Journal of Digital Contents Society Vol. 16 No. 3 Jun. 2015(pp. 445-455)
http://dx.doi.org/10.9728/dcs.2015.16.3.445

HTML 논리적 구조분석을 통한 본문추출 알고리즘

전현지*, 고 찬**

요 약

인터넷과 컴퓨터 기술이 발전함에 따라 정보의 양이 폭발적으로 증가하였으며, 이로 인해 다양한 웹 저작 도구 및 새로운 웹 표준의 출현과 웹에 대한 접근성이 보다 편리해지면서 매우 다양한 종류의 웹 콘텐츠들이 아주 빠르게 생산되고 있다. 하지만 웹 문서는 여러 블록으로 나누어 다양한 주제를 담아내고 있으며, 각각의 블록들이 서로 연관성이 없는 주제를 다루는 경우가 많을 뿐만 아니라 네비게이션, 단순한 장식물, 광고, 저작권 정보 등과 같이 콘텐츠로 볼 수 없는 블록들도 존재한다. 이러한 문제를 해결하기 위해 HTML 웹 문서의 정확한 본문영역만을 추출하여 사용자 요구조건을 충족하고 효과적으로 정보를 학습할 수 있도록 하며, 추후에는 문서를 체계적으로 관리할 수 있게 최적화된 웹 검색 시스템으로서의 재구성 방법을 제안하고자 한다.

키워드 : HTML, Data Mining, Text Extraction, HTML Structure

Text Extraction Algorithm using the HTML Logical Structure Analysis

Hyun-Gee Jeon*, Chan KOH**

Abstract

According as internet and computer technology develops, the amount of information has increased exponentially, arising from a variety of web authoring tools and is a new web standard of appearance and a wide variety of web content accessibility as more convenient for the web are produced very quickly. However, web documents are put out on a variety of topics divided into some blocks where each of the blocks are dealing with a topic unrelated to one another as well as you can not see with contents such as many navigations, simple decorations, advertisements, copyright. Extract only the exact area of the web document body to solve this problem and to meet user requirements, and to study the effective information. Later on, as the reconstruction method, we propose a web search system can be optimized systematically manage documents.

Keywords : HTML, Data Mining, Text Extraction, HTML Structure

1. 서론

1.1 서론

2014년 7월 만 3세 이상 인구의 인터넷 이용률은 83.6%로 전년대비 1.5%p 증가한 수준이며, 인터넷 이용자수는 41,118천명으로 전년대비 1,038명 증가로 나타났다. 이처럼 우리나라 경제활동 인구의 99%는 인터넷을 사용하고, 국내 스마트폰 가입자 수가 4천만 명을 넘어섰다는 결과가 발표되었다. 주된 인터넷 이용 용도로 상품 및 서비스 정보, 일반 웹서핑 등 자료 및 정보

※ Corresponding Author : Chan KOH
Received : March 15, 2015
Revised : June 24, 2015
Accepted : June 30, 2015
* Seoul National University of Science & Technology
Tel: +82-2-970-6707, Fax: +82-2-970-9441
email: guswlna@naver.com
** Seoul National University of Science & Technology
Tel: +82-2-970-6705, Fax: +82-2-970-9441
email: chankoh@seoultech.ac.kr
▣ 이 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.

접근/검색이 91.1%로 가장 높게 나왔으며 다음으로 이메일, SNS 등 커뮤니케이션이 89.8%로 뒤를 이었다고 한다[1]. 오프라인과 온라인의 경계가 무너진 지금, 웹이 일상생활에서 보편적으로 사용됨에 따라 웹 문서의 양이 급증하고 있으며, 사용자들은 보다 쉽고 빠르게 접근할 수 있게 된 것이다. 또한 웹 저작자가 다양한 내용을 표현할 수 있도록 하는 새로운 표준과 저작 도구들이 출현하면서 사용자를 위한 웹문서를 생산하는 기업 뿐 아니라, 개인 사용자들도 블로그, 온라인상에서 사용자들이 특정한 관심이나 활동을 공유하는 서비스인 SNS (Social Networking Service)를 핸드폰, 태블릿 PC 등을 이용하여 'when' or 'where' 으로 신속하게 확인할 수 있는 뉴스기사와 같은 서비스를 통해 웹 문서 생산에 가세함으로써 특정 형식에 얽매이지 않는 다양하고 자유로운 형식으로 작성된 웹 문서들이 빠르게 생산되고 있다. 웹 문서의 경우 일반 문서와는 다르게 내용이나 형식이 자유로울 뿐 아니라 빠르게 변화하고 있기 때문에 다양한 종류의 웹 문서들이 기하급수적으로 증가하는 웹 서비스 환경에서 사용자들이 요구하는 것을 빠르고 정확하게 제공하기 위해서는 웹 문서의 내용을 구분, 분석하고 문서에서 말하고자 하는 바를 이해할 수 있어야 한다.

