웹 기반 옛한글 문서 실시간 검색 플랫폼

한동민 · 황대영 · 조중현 · 김동균 · 이상정
순천향대학교

Web based Real-time Search Platform for Old Korean Documents

Dong-Min Han · Dae-Young Hwang · jung-hyeon jo · Dong-Kyun Kim · Sang-Jeong Lee
SoonChunHyang University
E-mail : ggamangk@gmail.com · dy.hwangs@gmail.com · junghyeon2003@gmail.com
kdk70@sch.ac.kr · sjlee@sch.ac.kr

요 약

오래된 고문서의 영구보존과 체계적인 연구를 위해서는 이 문서를 디지털화 하고 저장한 후 효율적으로 참조하고 검색할 수 있는 환경이 필수적이다. 그러나 현재 옛한글 문서들은 표현 방식이 통일되지 않아서 문서의 색인 및 검색에 어려움이 있다. 본 논문에서는 시대별로 수집한 옛한글 문서의 인코딩을 유니코드로 규격화하고, 실시간 분석과 검색을 위해 엘라스틱서치 저장소에 문서들을 색인하고 저장한다. 또한 옛한글의 체계적인 연구를 위해서 연구자들이 엘라스틱서치 기반의 저장소에 접근이 가능한 웹 기반의 검색 플랫폼을 설계하고 구현한다.

키워드
엘라스틱서치, 옛한글, 유니코드, 한양 PUA 코드

Ⅰ. 서 론

옛 한글 문서의 보존 및 저장과 연구의 효율성을 위해서 고문서의 디지털화는 필수 불가결한 일이나 유니코드 5.2 이전에는 옛한글의 모든 자소가 유니코드 상에 없었기 때문에 많은 문서들이 한양 PUA 코드로 작성되었다. 하지만 한양 PUA 코드는 표준 코드가 아니며 완성형 형태로 지원하는 글자의 개수가 5천여 개밖에 되지 않아 그 한계가 명확하기 때문에 사용이 어렵다[1]. 유니코드 5.2부터 옛한글의 모든 자소가 추가되어 옛한글의 모든 글자를 유니코드만으로 작성할 수 있게 되었다. 하지만 그 결과 과거의 문서와 최근 작성하는 문서 간에 코드가 호환되지 않아 문서의 색인과 검색에 어

려움이 있다. 또한 최근 기계 학습(machin learning)에 대한 관심이 높아짐에 따라 옛한글 문서에 대한 연구에도 기계학습을 접목시킬 수 있다. 기계학습에는 데이터 전처리와 분석이 필수적이기 때문에 옛한글 문서의 포맷을 통일하고 형태소를 분석하여 색인하는 과정의 필요성은 점점 커진다고 할 수 있다.

본 논문에서는 통일되지 않은 문서들의 포맷을 유니코드로 통일하고 파이썬으로 작성한 데이터 수집 모듈을 통해 옛한글 문서가 엘라스틱서치에 자동으로 색인이 되도록 한다. 또한 엘라스틱서치 저장소에 접근이 가능한 검색 플랫폼을 웹 기반으로 작성하여 기기 독립적으로 사용자들이 실시간 검색 플랫폼에 접근할 수 있도록 하였다. 사용자는 단순히 문서를 조회하는 것뿐만 아니라 전문 검색, 단어 검색이 가능하다. 그리고 옛한글을 검색할 때에는 별도의 입력기가 필요하므로 자바스크립트로 검색 플랫폼에 입력기를 추가로 구현한다.

