JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 9: 1135~1143, Sep. 2021

빈도 분석을 이용한 HTML 텍스트 추출

김진환[1] · 김은경[2*]

HTML Text Extraction Using Frequency Analysis

Jin-Hwan Kim[1] · Eun-Gyung Kim[2*]

[1]Graduate Student, Department of Computer Science & Engineering, Korea University of Technology and Education, Cheonan, 31253 Korea
[2*]Professor, School of Computer Science & Engineering, Korea University of Technology and Education, Cheonan, 31253 Korea

요 약

최근 빅데이터 분석을 위해 웹 크롤러를 이용한 텍스트 수집이 빈번하게 이루어지고 있다. 하지만 수많은 태그와 텍스트로 복잡하게 구성된 웹 페이지에서 필요한 텍스트만을 수집하기 위해서는 웹 크롤러에 빅데이터 분석에 필요한 본문이 포함된 HTML태그와 스타일 속성을 명시해야 하는 번거로움이 있다. 본 논문에서는 HTML태그와 스타일 속성을 명시하지 않고 웹 페이지에서 출현하는 텍스트의 빈도를 이용하여 본문을 추출하는 방법을 제안하였다. 제안한 방법에서는 수집된 모든 웹 페이지의 DOM 트리에서 텍스트를 추출하여 텍스트의 출현 빈도를 분석한 후, 출현 빈도가 높은 텍스트를 제외시킴으로써 본문을 추출하였으며, 본 연구에서 제안한 방법과 기존 방법의 정확도 비교를 통해서 본 연구에서 제안한 방법의 우수성을 검증하였다.

ABSTRACT

Recently, text collection using a web crawler for big data analysis has been frequently performed. However, in order to collect only the necessary text from a web page that is complexly composed of numerous tags and texts, there is a cumbersome requirement to specify HTML tags and style attributes that contain the text required for big data analysis in the web crawler. In this paper, we proposed a method of extracting text using the frequency of text appearing in web pages without specifying HTML tags and style attributes. In the proposed method, the text was extracted from the DOM tree of all collected web pages, the frequency of appearance of the text was analyzed, and the main text was extracted by excluding the text with high frequency of appearance. Through this study, the superiority of the proposed method was verified.

키워드 : 웹 본문 추출, 텍스트 마이닝, 웹 크롤링, 빅데이터, 텍스트 빈도 분석

Keywords : Web content extraction, Text mining, Web crawling, Big data, Text frequency analysis

Received 25 June 2021, **Revised** 1 July 2021, **Accepted** 19 July 2021

*** Corresponding Author** Eun-Gyung Kim(E-mail: egkim@koreatech.ac.kr Tel:+82-41-560-1350)
Professor, School of Computer Science & Engineering, Korea University of Technology and Education, Choenan, 31253 Korea

Open Access **http://doi.org/10.6109/jkiice.2021.25.9.1135** print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 사회적 트렌드 및 현상에 대한 언론과 대중의 반응을 분석하고, 사회적 인식 변화를 탐색하기 위한 목적으로, 뉴스 및 SNS 등의 다양한 매체로부터 수집한 텍스트를 이용한 빅데이터 분석이 활발하게 이루어지고 있다[1-5]. 이때 빅데이터 수집의 자동화를 위하여 웹 크롤러가 이용된다. 웹 크롤러는 대상 웹 사이트에서 수집 조건에 맞는 웹 페이지의 URL을 추출하고, URL에 접근하여 웹 페이지를 수집하며, 웹 페이지에서 필요한 텍스트를 추출하는 작업을 반복적으로 수행한다.

그러나 웹 페이지는 수많은 태그와 다양한 텍스트로 복잡하게 구성되어 있기 때문에, 이용 안내나 광고 문구 같은 의미 없는 텍스트는 제외하고 의미 있는 본문 텍스트만을 정확히 추출하기 위해서는 사람이 직접 웹 페이지의 구성을 분석하여 본문이 포함된 HTML태그와 스타일 속성을 웹 크롤러에 명시해야 한다. 하지만 웹 페이지의 구조가 변경되는 경우 해당 웹 페이지의 구조 분석을 다시 해야 한다는 단점이 있다. 이러한 단점을 보완하고자 [6]~[8]에서는 태그에 포함된 텍스트의 단어 밀도를 계산하여 높은 밀도를 가지는 태그 및 그와 인접한 태그로부터 텍스트를 자동으로 추출하는 방안을 제안하였지만, 이는 뉴스와 같이 본문 영역이 명확하고 단어 밀도가 높으며 태그와 스타일 구성이 간단한 경우에는 높은 정확도를 가지지만, 단어 밀도가 낮거나 웹 페이지의 태그와 스타일의 구성이 다양한 웹 페이지에서는 정확도가 떨어진다. [9, 10]에서는 이러한 다양한 구성의 웹 페이지에서 텍스트 추출의 정확도를 향상시키기 위해 태그에 포함된 텍스트의 필요 여부를 사람이 직접 판단하여 구축한 훈련 데이터를 학습시켜 본문을 추출하는 방안을 제안하였지만, 학습을 위한 충분한 훈련 데이터 구축이 어렵다는 한계점을 갖는다.

