검색 결과인 d_j의 메타데이터 테이블 T_j와의 연관성 정도를 T_i와 T_j간의 연관성 점수에 반영하기 위하여 포함한 것을 알 수 있다. 주어진 질의 q에 의해 검색되어진 데이터셋을 d_i, 메타데이터 테이블을 T_i라 하고, d_i와 연관성있는 데이터셋 d_j, 메타데이터 테이블 T_j를 추가적으로 검색하고자 하는 환경이라고 하면, T_i와 T_j의 연관성 점수는 식 (2)에 의해서 총 5개의 항의 합으로 계산할 수 있다.

$$Usage\ Frequency = \frac{1}{1 + e^{-\frac{vc + dc}{10000}}} - \frac{1}{2} \qquad (4)$$

간략한 예를 들기 위하여, 메타데이터 항목 중, 이름, 설명, 키워드 필드에만 각각 0.7, 0.2, 0.1의 가중치를, 매개변수 α는 0.3, β는 0.25, γ는 0.15, δ는 0.15, ϵ는 0.15의 가중치를 부여하여 데이터셋을 검색한다고 가정하기로 하자. 첫 번째 항은 메타데이터 T_i와 T_j의 필드 간의 연관성 점수를 계산하는 항이다. 주어진 두 개의 메타데이터 T_i와 T_j에서 T_i의 이름 필드와 T_j의 이름 필드와의 연관성 점수가 0.433, 마찬가지로 설명 필드의 연관성 점수가 0.244, 그리고 키워드 필드의 연관성 점수가 0.731이라고 하면, 매개변수 α가 0.3이고 이름에는 0.7, 설명에는 0.2, 키워드에는 0.1의 가중치를 부여하므로 메타데이터 T_i와 T_j의 필드 연관성 점수는 0.128이 된다. $0.3*((0.7*0.433+0.2*0.244+0.1*0.731)=0.128)$ 두 번째 항은 메타데이터 T_j와 원천 질의 q의 연관성 점수를 계산하는 항이다. 원천 질의 q와 메타데이터 테이블 T_j의 이름 필드와의 연관성 점수가 0.234이고, 설명 필드와의 연관성 점수가 0.642, 그리고 키워드 필드와의 연관성 점수가 0.550라고 하고 각각의 필드 별 가중치를 곱하고 매개변수 β를 곱하면, 질의 q와 메타데이터 테이블 T_j의 연관성 점수는 0.087이 된다 $0.25*(0.7*0.234+0.2*0.642+0.1*0.550)=(0.087)$. 세 번째 항은 메타데이터 테이블 T_j의 최신성을 계산한다. 예컨대, 현재 날짜가 2021년 9월 1일이고 메타데이터 테이블 T_j의 $dateModified$ 필드의 값이 2021년 8월 20일이라고 하면 매개변수 γ를 곱한 최신성 값은 0.043이 된다. $(0.15*\sqrt{\frac{1}{12}}=0.043)$ 네 번째 항은 메타데이터 T_j의 사용성 점수를 계산한다. 예를 들어, T_j의 클릭 수가 252이고 다운로드 수가 152이면 시그모이드 함수를 활용한 식에 클릭 수와 다운로드 수를 더한 값을 대입해 계산한 후 매개변수 δ를 곱한 메타데이터 T_j의 사용성 점수는 0.077이 된다$(0.15 * \frac{1}{1 + e^{-\frac{252 + 152}{10000}}} - \frac{1}{2} = 0.077)$. 다섯 번째 항은 데이터셋 d_j의 품질을 계산한다. 예를 들어 d_j의 데이터셋 품질의 평균 별점이 4점이라 하면 별점에 $\frac{1}{5}$를 곱한 뒤 매개변수 ϵ을 곱하여 데이터셋 품질 값은 0.12가 된다 $(0.15*\frac{4}{5}=0.12)$. 최종적으로 원천 질의 q에 대한 메타데이터 T_i와 T_j간의 연관성 점수는 위 다섯 항의 결과를 모두 더하면 0.455가 된다 $(0.128+0.087+0.043+0.077+0.12=0.455)$.

식(1)과 식(2)에서 사용한 연관성 점수를 계산하는 $Score$ 함수는 질의나 메타데이터 항목 등이 문자열임을 고려할 때, 임의의 두 문자열의 유사도(similarity)를 측정하는 데 사용할 수 있는 코사인 유사도[2], 유클리디안 기반 유사도[3], 오픈소스 검색엔진 elasticsearch[24]에서 사용하는 유사도 산출 알고리즘인 BM25[25] 등 어떤 유사도 측정방식을 사용해도 무방하다.