II. 이론적 배경

2.1 옛한글

옛한글이란 현대 한글에서 쓰지 않는 글자를 말한다. 옛한글을 컴퓨터에서 표기할 때 문자 코드 체계를 완성형으로 구현하기에는 그 양이 방대해서 첫가끝[1] 코드라는 조합형 체계로 구현되어 있다. 옛날 유니코드에 옛한글의 모든 자소가 없었을 때는 주로 한양 PUA라는 사용자 정의 영역 코드를 사용해서 옛한글 문서를 작성했으나 한양 PUA 코드는 비표준 코드이고 옛한글의 모든 글자를 표현할 수 없어 그 한계가 명확했다. 유니코드 5.2에서 옛한글의 모든 자소가 추가되어 옛한글 문서를 완벽하게 유니코드로 작성할 수 있게 되었다. 그 결과 최근의 문서들은 유니코드로 작성되었고, 과거 한양 PUA 코드로 작성되었던 문서들과 문서 인코딩이 통일되지 않아서 검색이 제대로 되지 않는 문제점이 발생했다.

2.2 엘라스틱서치

엘라스틱서치는 확장 가능한 고성능 정보 검색 라이브러리인 루씬을 기반으로 개발된 오픈소스 분산 검색엔진이다[2]. 엘라스틱서치는 설치와 적용 과정에 복잡한 설정 과정이 필요하지 않아서 기존 개발 혹은 운영 중이던 시스템에 적용하기 용이하며 분산 시스템이기 때문에 클러스터의 확장이 쉽고 자체적으로 부하를 노드에 분배하기 때문에 검색 용량이 증가했을 때 대응하기가 수월하다.

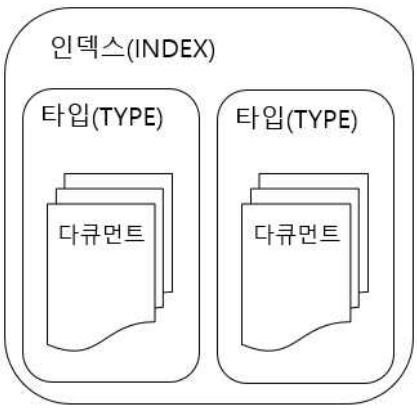

그림 1. 엘라스틱서치 데이터 계층

그림 1은 엘라스틱 서치의 데이터 계층을 나타낸 것이다. 엘라스틱서치는 JSON 문서 기반으로 인덱스, 타입, 그리고 다큐먼트로

1) 옛한글과 현대한글에서 나타낼 수 있는 한글의 초성, 중성, 종성의 자모 요소를 한 글자로 나타낼 수 있도록 부호화하고, 이 자모들을 초성, 중성, 종성 순서대로 한글을 표현하는 방식

데이터를 분류하여 저장한다. 하나의 인덱스 아래에는 여러 개의 타입이 올 수 있으며 여러 개의 타입 아래에는 여러 다큐먼트가 저장될 수 잇다. REST API에서 다큐먼트를 표현할 때는 '/index/type/document'와 같이 표현한다.

III. 실시간 검색 시스템

3.1 시스템 구성

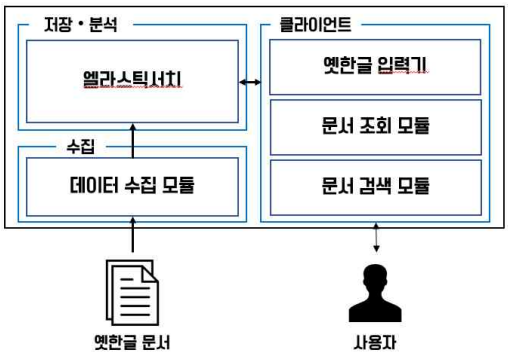

그림 2. 옛한글 문서 실시간 검색 플랫폼 구성도

그림 2는 옛한글 문서 실시간 검색 플랫폼의 시나리오이다. 사용자는 데이터 수집 모듈을 통해 옛한글 문서의 형식을 JSON 형식으로 변환해서 엘라스틱서치에 색인한다. 또한 엘라스틱서치는 모든 요청과 응답을 JSON 형식으로 하기 때문에 일반 사용자가 사용하기에는 불편함이 따르므로 클라이언트를 추가로 구성했다. 사용자는 클라이언트에서 엘라스틱서치에 색인된 옛한글 문서들을 조회하고 옛한글 입력기 모듈을 사용하여 옛한글을 입력하여 옛한글 문서의 전문 검색, 부분 검색을 할 수 있다.