본 논문에서는 웹 페이지에서 출현하는 텍스트의 빈도를 이용하여 태그 구성과 스타일 속성이 다양한 웹 페이지에서 본문을 추출하는 방법을 제안하였다. 제안한 방법은 수집된 모든 웹 페이지의 DOM 트리에서 텍스트를 추출하여 빈도를 분석하고, 출현 빈도가 높은 텍스트를 제거하여 추출할 텍스트를 결정하기 때문에, 태그와 스타일 정보를 필요로 하지 않으며, 훈련 데이터 역시 구축하지 않아도 된다.

본 논문의 2장에서는 웹 페이지에서의 본문 추출과 관련된 연구에 대해 기술하고, 3장에서는 본 논문에서 제안하는 텍스트 추출 방식에 대하여 설명한다. 4장에서는 다른 방식과의 비교를 통해 제안한 방식의 우수성을 입증한다.

Ⅱ. 관련 연구

웹 콘텐츠 추출이란 HTML, CSS, JavaScript 등의 수많은 코드와 복잡한 구조로 이루어진 웹 페이지에서 필요한 정보 추출을 자동으로 수행하는 것으로, 이를 위해 DOM 트리 분석, 불필요한 태그 제거, 텍스트 블록 구분 등의 관련 연구가 진행되어 왔다.

웹 페이지는 텍스트 노드(Text Node)를 하나의 블록(block)으로 여러 개의 텍스트 블록으로 분할할 수 있다[9]. 텍스트 노드는 본문을 비롯한 웹 페이지 상의 모든 텍스트를 포함하고 있으며, DOM 트리(Document Object Model tree) 구조에서 트리의 최하위에 리프 노드(Leaf Node)로 표현된다.

[6]에서는 웹 페이지에서 텍스트 블록을 추출한 후, 해당 블록에서 내용과 링크를 구성하는 단어의 수를 이용하여 계산한 단어/링크 밀도로 블록 정보를 나타내고, 블록의 전후에 출현하는 블록 정보를 학습하여 블록의 본문 여부를 구분하였으며, [7]에서는 본문 추출에 참고하는 인접 블록 정보를 확장하여 [6]의 성능을 향상시켰다. [8]에서는 웹 페이지로부터 정확한 본문 영역만을 추출하기 위하여 전처리를 통해 불필요한 DOM 요소들을 제거한 후 가장 긴 텍스트를 가진 블록 요소를 추출하고, 블록에 포함된 텍스트 간의 유사도에 따라 군집화하여 최종적으로 본문을 추출하는 방법을 제안하였다. 하지만 이러한 방식은 뉴스와 같이 템플릿이 단순한 경우에는 정확도가 높게 나타나지만, 블로그와 같이 다양한 템플릿을 가지는 경우에는 정확도가 낮다는 단점이 있다.

[9]에서는 DOM 트리 구조 중 텍스트 블록을 128개의 특징을 갖는 시퀀스 형태의 입력 데이터를 생성하여 시퀀스 레이블링(Sequence Labeling)의 문제로 텍스트 추출 문제에 접근하여 텍스트 블록의 본문 여부를 판단하고자 하였다. [10]에서는 텍스트 블록에서 추출해야 하는 특징이 많다는 [9]의 단점을 보완하기 위해 특징 추출의 자동화를 위해 텍스트 블록의 부모 태그 및 단어

분포로 구성된 입력 데이터를 이용한 학습 방안을 제안하였다. 그러나 지도학습을 기반으로 하는 본문 추출 방식은 학습을 위한 충분한 훈련 데이터 구축이 어렵다는 한계점을 갖는다. 따라서 본 논문에서는 태그와 스타일 정보 및 훈련 데이터 없이 웹 페이지에서 필요한 텍스트를 정확히 추출하기 위하여 텍스트의 출현 빈도를 이용하는 방법을 제안하였다.

Ⅲ. 빈도 분석을 이용한 HTML 텍스트 추출

일반적으로 빅데이터 분석을 위해 웹 페이지로부터 본문을 추출하기 위해서는 텍스트가 포함된 태그와 스타일 속성을 웹 크롤러에게 명시해줘야 한다. 예를 들어, 그림 1과 같이 웹 페이지(web_doc.html)에는 본문 텍스트가 p 태그에 포함되어 있다면 웹 크롤러(web_crawler.py)는 p 태그에 접근하기 위하여 p 태그의 상위인 div 태그와 id 선택자(main-container)를 알고 있어야 한다. 이러한 정보 없이 웹 크롤러가 단순히 p 태그에 있는 텍스트를 모두 추출한다면, 웹 페이지 내의 다른 p 태그가 포함하는 텍스트까지 추출하기 때문에 정확한 본문 내용을 추출할 수 없게 된다.