오늘날의 웹은 하나의 전체구조를 여러 블록으로 나눠 개별적인 주제를 담아내고 있다. 각 블록은 서로 연관성이 없는 주제를 다루는 경우가 많을 뿐만 아니라 메뉴, 배너, 광고, 저작권 정보와 같이 콘텐츠로 볼 수 없는 블록들도 존재한다. 이로 인해 사용자는 정보의 홍수 속에서 살아가고 있는 반면, 넘쳐나는 정보 속에서 유용한 정보를 찾는 것이 더 어려워졌다. 따라서 사용자가 효과적으로 정보를 학습함에 최적화 된 웹 문서로 재구성하는 기술이 요구되는 실정이다. 최근 이러한 이유로 웹 문서 내에 포함된 광고, 메뉴, 댓글 등과 같은 클러터(Clutter)들이 포함된 영역을 자동으로 구분하는 웹 영역 추출에 관련한 연구가 활발히 진행되고 있으며, 특히 정보를 표현하는 가장 중요한 영역인 본문영역의 자동 추출에 관한 연구들이 활발하게 진행되고 있다. 초기 연구들에서는 HTML 형태의 웹 문서를 DOM(Document Object Model) Tree 형태로 표현하고, 표현 된 DOM Tree Node를 분석하는 HTML 문서의 구조적 특징에 따른 방법을 시도하였다. [2,3] 그리고 최근에는 HTML 문서의 구조적 특징에 대한 제한점을 극복하고 내용기반 본문을 추출하고자 하는 웹 문서의 본문에 출현하는 단어나 내용을 이용한 언어 모델을 학습하고 학습된 언어 모델을 이용하여 본문 영역을 구분해 내는 방법들[4,5]이나 웹 문서로서의 HTML 문서 내 태그 및 하이퍼링크의 특징 등을 이용한 본문추출 방법들[6-8]이 제안되기도 하였다.

2. 관련연구

문서의 자동분류 방법은 구성된 색인어 집합과 유사한 문서를 동일한 그룹으로 분류한다. 많은 양의 문서를 효과적으로 관리하고, 키워드 검색을 통한 정확한 결과를 나타낼 수 있게 하는 동시에 방대한 양의 작업을 감소시키는데 그 목적이 있다.

2.1 웹 문서 본문추출, 분류방법

웹 문서로부터 본문을 추출하기 위해 가장 많이 사용되는 방법은 HTML 문서의 구조적 특징을 이용한 DOM 트리 구조를 이용하는 것이다[2,3]. [2]에서는 시각적 블록에 기반 한 페이지 구분 및 본문 블록 추출을 위해 HTML문서를 DOM트리로 표현하고, 트리내의 각 노드별로 가로, 세로 크기 및 배경 색상과 문서 내에서의 절대적 출현 위치 정보 등을 노드의 특징으로 구성하였다. 이 후, 이와 같은 특징과 노드들의 문서 내 출현 밀집도 등을 이용하여 그룹화 하고, 다시 한 번 그룹의 특징 정보를 이용하여 그룹별로 본문 여부를 결정하였다. 이와 같은 연구들은 기계 학습 기법을 이용하여 템플릿 정보 없이 자동으로 본문 영역을 추출하는 것에 대한 가능성을 검증하였으나, 단순히 화면에 보이는 영역 정보만을 이용하거나 트리 구조로 표현되는 문서의 구조적 특징만을 이용함으로서 광고, 메뉴, 댓글등과 같이 본문과 관계없는 내용들이 웹 문서 내에서 더 큰 영역을 차지하는 경우 본문을 제대로 분류해내지 못하는 단점이 있다.