Ⅳ. 데이터셋 검색 시스템 프로토타입 구현

4장에서는 데이터셋 검색 시스템의 프로토타입을 구현한 결과를 제시한다. 프로토타입에서의 핵심은 주어진 질

2) 코사인 유사도: $s(x,y) = \frac{x \cdot y}{\| x \| \| y \|}$

3) 유클리디안 거리 계산식: $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Calculate number of metadata items n
Set weight w_i for each i
Execute a user-submitted query q
For each dataset in our database
Calculate relevance score s by invoking $Score(q,T,w_i)$
  Return search results in descending order of relevance score s

If user requests other datasets related to the ones returned by the original query
For each dataset except for the ones chosen by the user
  Execute $Score(T_i, T_j, w_i)$ to calculate relevance score s
  Return search results in descending order of relevance score s

그림 4. 데이터셋 검색 시스템 동작 개요
Fig. 4. Dataset search system operation overview

표 1. 개방형 데이터셋 포털 특징
Table 1. Characteristics of open dataset portal

	Data.gov	Kaggle	European Data Portal
URL	data.gov	kaggle.com	data.europa.eu
Number of datasets	335,221	109,665	1,492,939
Number of property	24	44	45
Metadata format	DCAT, Schema.org	Schema.org	DCAT, Schema.org
Characteristic	datasets of U.S.	datasets for machine learning competitions	datasets from public sectors of Europe

의에 의해 검색된 데이터셋 들에 대한 순위값을 산출하는 것으로 전체적인 pseudocode는 <그림 4>와 같다.

데이터셋 검색 시스템을 구현하기 위해 널리 사용되고 있는 잘 알려진 데이터셋 저장소인 데이터거브(Data.gov), 케글(Kaggle), 유럽 데이터 포털(data.europa.eu)에서 데이터셋을 총 24,729개를 수집하였으며, 각각의 특징은 아래 <표 1>에 정리한다.

데이터셋 수집은 오픈소스로 공개된 BeaultifulSoup[26]에서 제공하는 라이브러리를 활용한다. 각 포털에서 상대적으로 메타데이터가 풍부하게 기술되지 않은 데이터셋의 수집은 하지 않았으며, 결과적으로 데이터거브에서는 총 335,221건의 데이터셋중 1,1715건, 캐글에서는 총 109,665건의 데이터셋 중 9,536건, 유럽 데이터 포털에서는 3,478건을 수집하였다. 데이터 수집은 Python 라이브러리인 Selenium[27]을 사용하여 가상의 크롬 웹브라우저를 Python 코드로 작동하게 하여, 가상의 웹브라우저가 데이터 목록 화면을 방문한 페이지에 나타난 모든 데이터셋 상세페이지 링크를 저장하여 순서대로 방문하고, 데이터셋 상세페이지에서 메타데이터 표현방식이 JSON-LD[28]이냐 Microdata[29] 방식이냐 여부에 따라 달리 메타데이터를 추출하여 리스트에 저장하는 방식을 취하였다.

수집한 데이터는 Python의 re 라이브러리를 사용하여 메타데이터의 공백문자나 \n, \t와 같은 문자를 “”로 대체하는 전처리 과정을 거친다. 예컨대 <그림 5>는 수집한 데이터셋 메타데이터 중 설명(description) 항목에서 “\n\n” 이 반복적으로 나타나는 경우이며, 이런 경우 전처리 과정을 거쳐 해당 문자열이 제거되도록 구현하였다.

JSON 형태로 수집한 메타데이터는 원형 그대로 MongoDB에 저장하며, MongoDB 플러그인인 Mongoosastic[30]을 사용하여 Elasticsearch와 연계하여 동기화와 검색을 위한 인덱

```
{
  "@context": "http://schema.org/",
  "@type": "Dataset",
  "name": "Annual Financial Data For Hybrid Metrics",
  "description": "## What's Hybrid Metrics\n\n**Hybrid Metrics shows the balance of financial performance and environmental, social impact.**\n\nIf companies try to achieve sustainable growth,
                                                                    . . . . .
Acknowledgements\n\nThe financial data is acquired from [SimFin](https://simfin.com/) .\nThank you for the very useful API and SimFin's great mission.\n",
  "url": "https://www.kaggle.com/takahirokubo0/annual-financial-data-for-hybrid-cdp-kpi",
  "version": 2,
  "keywords": [
    "subject, people and society, business",
    "subject, people and society, business, finance, investing",
    "subject, people and society, social issues and advocacy",
    "subject, earth and nature, environment",
    "subject, earth and nature, environment, pollution"
  ],
  "license": {
    "@type": "CreativeWork",
    "name": "CC BY-NC-SA 4.0",
    "url": "https://creativecommons.org/licenses/by-nc-sa/4.0/"
  },
  "identifier": [
    "952982"
  ],
  "includedInDataCatalog": {
    "@type": "DataCatalog",
    "name": "Kaggle",
    "url": "https://www.kaggle.com"
  },
  "creator": {
    "@type": "Person",
    "name": "Takahiro  Kubo",
    "url": "https://www.kaggle.com/takahirokubo0",
    "image": "https://storage.googleapis.com/kaggle-avatars/thumbnails/259650-gr.jpg"
  },
```