3.2 옛한글 다큐먼트 구성

본 논문은 문서의 검색의 정확도를 높이고 쿼리를 효율적으로 작성하기 위해 옛한글 문서 다큐먼트의 구성을 통일했다. 모든 옛한글 문서들은 JSON 형식으로 엘라스틱서치에 색인된다.

표 2. 옛한글 다큐먼트 JSON 구성

Name	Value
index	문서 발행 세기
type	책의 제목
tag	구분자
text_entry	단락
order	행의 순서
author	저자
year	문서 발행 연도

```
GET /ac18/_mapping?pretty
"properties": {
  "author": {
    "type": "text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    }
  },
  "order": {
    "type": "long"
  },
  "tag": {
    "type": "text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    }
  },
  "text_entry": {
    "type": "text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    },
    "analyzer": "whitespace"
  },
  "year": {
    "type": "long"
  }
}
```

그림 3. 엘라스틱서치에 색인한 다큐먼트의 매핑 정보

표 2는 옛한글 문서가 색인될 때의 JSON 문서 구성을 나타낸다. 다큐먼트를 조회할

때 세기별, 책별 다큐먼트 조회와 검색을 효율적으로 요청하기 위하여 index는 책이 발행된 세기로, type은 책의 제목으로 구성했다. tag는 옛한글 문서에서 행을 구분하는 구분자이고, text_entry는 행의 전문을 나타낸다. author는 해당 옛한글 문서의 저자, year는 해당 옛한글 문서의 작성 연도를 나타내며 order는 행의 순서를 나타낸다.

그림 3은 RESTful API를 사용하여 실제 문서의 매핑 정보를 확인한 것으로 각각의 필드에 맞는 타입이 매핑된 것을 확인할 수 있다.

3.3 웹 클라이언트

그림 4는 웹 클라이언트의 구성을 나타낸다. 웹 클라이언트는 NodeJS 기반으로 작성되었으며 크게 문서 조회 모듈과 문서 검색 모듈로 나뉜다. 문서 조회 모듈은 엘라스틱서치 저장소로 쿼리를 보내서 요청한 문서의 내용을 가져와 화면에 표시한다. 그리고 문서 검색 모듈은 엘라스틱 저장소에서 사용자가 요청한 단어가 포함된 모든 문서들을 화면에 표시한다. 이 과정에서 사용자가 손쉽게 옛한글을 입력해서 검색할 수 있도록 옛한글 입력기가 자바스크립트로 구현되어 있다.