2.2 텍스트 블록의 단어/링크 밀도에 의한 본문 분류

웹 문서의 본문 추출을 위한 공개시스템으로 현재까지 가장 성능이 좋다고 알려진 방법은 독일의 L3C 연구소에서 제안한 boilerpipe이다[6,11]. 이 방법은 HTML 문서를 경험적인 규칙을 이용하여 텍스트 블록들로 구분하는 단계, 각 텍스트 블록별로 블록 내 단어수와 링크가 삽입된 단어의 수에 따라 단어 밀도와 링크 밀도를 계산하여 특징을 추출하는 단계, 그리고 이전/이후 에 출현한 텍스트 블록의 정보(단어/링크 밀도)를 현재 텍스트 블록의 특징으로 함께 고려하여 학습하고, 텍스트 블록별로 본문 여부를 분류하는 단계로 이루어진다.

HTML 문서를 텍스트 블록들로 구분하는 첫 단계에서는 HTML 태그를 이용한다. 즉, HTML 문서 내에 출현한 모든 태그가 구분자가 되어 이들 구분자로 구분되는 텍스트 단위를 각각의 블록으로 설정한다. 이렇게 텍스트 블록이 구분되면, 다음 식(1)에 따라 각 텍스트 블록 TB_i에 대한 단어 밀도($D_{WORD}(TB_i)$), 링크밀도 ($D_{LINK}(TB_i)$)를 계산한다. 식 (1), (2)에서 $Word(TB_i)$, $Sentence(TB_i)$ $LinkedWord(TB_i)$, 는 각각 텍스트 블록 TB_i내의 모든 단어 집합, 문장 집합, ‘a’태그에 의해 하이퍼링크가 삽입된 단어집합이다. [11]에서는 텍스트 블록 TB_i에서 문장을 구분 할 때, 단어의 개수를 이용하였으며, 실험적으로 매 80개의 단어까지를 하나의 개별적인 문장으로 결정하여 사용하였다.

$$D_{WORD}(TB_i) = \frac{|Word(TB_i)|}{|Sentence(TB_i)|} \quad (1)$$

$$D_{LINK}(TB_i) = \frac{|LinkedWord(TB_i)|}{|Word(TB_i)|} \quad (2)$$

위의 방법은 텍스트 블록의 링크/단어 밀도만으로도 본문을 분류해냄으로서 짧은 시간 내에 추출 가능한 소수의 특징만으로 본문을 정확하게 분류 하였다는데 그 의의가 있다. 그러나 정확도 측정 시에 본문 및 비 본문에 해당하는 모든 분류 결과에 대한 정확도를 합산 계산함으로서 전체 분류 정확도가 상대적으로 높은 비율을 차지하는 비본문의 분류 정확도에 가깝게 결정되어 분류하고자 하는 본문에 대한 추출정확도 만을 정확하게 검증하였다고 보기가 어렵다[13].

2.3 텍스트 블록 주변의 문맥정보를 이용한 추출

HTML 형식으로 작성된 웹 문서로부터 기계학습 기법을 이용하여 정확하게 본문 내용을 추출하기 위하여 기존에 제안, 검증된 특징인 텍스트 블록 내의 단어/링크 밀도 이외에 텍스트 블록 주변의 HTML 태그 분포 정보 및 이웃 텍스트 블록의 특징 정보를 문맥 정보로 추출하여 함께 이용하는 방법을 제안한다[14]. 먼저 텍스트 블록 주변의 태그분포를 분석, 이때 태그의 트리 구조를 기반으로 그 내용이 구성되며, 특정 텍스트 블록의 부모 태그(트리 구조로 표현되었을 때 해당 텍스트 블록을 포함하는 바로 상위의 HTML 태그 정보)를 해당 텍스트 블록의 본문 여부 판단을 위한 특징 정보로 이용한다. 그리고 난 후 웹 문서의 본문 내용을 보다 정확하게 추출하기 위한 두 번째 특징 정보로서 이웃 텍스트 블록의 정보 및 이웃 텍스트 블록과의 거리를 나타낸다. 먼저 이웃 텍스트 블록과의 거리는 태그 단위로 줄이 구분되도록 전처리 된 HTML 파일에서 텍스트 블록 사이의 줄 간격으로 정의하였고, 두 번째로 이웃 텍스트 블록의 특징 정보로는 이전/이후에 위치한 각각 2개씩의 근접 텍스트 블록의 특징 정보를 이용한다. 특정 텍스트 블록의 특징 정보에 이전/이후의 텍스트 블록 특징 정보를 함께 포함함으로써 기계학습을 이용한 자동 분류 모델 학습 시 전후에 위치한 텍스트 블록의 특징을 반영하여 모델을 학습할 수 있도록 한다. 마지막 분류 학습을 위한 특징과 분류 알고리즘으로 하나의 텍스트 블록에 대해서는 기존에 [6]에서 제안된 텍스트 블록 내 단어 밀도, 링크 밀도와 함께 텍스트 블록 주변 정보인 텍스트 블록의 부모 태그, 이전 텍스트 블록과의 거리, 다음 텍스트 블록과의 거리 등 총 5개의 특징이 추출된다. 최종적으로 특정 텍스트 블록의 특징 정보는 자기 자신, 이전에 출현한 최근 접 2개의 텍스트 블록, 이후에 출현한 최근 접 2개의 텍스트 블록 등 총 5개 텍스트 블록의 각각의 특징 정보를 포함한 25개