그림 5. 하이브리드 측정을 위한 년간 회계 데이터셋의 메타데이터
(https://www.kaggle.com/datasets/takahirokubo0/annual-financial-data-for-hybrid-cdp-kpi)
Fig. 5. Metadata of annual financial data for hybrid metrics dataset
(https://www.kaggle.com/datasets/takahirokubo0/annual-financial-data-for-hybrid-cdp-kpi)

싱을 수행한다. 수집된 데이터셋의 메타데이터에 대한 인덱스가 생성되고 난 후, Node.js 웹서버와 연동하여 Elasticsearch를 통한 검색이 가능하도록 프로토타입을 구현하였다. 3장에서 언급한 식(1)과 식(2)는 <그림 4>의 Score(q,T,w_i)와 $Score(T_i, T_j, w_i)$를 사용하여 구현하였으며 이때 사용한 순위 함수는 Elasticsearch의 기본 순위 함수인 BM25를 사용하여 구현하였다. 다만, 본 연구에서 제안한 새로운 방식의 순위 알고리즘의 효용성을 입증하기 위해서는 구현한 프로토타입을 활용하여 추가적인 실험이 필요한 상황이다.

V. 결론 및 향후 연구과제

본 논문은 하나 이상의 데이터셋을 데이터 분석에 활용하고자 하는 데이터 분석가가 필요로 하는 데이터셋을 검색하는 데, 데이터셋의 설명자료에 해당하는 메타데이터를 색인의 일부로 활용하여 검색을 수행하는 것은 물론, 검색된 데이터셋을 정렬하는 데 본 논문에서 제안하는 차별화된 연관성 점수 계산 방식을 사용하여 이를 순위값으로 활용함으로써 데이터 분석가의 원천 요구사항을 기술한 질의는 물론, 주어진 질의에 의해 검색된 데이터셋과 융합을 할

수 있는 후속 데이터셋 검색을 용이하게 수행할 수 있어 단일 데이터셋을 사용한 데이터 분석은 물론 여러 데이터셋 융합에 의한 데이터 분석이 필요한 경우의 데이터셋 검색에 적용할 수 있는 기술을 고안했다는 데서 그 의의를 찾을 수 있다. 또한 본 논문에서 제안한 방식은 메타데이터를 이용한 색인을 기반으로 검색을 수행하는 방식이므로 이미지, 동영상, 사운드 등 다양한 형태의 멀티미디어 데이터셋을 대상으로 검색을 수행하고자 하는 경우에도 적용이 가능할 것으로 기대된다.

다만, 현 단계에서는 기존 오픈소스 검색엔진인 Elasticsearch를 사용해 주어진 질의에 부합하는 연관성 점수를 순위값으로 반환하는 프로토타입 구현을 마친 상태이며, 향후 연구과제로는 이의 효용성을 입증하기 위해서 제한적이나마 구축한 데이터셋에 대한 샘플 질의를 선정하고, 각 샘플 질의 수행결과에 대한 precision, recall, F-measure를 측정함으로써 우리가 신규로 제안한 방식과 기존 텍스트 검색에 최적화된 BM25 함수나 기타 유사도 측정 함수를 적용한 순위 알고리즘과의 성능 차이를 보이는 것이 필요하다.

참 고 문 헌 (References)