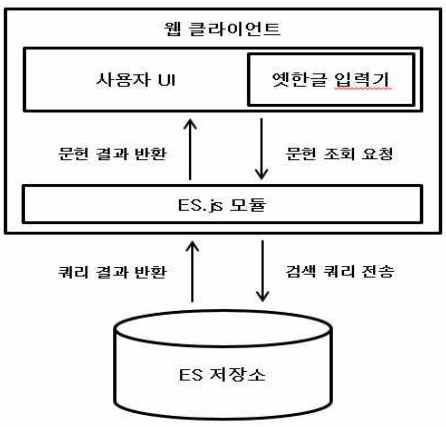

그림 4. 웹 클라이언트 구성도

3.4 데이터 수집 모듈

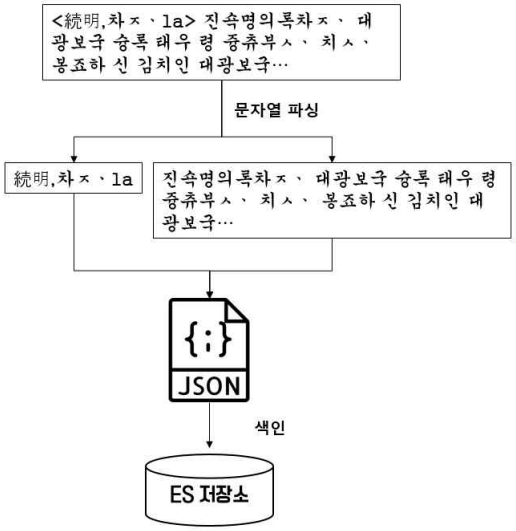

그림 5. 데이터 수집 모듈 시나리오

데이터 수집 모듈은 파이썬으로 작성되었으며 옛한글 문서를 괄호를 기준으로 파싱하여 구분자와 내용을 분리한다. 그리고 딕셔너리에 문서에 대한 추가정보와 함께 저장한 뒤 elasticsearch 모듈을 사용하여 엘라스틱서치에 색인한다. 이 때 파이썬의 딕셔너리 자료구조와 JSON 문서의 형식은 일치하기 때문에 추가적인 변환 과정은 필요로 하지 않는다. 그림 5는 데이터 수집 모듈의 시나리오를 나타낸다.

Ⅳ. 구현 및 테스트

4.1 구현

본 논문의 옛한글 실시간 검색 시스템은 엘라스틱서치 5.1.1을 기반으로 구현하였다. 데이터 수집 모듈은 파이썬으로 작성하였다. 구현에 사용한 옛한글 문서는 속명의록, 몽어노걸대이다[3].

그림 6. POSTMAN을 사용하여 색인한 다큐먼트를 확인

데이터 수집 모듈로 색인된 옛한글 다큐먼트에서 tag는 not_analyzed로 색인되며 텍스트 전문은 엘라스틱서치에서 whitespace 분석기를 통해 토큰 단위로 나뉘어 색인되도록 매핑했다. 이 과정에서 엘라스틱서치에 데이터가 제대로 색인되었는지 확인하기 위하여 그림 6과 같이 POSTMAN이라는 구글 확장 도구를 사용하여 엘라스틱서치 저장소에 GET 요청을 전송하여 색인된 문서를 확인했다.

클라이언트는 NodeJS 기반으로 작성되었으며 문서 조회, 검색 모듈로 구성된다. 모든 요청은 엘라스틱서치 저장소에 쿼리로 전송된다. 옛한글 입력기는 그림 7과 같이 구현하였는데 일반적으로 사용하는 키보드 배열은 두벌식이나 옛한글은 세벌식 조합형 방식이므로 사용자의 편의성을 위해 마우스로 입력을 받도록 구현하였다.

그림 7. 구현된 입력기 화면

4.2 테스트

테스트에서는 클라이언트에서 엘라스틱서치 기반 저장소에 색인된 모든 인덱스의 이름을 가져온다. 해당 인덱스에 들어있는 타입 중에 하나를 선택하면 해당 타입에 들어있는 모든 다큐먼트를 쿼리를 이용하여 그림 8과 같이 조회하도록 구현했다. 이 때, 엘라스틱서치는 내부 다큐먼트들에 대한 순서가 존재하지 않으므로 클라이언트에서 다큐먼트를 문장 순서대로 표시하기 위해서 우선 전체 다큐먼트를 클라이언트로 가져온 뒤에 order의 값을 기준으로 정렬해서 출력했다. 이 때 일반 사용자가 보기 편하도록 클라이언트 화면을 그림 9와 같이 구성했다.