하지만 웹 크롤러가 텍스트를 추출하기 위한 태그 및 스타일 속성에 대한 정보는 개발자가 직접 해당 웹 페이지를 분석하여 알아내야 하며, 웹 페이지의 구조가 바뀔 때마다 정보를 업데이트해야 한다. 또한 블로그와 같이 이용자마다 웹 페이지를 구성하는 태그와 스타일 속성이 다양한 경우에는 개발자가 대응해야 하는 템플릿을 모두 파악하기 어렵다는 단점 역시 존재한다. 이처럼 기존 방식에서 발생하는 텍스트 수집의 비효율성은 웹 페이지의 구조나 구성에 의존하기 때문에 발생하며, 이를 해소하여 효율적으로 텍스트를 수집하기 위해서는 웹 페이지의 다른 특성에 주목하여 텍스트를 추출할 필요가 있다. 이를 위해 본 논문에서는 텍스트의 출현 빈도를 기반으로 웹 페이지에서 본문 내용만을 추출하는 방법을 제안하였다.

웹 페이지에서 추출되는 텍스트는 해당 문서 내에서 차지하는 비중이 텍스트마다 다르다. 이중 웹 페이지의 이용 안내, 지시 문구와 같은 텍스트는 비록 이용자의 편의를 향상시키기 위한 의도로 웹 페이지에서 높은 빈도로 출현하지만, 기능적인 목적만을 가질 뿐 추후 텍스트 분석을 위한 용도로는 거의 채택되지 않는다. 이와는 대조적으로 실제 수집 대상이 되는 텍스트는 의미적으로는 중요하지만, 그 형태가 중복되는 경우가 드물어 동일한 텍스트가 여러 웹 페이지에서 출현하는 경우가 적은 편이다. 예를 들어, 그림 2는 웹 크롤러를 이용하여 네이버 블로그에서 '코로나 백신'을 키워드로 수집한 800개의 웹 페이지에서 추출한 텍스트의 빈도를 내림차순으로 정렬한 것으로, 가장 높은 출현 빈도를 나타낸 텍스트는 '알림을 모두 삭제하시겠습니까', '나만의 즐겨찾기를 추가해 보세요', '이 블로그 카테고리 글' 등인데, 이는 해당 블로그의 이용과 관련된 안내 문구나 지시 문구로서, 수집한 모든 웹 페이지에서 공통으로 출현한 것이다. 하지만 빈도 순위가 하위권에 위치하는 텍스트들은 실제 블로그 이용자들에 의해 작성된 텍스트로, 코로나 백신과 관련된 정보나 저자의 생각에 대한 내용이 주로 담겨져 있으며, 텍스트가 작성자의 저작 스타일에 따라 독자적인 형태로 표현된 것을 알 수 있다.

web_doc.html

```
<html>
...
<div class="location_component">
  <div class="location">
    <div id="main-container">
      <p style="line-height:1.9;" >
        <span> CONTENT TEXT </span>
      </p>
    </div>
  </div>
  <p style="line-height:1.9;" >
    <span> TEXT </span>
  </p>
</div>
...
</html>
```

web_crawler.py

```
def extract_text(script):
  block = ''
  candidate_style_id_list = [
    'main-container', 'postViewArea',
    'sect_dsc', 'post-view']

  for style_selector in candidate_style_class_list:
    try:
      block = script.find_element_by_id(style_selector)
      break
    except Exception as e:
      pass
  return block.text
```