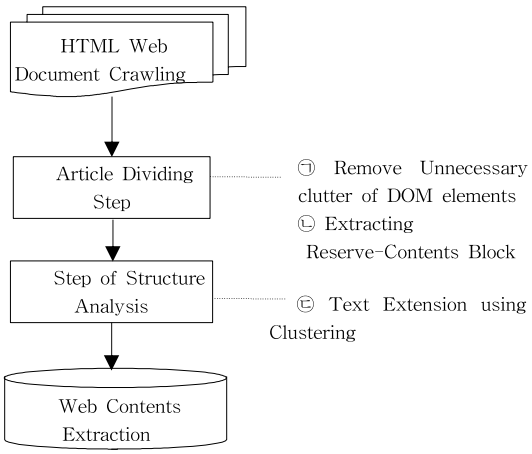
특징 정보로 구성된다.

단어 및 링크 밀도에 더하여 문서 내 텍스트 블록 주변의 태그 분포 정보와 이웃에 위치한 주변 텍스트 블록의 정보를 함께 이용하는 방법을 제안한 본 연구에서는 실험을 통하여 기존에 가장 좋다고 알려진 시스템[6]에 비해 공개된 데이터에 대해서는 약 3%, 직접 수집한 데이터에 대해서는 약 19%의 성능을 향상시킴으로서 제안하는 방법의 타당성을 검증하였으며, 서로 다른 두 데이터에서 유사한 성능을 보였다.

3. 제안하는 본문추출 알고리즘

크롤러(Crawler) 혹은 사용자에 의해 추출된 웹 페이지를 적절히 인덱싱(Indexing)하기 위해서는 해당 문서의 본문영역을 추출(Filtering) 하는 것이 정보의 정확도에 있어 매우 중요한 부분이다.

(그림 1) 제안된 논리적 본문영역 추출 방법



(Figure 1) The Proposed method extracts the logical text area

웹 문서 내 구조를 살펴보면 본문과 상관없는 클러터 부분이 전체 HTML구성 텍스트에서 40%이상을 차지한다. 물론 모든 페이지에서 공통적으로 등장한다면 단순한 확률분포 기반 인덱싱 만으로도 어느 정도의 검색과 무관한 데이터를 가려낼 수 있기는 하지만, 그 외의 다양한 용도로 사용될 수 있고, 검색 시스템의 방법론 등을 고려해 보았을 때 논리적으로 본문영역을 추출하는 것이 매우 중요한 기능임에는 틀림없다.

본 절에서 제안하는 방법은 영역구분단계와 구조분석단계로 나눠 순차적으로 본문을 분석하고자 한다. 영역구분단계에서 전처리와 후처리를 통해 불필요한 클러터를 제거한 후 Data Type의 길이정보와 키워드(색인어) 가중치를 이용한 정보를 동시에 만족하는 블록들에 대해서 후보 본문 블록으로 추출한다. 추출 된 후보 본문 블록들을 통해, 구조분석단계에서 후보 본문 블록(Standard Block)에 대한 전후 블록태그 분석 방법 등을 이용하여 최종적인 웹 문서 내 본문들을 찾아낸다. 본 연구에서 본문추출에 있어 2단계인 영역구분단계와 구조분석단계로 나누는 이유는 영역구분단계를 통계적인 방법이라 통칭하고, 구조분석단계를 지식기반 분석 방법이라 통칭하면 통계적인 방법의 장점인 기존에 다양한 형식의 Template Info가 있는 경우 빠른 시간 안에 분석이 가능하다는 점과 지식기반 분석 방법의 장점인 분석규칙을 만들어 사용하므로 규칙에 따라 분류될 문서들의 경우 높은 정확도를 나타낸다는 점이다. 결과적으로 각 단계를 통해 서로의 단점을 보완하고 장점을 부각시켜 본문영역 추출에 정확도 상승을 기대할 수 있는 것이다.