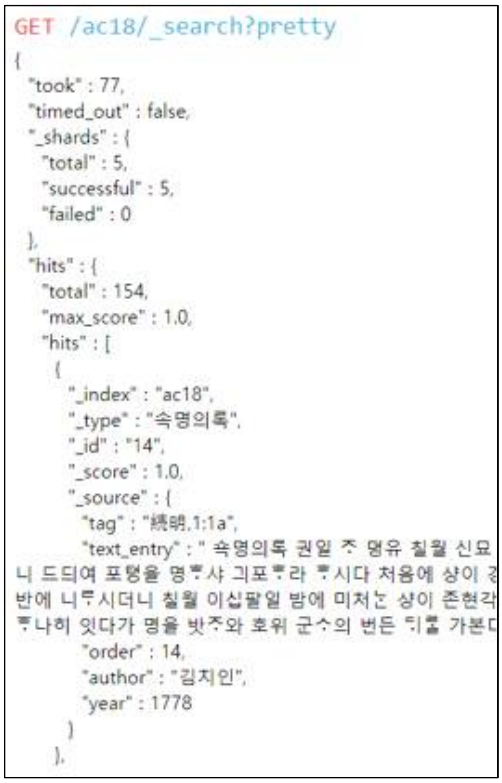

그림 8. RESTful API로 문서 조회를 요청한 결과 JSON 문서

또한 해당 타입에서 특정 단어를 검색했을 때 엘라스틱서치에 쿼리문을 전송하여 그 결과를 클라이언트에 표시한다. 그림 10은 RESTful API로 엘라스틱서치 저장소에 검색 쿼리를 전송해서 받아온 검색 결과의 JSON 문서이다.

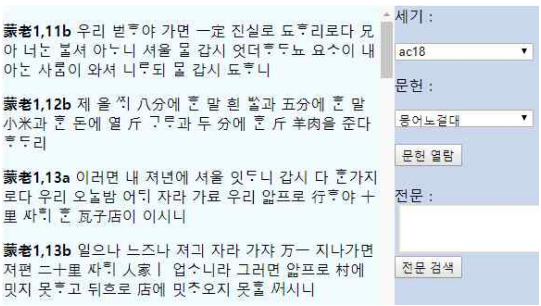

그림 9. 문서 조회 클라이언트 화면

그림 10. RESTful API로 전문 검색을 요청한 결과 JSON 문서

그림 11. 전문 검색 클라이언트

문서 조회와 마찬가지로 일반 사용자가 보기 편하도록 그림 11처럼 클라이언트에서 결과를 표시하였다. 검색 요청을 한 '잇ᄃᆞ니'가 포함된 문서의 내용만 표시되는 것을 확인할 수 있다.

Ⅴ. 결론 및 향후 연구

본 논문 화면에서는 옛한글 문서의 인코딩을 유니코드로 통일한 옛한글 문서 저장소를 엘라스틱서치 기반으로 구현하고 옛한글 문서 저장소에서 문서 조회, 검색이 가능한 클라이언트를 설계·구현하였다.

파이썬으로 작성된 데이터 수집 모듈에서 자동으로 옛한글 문서를 엘라스틱서치 기반 저장소에 색인하고 저장소에 색인된 옛한글 문서들을 클라이언트에서 조회, 검색이 가능하도록 하여 연구자들이 좀 더 쉽게 자료에 접근이 가능하도록 했다. 또한 조회 및 검색 시에 옛한글을 입력할 수 있는 별도의 입력기를 추가로 구현해주고 테스트하였다.

향 후 옛한글의 형태소 분석을 보완하여 어절 단위 검색의 정확도를 높이는 연구를 수행할 예정이다. 옛한글 형태소 분석이 완료되어 적용된다면 검색의 정확도가 더욱 높아지고 기계학습을 할 수 있게 되어 말뭉치 생성, 옛한글 문서 번역 등 다양한 연구에 접목 가능하게 되므로, 향후 옛한글 형

태소 분석기를 설계·구현하고 하둡 스파크와 연동하여 옛한글 문서들을 기계 학습시킬 예정이다.

참고문헌

[1] 이동주, 연종흠, 황인범 & 이상구, 꼬꼬마 : 데이터베이스를 활용한 세종말뭉치 활용도구, 정보과학회논문지, pp. 1046-1050, 2010.
[2] 이상용, 아파치 엘라스틱서치 기반 로그스태시를 이용한 보안로그분석 시스템, 학위논문(석사), 2015
[3] 임용기, 21세기 세종계획 국어 특수자료구축 보고서, 국립국어원, 2005.
[4] https://www.elastic.co/
[5] 옛한글 입력기, http://ohi.pat.im/