Fig. 1 Text Extraction method of web crawler

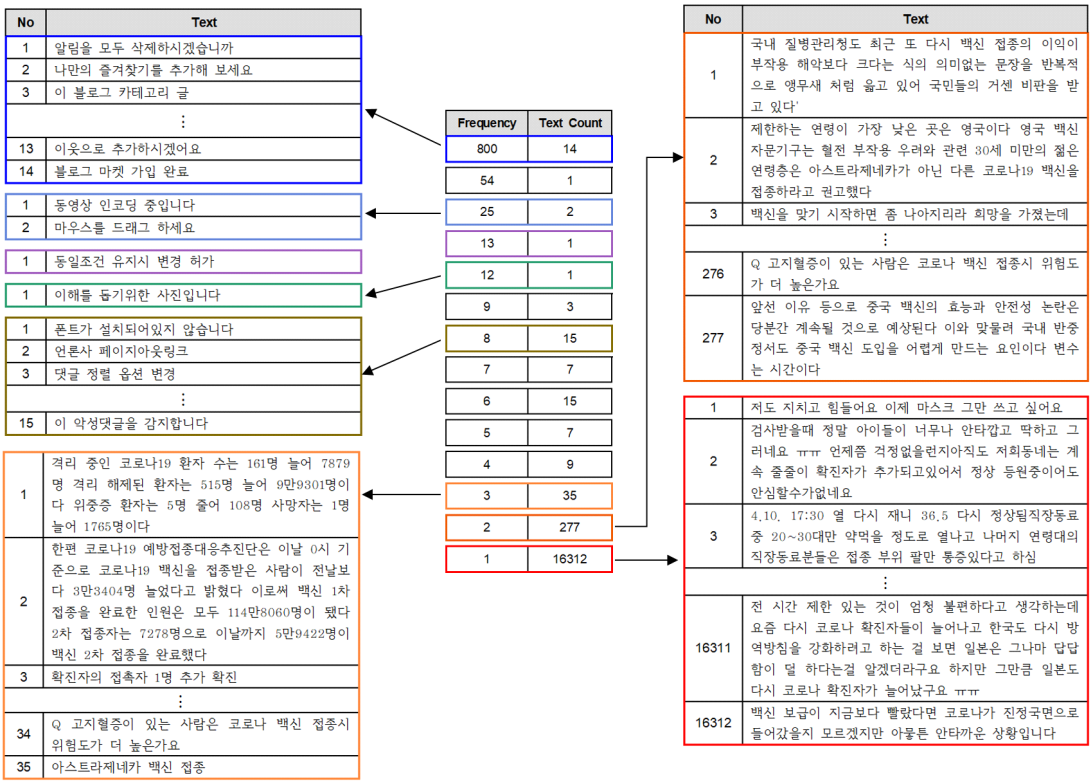


Fig. 2 Text count and text content by frequency

본 연구에서 텍스트의 출현 빈도를 기반으로 수집한 웹 페이지로부터 본문을 추출하는 절차는 그림 3과 같다. 먼저 웹 크롤러가 수집한 웹 페이지에서 계산량을 줄이기 위해 불필요한 태그를 제거할 필요가 있다[8]. 본 연구에서는 'head', 'footer', 'header', 'script', 'style', 'link', 'iframe', 'a', 'em'과 같이 본문 내용과 관련성이 적고, 주로 광고나 다른 글로 연결되는 태그를 제거하였다.

다음 단계에서는 불필요한 태그가 제거된 웹페이지에서 텍스트만을 추출하여 텍스트 출현 빈도를 포함하는 텍스트 블록(block)을 생성한다. 수많은 태그와 텍스트가 복잡하게 얽힌 웹 페이지에서 텍스트만을 추출하기 위해서는 그림 4와 같이 DOM 트리를 생성하여 웹 페이지를 분석하였다. 텍스트는 DOM 트리의 최하위 노드인 텍스트 노드에 포함되어 있으므로, 이러한 특성을 이용하면 DOM 트리로부터 텍스트를 어렵지 않게 추출할 수 있다. 따라서 본 연구에서는 파이썬의 html5lib 모듈을 이용하여 DOM 트리를 형성하고, 텍스트 노드에서 텍스트를 추출하여 텍스트 출현 빈도(text_block_freq)를 포함하는 텍스트 블록을 생성하였다. 그림 4와 같이 web_doc.html의 DOM 트리를 생성하면 총 7개의 텍스트 노드가 생성되며, web_doc.html은 각각의 텍스트 노드에서 텍스트를 추출하여 이에 대응하는 7개의 텍스트 블록으로 표현될 수 있다. 텍스트의 출현 빈도는 수집된 모든 웹 페이지에서 동일한 텍스트를 가진 텍스트 블록의 수를 나타내며, 텍스트 블록 생성 초기에는 0을 기본값으로 갖는다.

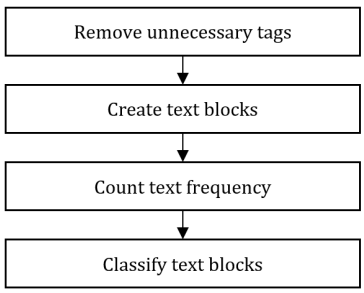
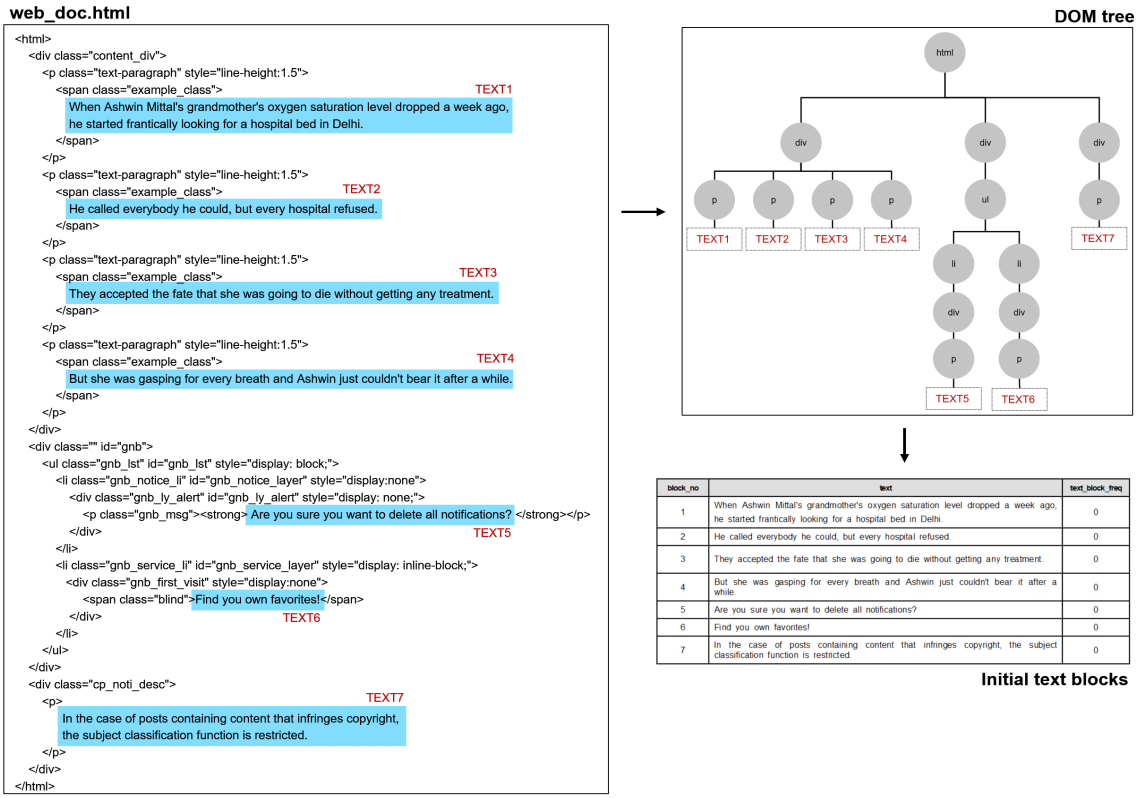