[1] Data Catalog Vocabulary (DCAT) - Version 2, https://www.w3.org/TR/vocab-dcat-2/ (accessed Feb. 04, 2020).
[2] Schema.org https://schema.org/ (accessed Mar. 17, 2022).
[3] A. Chapman, E. Simperl, L. Koesten, G. Konstantinidis, L. Ibáñez, E. Kacprzak, and P. Groth, “Dataset search: a survey,” The VLDB Journal, Vol. 9, No.1, pp. 251-272, Jan. 2020.
doi: https://doi.org/10.1007/s00778-019-00564-x
[4] M. Thelwall and K. Kousha, “ Figshare: a universal repository for academic resource sharing?” Online Information Review, Vol. 40, No. 3, pp. 333–346, June 2016.
doi: https://doi.org/10.1108/OIR-06-2015-0190
[5] M. Altman, E. Castro, M. Crosas, P. Durbin, A. Garnett, and J. Whitney, “Open journal systems and dataverse integration—helping journals to upgrade data publication for reusable research,” Code4Lib Journal, Issue 30, Oct. 2015.
[6] Elsevier scientific repository, https://datasearch.elsevier.com/ (accessed July 4, 2022).
[7] Data.gov, https://data.gov/ (accessed July 5, 2022).
[8] Korean public data portal (data.go.kr), https://www.data.go.kr/en/index.do (accessed June 13, 2022).
[9] Kaggle, https://www.kaggle.com/ (accessed May 14, 2022).
[10] European data portal, https://data.europa.eu/en (accessed June 15, 2022).
[11] Google dataset search, https://datasetsearch.research.google.com (accessed June 7, 2022).
[12] J. Hendler, J. Holm, C. Musialek, and G. Thomas, “Us government linked open data: Semantic.data.gov.,” IEEE Intelligent Systems, Vol. 27, No. 3, pp. 25–31, May 2022.
doi: https://doi.org/10.1109/MIS.2012.27
[13] Linked open data cloud, https://lod-cloud.net/ (accessed Mar. 28, 2022).
[14] Open data monitor, https://opendatamonitor.eu/ (accessed June 21, 2022).
[15] Uk open data portal, https://data.gov.uk/ (accessed June 20, 2022).
[16] CKAN – The open source data management system, https://ckan.org/ (accessed Mar. 27, 2022).
[17] Apache Lucene, https://lucene.apache.org/ (accessed Oct. 15, 2021).
[18] Apache Solr, https://solr.apache.org/ (accessed Oct. 15, 2021).
[19] R. Miller, “Open Data Integration,” Proceedings of the VLDB Endowment, Vol. 11, No. 12, pp. 2130-2139, Aug. 2018.
doi: https://doi.org/10.14778/3229863.3240491
[20] N. Noy, M. Burgess, and D. Brickley, “Google dataset search: building a search engine for datasets in an open web ecosystem,” The World Wide Web Conference 2019, San Francisco, USA, pp. 1365-1375, May 13, 2019.
doi: https://doi.org/10.1145/3308558.3313685
[21] S. Sansone, A. González-Beltrán, P. Rocca-Serra, G. Alter, J. Grethe, H. Xu, I. Fore, J. Lyle, A. Gururaj, X. Chen, H. Kim, N. Zong, Y. Li, R. Liu, I. Burak Ozyurt, and L. Ohno-Machado, “Dats, the data tag suite to enable discoverability of datasets,” Scientific data, Vol. 4, No. 1, pp. 1-8, June 2017.
doi: https://doi.org/10.1038/sdata.2017.59
[22] S. Neumaier and A. Polleres, “Enabling spatio-temporal search in open data,” Journal of Web Semantics, Vol. 55, pp. 21-36, Mar. 2019.
doi: https://doi.org/10.1016/j.websem.2018.12.007
[23] S. Neumaier, J. Umbrich, A. Polleres, “Automated quality assessment of metadata across open data portals,” Journal of Data and Information Quality, Vol. 8, No. 1, pp. 1-29 Oct. 2016.
doi: https://doi.org/10.1145/2964909
[24] Elasticsearch, https://www.elastic.co/kr/ (accessed Mar. 04, 2020).
[25] Practical BM25-Part 2: The BM25 algorithms and its variables, https://www.elastic.co/kr/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables (accessed Mar. 05 2020).
[26] Beautiful Soup documentation, https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (accessed Dec. 12, 2021).
[27] Selenium, https://www.selenium.dev/ (accessed Jan. 15, 2022).
[28] JSON for linking data, https://json-ld.org/ (accessed Feb. 22, 2022).
[29] HTML Microdata, https://www.w3.org/TR/2021/NOTE-microdata-20210128/(accessed Feb. 23, 2022).
[30] Mongoosastic, https://mongoosastic.github.io/mongoosastic/ (accessed Mar. 03, 2022).

저 자 소 개

최 우 영

- 2019년 : 공주대학교 화학공학과 (학사)
- 2022년 : 명지대학교 데이터테크놀로지 전공 (석사 수료)
- ORCID : https://orcid.org/0000-0003-3148-4290
- 주관심분야 : 데이터베이스, 데이터셋 검색, 소셜 네트워크

전 종 훈

- 1986년 : University of Denver 전산과학과 (학사)
- 1988년 : Northwestern University 컴퓨터공학과 (석사)
- 1992년 : Northwestern University 컴퓨터공학과 (박사)
- 1992년 ~ 1995년 : University of Central Oklahoma 전산과학과 조교수
- 1995년 ~ 현재 : 명지대학교 컴퓨터공학과/융합소프트웨어학부 교수
- 2011년 ~ 현재 : ㈜프람트테크놀로지 대표이사
- ORCID : https://orcid.org/0000-0003-3396-4239
- 주관심분야 : 데이터베이스, 데이터셋 검색, 지능형 소프트웨어