3.1 영역구분단계

3.1.1 전처리

계산량과 오류를 줄이기 위해 불필요한 DOM 요소들을 제거하는 과정이다. 웹 페이지 본문의 특징을 살펴보면, 본문 내에는 링크가 드물고, 비교적 장문의 텍스트를 포함하며, 본문의 레이아웃을 구성하는 태그가 비교적 적다는 사실을 알 수 있다. 그렇기 때문에 모든 문서에 공통적으로 포함되는 단어는 변별력이 없어 문서 분류에 영향을 주지 못하고, 키워드(색인어) 생성 시 성능저하를 유발하게 되므로 불용어로 간주하여 처리한다. 예를 들면 시작/종료태그, 광고, 댓글, 메일주소, 저작권 표시 등을 말한다. 전처리 과정은 첫째, 웹 사이트의 정보를 나타내는 <head>내<meta>,<style>,<script>,<link>,<title>과 같은 공통요소이다. 이러한 요소들은 웹 문서에서 charset(ANSI/UTF-8/Unicode), meta type

과 같은 중요한 정보를 표현하는데 사용하지만 본문요소와 관련이 적으므로, 하나의 문자열로 취급하여 제거한다. 둘째, 메뉴리스트에 대한 구조를 표현할 때 쓰이는 <ul>,<ol>,<li>,<dl>,<dt>,<dd> 요소이다. 리스트는 메뉴를 나타내는 네이게이션 바, 혹은 광고리스트를 표현할 때 주로 사용하며, 본문 내에는 일정 등을 나타내는 텍스트 정렬 요소로 사용되거나 혹은 포함되지 않는 불필요한 요소로 취급하여 제거한다. 셋째, 본문에서는 극히 소수만 발견되며 주로 광고나 다른 글로 연결되는 <a>와 <img>가 있다. <a> 의 경우 흥미로운 점은 본문 외 영역에서 본문 내 속한 단어의 개수보다 링크가 걸려있는 단어의 수가 더 많다는 점이다. 이 말은 즉 본문에 있는 단어들 중에 링크가 걸린 단어는 적은 반면 광고, 메뉴 등 클러터에 있는 단어들 중에 링크가 걸린 부분의 단어가 상대적으로 높다는 것을 의미한다. 이러한 결과를 보면 단순히 웹 문서 내에서 링크 요소들만 제거해도 상당수의 클러터를 걸러낼 수 있다는 것이다. 넷째, 텍스트를 꾸며주고 강조하는 역할인 <h1>~<h6>,, ,<textarea>,<font>,
 font 관련 표현 요소이다. font요소의 경우 HTML 레이아웃 구조의 영향을 미치는 요소가 아닌 단순히 문장을 여러 개로 나누거나 사용자로 하여금 텍스트의 외형적으로 강조 혹은 구분만을 의미하기 때문에 클러터로서 제거된다고 해도 본문영역에 영향이 적다. 그 외적인 요소로는 html 구조와 관련이 적은 기타 요소로서 <noscript>,<iframe>, <input>,<button>,<form>,<fieldset>,<object>, <param>,<label>,<select>,<option>,<xmp>,<address>,<hr> 가 있다. 대체로 위의 요소들의 경우 웹 문서의 기능적인 역할을 담당하고 있기 때문에 본문의 영역 즉, 텍스트 부분만을 추출하는 본 연구에서는 불용어로 간주하여 제거된다. 그림 2는 제안된 전처리 규칙에 대한 수도코드이다. 각각의 웹 페이지에 삽입되는 수도코드의 경우 현 단계에서 자동적으로 입력하는 것에 한계점이 발생함으로 연구자가 모든 데이터 세트에 대해서 직접 작성해야한다는 단점이 발생된다.

규칙을 통해 제거 된 문서 중에서 영역구분단계를 거쳐 후보 본문 블록으로 추출된다.