Fig. 3 Text extraction procedure based on the frequency of text appearance



block_no	text	text_block_freq
1	When Ashwin Mittal's grandmother's oxygen saturation level dropped a week ago, he started frantically looking for a hospital bed in Delhi.	0
2	He called everybody he could, but every hospital refused.	0
3	They accepted the fate that she was going to die without getting any treatment.	0
4	But she was gasping for every breath and Ashwin just couldn't bear it after a while.	0
5	Are you sure you want to delete all notifications?	0
6	Find you own favorites!	0
7	In the case of posts containing content that infringes copyright, the subject classification function is restricted.	0

Fig. 4 DOM tree structure and text block representation of web document

다음으로, 수집된 모든 웹 페이지에 대한 텍스트 블록의 생성이 완료되면, 추출한 모든 텍스트의 출현 빈도를 누적 집계하여 각 텍스트의 출현 빈도를 결정한다. 만약 그림 4의 6번 텍스트 블록과 동일한 텍스트를 포함한 9개의 다른 텍스트 블록이 출현하면 해당 텍스트 블록의 출현 빈도는 모두 10이 되며, 이러한 과정을 반복하여 모든 텍스트 블록의 출현 빈도를 결정한다.

마지막 단계에서는 각 텍스트 블록이 본문인지 아닌지를 분류한다. 본 연구에서는 본문 여부를 텍스트 블록의 출현 빈도의 순위를 기반으로 결정하였으며, 임계값(threshold) 이하의 순위를 갖는 텍스트 블록을 본문으로 분류하였다. 이때 출현 빈도가 낮을수록 낮은 숫자 즉, 높은 순위가 부여된다. 예를 들어, 3개 텍스트 블록의 출현 빈도가 각각 1, 15, 100인 경우, 각 텍스트 블록에 부여되는 출현 빈도 순위는 각각 1, 2, 3이 된다. 이때 어떤 빈도까지를 본문으로 분류하느냐에 따라 본문 분류의 정확도(accuracy)가 달라진다. 따라서 출현 빈도의 임계값을 결정할 필요가 있다.

본 연구에서는 임계값을 결정하기 위해 수집한 웹 페이지의 수를 100개에서 800개까지 100개 단위로 구분하여, 출현 빈도의 순위(frequency rank)에 따른 정확도의 변화를 그림 5와 같이 분석하였다. 출현 빈도의 순위는 출현 빈도가 가장 낮은 것을 1로 표현하였다. 그림 5는 수집된 웹 페이지의 표본이 달라져도 임계값을 동일하게 지정할 수 있다는 것을 보여준다. 예를 들어, 100개의 웹 페이지 표본에서 추출된 총 6개의 출현 빈도 중 5번째 등수의 출현 빈도를 갖는 텍스트 블록을 본문으로 분류하였을 때 정확도가 급격하게 낮아졌고, 500개의 웹 페이지 표본에서는 총 11개의 출현 빈도 중 10번째 등수의 출현 빈도를 갖는 텍스트 블록을 본문으로 분류하였을 때 정확도가 급격하게 낮아졌다. 또한 800개의 웹 페이지 표본에서는 총 16개의 출현 빈도 중 15번째 등수의 출현 빈도를 갖는 텍스트 블록을 본문으로 분류하였을 때 정확도가 급격하게 낮아졌다. 이처럼 다른 표본에서도 마찬가지로 전체 k개의 출현 빈도 중 $k-1$번째 등수의 출현 빈도를 갖는 텍스트 블록을 본문으로 분

류했을 때 동일한 현상이 나타나는 것을 확인할 수 있다. 따라서 본 연구에서는 수집한 웹 페이지에서 추출된 k개의 출현 빈도 중 $k-2$번째 등수 이하의 출현 빈도를 갖는 모든 텍스트 블록을 본문으로 분류하였다.

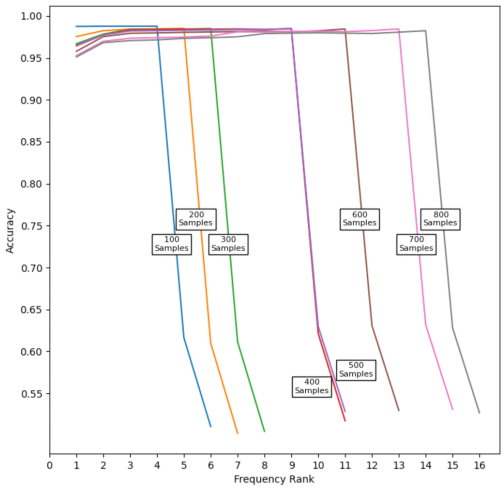


Fig. 5 Change of accuracy according to frequency ranking

Ⅳ. 성능 분석

4.1. 실험 데이터

본 연구에서 성능 분석을 위한 검증 데이터로는 네이버 블로그에서 '코로나 백신'을 키워드로 수집한 2,800개의 웹 페이지를 이용하였다. 네이버 블로그는 블로그의 외형을 이용자가 쉽게 변경할 수 있는 다양한 템플릿을 제공하기 때문에 이용자마다 블로그의 태그 구성과 스타일 속성이 다양하게 나타난다. 네이버 블로그의 이러한 특성은 웹 페이지의 다양한 구조에 상관없이 텍스트 추출 자동화의 성능 시험에 적절한 데이터라고 할 수 있다.

수집된 웹 페이지 중 학습 데이터로 이용하기 위해 무작위로 선정한 800개의 웹 페이지로부터 31,608개의 텍스트 블록을 생성하고, 레이블링 작업을 통해 16,652개(52.6%)의 텍스트 블록을 본문 블록으로 분류하였다. 나머지 2,000개의 웹 페이지에서는 84,061개의 텍스트 블록을 생성하고, 46,079개(55.5%)의 본문 블록을 분류하여 성능 평가 데이터로 이용하였다.

4.2. 성능평가 방법

본 논문에서 제안한 텍스트 빈도 분석을 이용한 텍스트 추출 방법과 단어/링크 밀도를 이용한 방법[6] 및 시퀀스 레이블링을 이용한 방법[10]의 성능 비교를 위해, 각각의 텍스트 추출 방식으로 텍스트 블록을 본문/비본문으로 분류한 후 실제 데이터와 비교하여 표 1과 같이 TP, FN, FP, TN의 네 가지 유형으로 분류하였다[11].

TP는 분류 모듈이 본문으로 분류한 텍스트 블록이 실제 본문인 경우의 수, FN은 분류 모듈이 비본문으로 분류한 텍스트 블록이 실제 본문인 경우의 수를 의미한다. 또한 FP는 분류 모듈이 본문으로 분류한 텍스트 블록이 실제 비본문인 경우의 수, TN은 분류 모듈이 비본문으로 분류한 텍스트 블록이 실제 비본문인 경우의 수를 의미한다.

Table. 1 F-measure confusion matrix

		Actual Class	
		Content	No Content
Predict Class	Content	True Positive (TP)	False Positive (FP)
	No Content	False Negative (FN)	True Negative (TN)

표 1의 분류 방식을 이용하여 텍스트 분류의 정확도 평가를 위해서 분류 문제의 평가에서 가장 광범위 하게 사용되는 $F-Measure$ 값을 식 (1)과 같이 측정한다[6].

$$F-Measure = \frac{2\times \mathrm{Precision}\times Recall}{\mathrm{Precision}\times Recall} \quad (1)$$

$$\text{재현율}(Recall) = \frac{TP}{TP+FN} \quad (2)$$

$$\text{정밀도}(\mathrm{Precision}) = \frac{TP}{TP+FP} \quad (3)$$

식 (1)에서 재현율(Recall)은 실제로 본문인 텍스트 중 분류 모듈이 본문이라고 분류한 텍스트의 비율로서 실제 텍스트에서의 본문 비중을 나타내고, 식 (2)와 같이 구할 수 있다. 또한 정밀도(Precision)는 분류 모듈이 본문으로 분류한 텍스트 중 실제로 본문인 텍스트의 비율로서 모듈이 분류한 텍스트에서의 본문 비중을 나타내고, 식 (3)과 같이 구할 수 있다. 재현율과 정밀도는 성

능을 평가하는 관점이 다르기 때문에, 이들의 조화 평균인 $F-Measure$ 값을 이용하여 실제 텍스트의 관점과 분류 모듈의 관점에서 상호보완적으로 성능을 평가할 필요가 있다.

4.3. 단어/링크 밀도 방법

단어/링크 밀도를 이용한 텍스트 분류 방법은 본문을 분류하기 위해 식 (4)와 식 (5)처럼 텍스트 블록 TB_i에 대한 단어 밀도 $D_{word}(TB_i)$와 링크 밀도 $D_{link}(TB_i)$를 계산해야 한다[5]. 여기서 $WordCount(TB_i)$와 $SentenceCount(TB_i)$는 각각 텍스트 블록에 포함된 단어와 문장 개수를 의미하며, $LinkCount(TB_i)$는 하이퍼링크가 포함된 단어의 개수를 의미한다.

$$D_{word}(TB_i) = \frac{WordCount(TB_i)}{SentenceCount(TB_i)} \quad (4)$$

$$D_{link}(TB_i) = \frac{LinkCount(TB_i)}{WordCount(TB_i)} \quad (5)$$

일반적으로 텍스트를 구성하는 각각의 문장은 마침표(.)나 줄바꿈(\n) 등의 구분자에 의해 구분되지만[12], 본 연구에서 수집한 블로그와 같이 자유분방하게 작성되는 텍스트의 경우 구분자에 의한 문장 구분이 생략되는 경우가 많으므로 문장의 개수를 정확히 알 수 없어 단어 밀도를 정확히 계산할 수 없는 경우가 많다.

$$TextCount_{mean} = \frac{\sum_{i=1}^{N} TextCount(TB_i)}{N} \quad (6)$$

$$SentenceCount(TB_i) = \frac{TextCount(TB_i)}{TextCount_{mean}} \quad (7)$$

따라서 본 연구에서는 웹 페이지로부터 추출한 N개의 텍스트 블록에 대하여 식 (7)과 같이 각 텍스트 블록의 글자 수 $TextCount(TB_i)$를 식 (6)과 같이 구한 텍스트 블록의 평균 글자 수인 $TextCount_{mean}$로 나누어 각 텍스트 블록의 문장 개수 $SentenceCount(TB_i)$를 계산하였다.

단어/링크 밀도를 이용한 실험 데이터 분류 결과는 표 2와 같다. 총 84,061개의 텍스트 블록 중 TP와 TN이 각각 45,772개와 92개로 분류되었다.

Table. 2 Text Extraction of Word Density method

		Actual Class	
		True	False
Predict Class	True	45,772	37,260
	False	937	92

4.4. 시퀀스 레이블링 방법

시퀀스 레이블링을 이용한 텍스트 분류 방법은 텍스트 블록의 순서 정보를 텍스트 추출에 반영하기 위하여 그림 6과 같이 양방향 LSTM 구조의 지도학습 방식을 이용한다[10].

본 연구에서는 신경망 학습을 위해 텍스트 블록의 상위 태그 정보와 텍스트의 단어 정보를 합쳐 146차원의 입력 데이터를 구성하였다. 이 중 상위 태그 정보의 크기는 학습 데이터에서 생성한 텍스트 블록이 가지는 상위 태그 개수의 최대치인 62로, 단어 정보의 크기는 텍스트 블록에서 형태소 분석을 통해 추출된 단어 개수의 최대치인84로 구성하였다. 신경망 구현을 위해 오픈소스 라이브러리인 케라스(Keras)를 이용하였으며, LSTM 계층의 출력값을 텍스트 블록이 본문일 확률을 0과 1 사이 값으로 변환하기 위해 시그모이드(sigmoid) 함수를 사용하였다. 또한 모델 학습을 위해 분류의 정확도와 학습 시간을 고려하여 은닉층을 256개로 설정하였으며, 손실함수로는 이진 교차 엔트로피(Binary Cross Entropy)를 이용하였다.

Table. 3 Text Extraction of Sequence Labeling method

		Actual Class	
		True	False
Predict Class	True	46,204	1,066
	False	505	36,286

시퀀스 레이블링에 의한 텍스트 분류 결과는 표 3과 같이 전체 84,061개의 텍스트 블록 중 TP와 TN으로 분

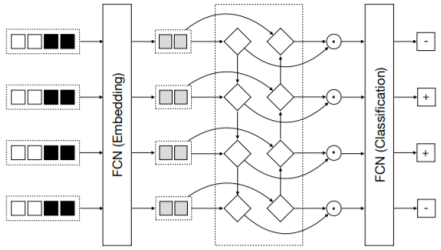


Fig. 6 BoilerNet architecture with bidirectional LSTM layer

류된 것이 각각 46,2040개와 36,286개로 나타났다.

4.5. 텍스트 빈도 분석 방법

본 논문에서 제안한 텍스트 빈도 분석 방법의 성능 평가를 위해, 84,061개 텍스트 블록의 출현 빈도를 집계한 결과 총 23개의 출현 빈도($k=23$)가 나타났으며, 이 가운데 크기가 $k-2$번째, 즉 21번째로 큰 출현 빈도인 95를 분류 임계값으로 설정하여, 해당 임계값 이하의 출현 빈도를 갖는 텍스트 블록을 본문으로 분류하였다.

Table. 4 Text Extraction of Proposed method

		Actual Class	
		True	False
Predict Class	True	46,614	1,352
	False	95	36,000

텍스트 빈도 분석 방법에 의한 텍스트 분류 결과는 표 4와 같이 전체 84,061개의 텍스트 블록 중 TP와 TN으로 분류된 경우는 각각 46,614개와 36,000개로 나타났다.

4.6. 성능평가 결과 비교

2,000개의 웹 페이지로부터 추출한 84,061개의 텍스트 블록에 대한 성능평가 결과는 표 5와 같다.

Table. 5 Comparison of the method performance

	Word Density(%)	Sequence Labeling(%)	Proposed Method(%)
Recall	98.0	98.9	99.8
Precision	55.1	97.7	97.2
F-Measure	70.6	98.3	98.5

단어/링크 밀도 방법의 경우에는 텍스트 길이 정보에 의존하여 텍스트를 분류하기 때문에 짧은 텍스트를 본문으로 분류하지 못해 전체적으로 성능이 낮게 측정되었다. 이에 반해 시퀀스 레이블링 방법에서는 인접한 태그와 단어 정보를 특징으로 학습된 모델을 이용하여 텍스트를 분류하기 때문에 재현율과 정밀도 모두 높게 나타났다. 본 논문에서 제안한 방법은 정밀도의 경우 시퀀스 레이블링 방식보다 0.5% 낮게 나타났지만 재현율은 0.9% 높게 나타났고, $F-Measure$ 값은 세 가지 방법 중 가장 높게 나타났다.

결과적으로 본 논문에서 제안한 방법은 기존의 방법들과는 달리, 인접 태그나 단어 정보 등의 특징을 추출하지 않고 간단히 텍스트의 출현 빈도 분석만으로 우수한 본문 분류 성능을 보임을 알 수 있다.

V. 결 론

본 연구에서는 다양한 HTML태그와 스타일 속성을 가지는 웹 페이지에서 텍스트 추출을 자동화하기 위해 웹 페이지에 나타나는 텍스트의 빈도 분석을 이용하는 방법을 제안하였으며, 기존의 단어/링크 밀도 및 시퀀스 레이블링을 이용하는 방법보다 정확도가 향상된 것을 확인하였다. 본 논문에서 제안한 방법에서는 웹 페이지에 출현하는 텍스트 빈도 외에 HTML 태그나 스타일 속성, 단어 밀도, 인접한 태그 등의 특징은 이용하지 않았다. 따라서 텍스트를 포함하는 태그와 스타일 속성을 알아내기 위해 웹 페이지를 분석하지 않아도 되며, 인접한 태그나 단어 정보를 추출하기 위한 별도의 전처리 과정을 거치지 않아도 된다는 장점이 있다. 이러한 장점은 빅데이터 수집 시 웹 페이지에서 텍스트를 추출하는 작업을 쉽게 자동화하여 텍스트 수집 과정의 효율성을 크게 향상시킬 것으로 기대된다. 향후 블로그로 한정된 수집채널을 SNS와 학술 논문, 뉴스 등으로 확장하여 텍스트 빈도 분석 방법의 범용성을 확보하고, 이를 기반으로 웹 크롤링 시스템을 구현할 계획이다.

ACKNOWLEDGEMENT

This paper was supported by the Education and Research Promotion Program of KOREATECH in 2021.

References

[1] H. S. Bang and H. S. Moon, “A study on the methodology to express the main topics of text in time series using text mining,” *Journal of the Korean Data And Information Science Society*, vol. 30, no. 6, pp. 1259-1276, 2019.

[2] S. R. Lee and E. J. Choi, “Comparison of responses to issues in SNS and Traditional Media using Text Mining - Focusing

on the Termination of Korea-Japan General Security of Military Information Agreement(GSOMIA)," *Journal of Digital Convergence*, vol. 18, no. 2, pp. 277-284, 2020.

[3] J. H. Lee, H. J. Seon, and H. J. Lee, "Positioning of Smart Speakers by Applying Text Mining to Consumer Reviews: Focusing on Artificial Intelligence Factors," *Knowledge Management Review*, vol. 21, no. 1, pp. 197-210, 2020.

[4] M. G. Cha and J. Y. Lee, "A Study on Spatial Co-experience through Social Data," *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, vol. 7, no. 6, pp. 851-859, 2017.

[5] K. W. Cho and Y. W. Woo, "Topic Modeling on Research Trends of Industry 4.0 Using Text Mining," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 7, pp. 764-770, 2019.

[6] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proceedings of the third ACM international conference on Web Search and Data Mining (WSDM)*, New York: NY, pp. 441-450, 2010.

[7] W. M. Song, W. S. Kim, and M. W. Kim, "Contents Extraction from HTML Documents using Text Block Context," *Journal of KISS : Software and Applications*, vol. 40, no. 3, pp. 155-163, 2013.

[8] H. G. Jeon and C. Koh, "Text Extraction Algorithm using the HTML Logical Structure Analysis," *Journal of Digital Contents Society*, vol. 16, no. 3, pp. 445-455, 2015.

[9] T. Vogels, O. E. Ganea, and C. Eickhoff, "Web2text: Deep structured boilerplate removal," in *Proceedings of the 40th European Conference on Information Retrieval*, pp. 167-179, 2018.

[10] J. Leonhardt, A. Anand, and M. Khosla, "Boilerplate Removal using a Neural Sequence Labeling Model," in *Companion Proceedings of the Web Conference 2020 (WWW '20)*, New York: NY, pp. 226-229, 2020.

[11] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168-192, 2021.

[12] S. H. Kim and H. J. Kim, "Logistic Regression Ensemble Method for Extracting Significant Information from Social Texts," *KIPS Transactions on Software and Data Engineering*, vol. 6, no. 5, pp. 279-284, 2017.

김진환(Jin-Hwan Kim)

2016년 2월 : 한국기술교육대학교 컴퓨터공학부 졸업(학사)
2019년~현재 : 한국기술교육대학교 컴퓨터공학과 석사과정
※관심분야 : 빅데이터, 텍스트마이닝, 웹 크롤링, 기계학습

김은경(Eun-Gyung Kim)

1983년 2월 : 숙명여자대학교 물리학과 졸업
1986년 2월 : 중앙대학교 전자계산학과 석사
1991년 2월 : 중앙대학교 컴퓨터공학과 박사
1992년 3월~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수
※관심분야 : 빅데이터 분석, 딥러닝, 트리즈 등