(그림 2) 전처리/후처리 계산과정을 나타내는 수도코드

```
<script type="text/javascript">
    $(document).ready(function(){
        var cl_head='meta,style,script,link,title';
        var cl_gnb='ul,li,dl,dt,dd';
        var cl_font='h1,h2,h3,h4,h5,h6,strong,em,textarea,font,br';
        var cl_other='noscript,iframe,input,button,form,fieldset,'+
        'object,param,label,select,option,xmp,address,hr';

        $(cl_head).remove();
        $(cl_gnb).remove();
        $('img').remove();
        $(document).find('a').remove();
        $(cl_font).remove();
        $(cl_other).remove();
    });
</script>
```

(Figure 2) The pre-treatment / post processing calculations indicate the main code

3.1.2 영역구분단계

영역구분 단계에서는 전처리 규칙을 통해 클러터를 제거한 문서들 중에서 Data Type의 길이정보와 키워드(색인어) 가중치를 이용한 결과정보의 값이 동시에 만족하는 블록에 대해 후보 본문 블록으로 추출한다.

먼저 Data Type의 길이정보를 이용한 블록 추출 방법으로 본문 요소 내에는 다른 요소들에 비해서 대량의 텍스트가 포함되어 있다는 사실에 근거하여 후보 본문 블록을 추출한다. 첫째, 링크가 없는 문자열 중 Length 값이 가장 큰 블록요소를 추출한다. 문단, 문장을 구성하는 Text Tag인 ,<p>, 중 가장 텍스트가 긴 요소를 말하며 이때, 웹 문서 내 <p>,요소는 요소와 텍스트를 감싸는 동일한 역할을 하기 때문에 실험 데이터의 통일성을 위하여 요소로 통일한다. 그림 3은 길이정보에 대한 계산과정을 나타내는 수도코드이다. 클러터가 제거된 HTML 문서에 String Type으로 이루어진 문자열 요소들에 대한 index 값을 기준으로 각 블록에 대한 문자열 Length 값을 일괄적으로 계산한다. 각각의 결과 값은 List 형태로 작성하여 요소 정보를 반환하면 된다. 하지만, 단순히 Data Type의 길이정보가 가장 높은 값을 갖는 요소를 후보 본문블록으로 추출하게 되면, 전체 본문 중에서 가장 긴 문장을 갖는 일부분(예를 들면, 비 본문 영역인 댓글정보가 본문보다 길 때)이 추출 될 경우가 발생한다.

(그림 3) Data Type의 길이정보 계산 과정 수도코드

```
<script type="text/javascript">
  //<![CDATA[
    function checkButton(){
      var index = "";
      var divcount = $( "div" ).length;
      var arr = [];
      var cnt = 0;
      var str = "";
       for (var index=0; index < divcount; index++){
         if(!$("div").eq(index).children().text()){
           str =$("div").eq(index).text();
           arr[cnt] = ["div ID ["+"div"+cnt+"]",
           " length ["+str.length+"]" , " str [" + str +"]" ];

           cnt++;
         }
         // else
         // {
         //    str="";
         // }
       }
        var divAnswer = $("#answer");
        var arrAnswer = "";
        for (var arrIndex=0; arrIndex<arr.length; arrIndex++){
          arrAnswer = arrAnswer + arr[arrIndex]+"</br>";
        }
          divAnswer.html(arrAnswer);
    }
  //]]>
</script>
```

(Figure 3) The length information in the data type calculations main code

이러한 문제점을 보완하기 위해 길이정보와 함께 특정단어가 문서 내에서 얼마나 자주 등장하는지를 나타내기 위한 일련의 과정으로 형태소 분석을 거친 블록들에 대해 각 키워드 스코어(빈도수)를 기준으로 가중치 값을 부여, Data Type의 길이정보를 이용한 결과 값과 동시에 만족하는 블록들에 대해 후보 본문 블록으로 추출한다.

키워드 (색인어) 가중치를 이용한 블록 추출이란 먼저 색인어의 경우 검색엔진을 이용하는 사용자가 어떠한 특정 키워드를 넣어 검색을 할 때 이전에 분류 되었던 자료 중 해당 키워드와 일치하는 것을 검색하게 된다. 여기서 DB안에 존재하는 키워드를 색인어라 부르며, 색인어는 엔진이 생성하는 경우도 있고, 해당 페이지에 올리는 사용자가 직접 지정하는 경우가 있다. 그렇기 때문에 웹 문서에서 어떠한 키워드를 중심키워드로 정하느냐에 따라서 본문영역 추출에 대한 정확도가 달라질 수 있다. 가중치를 이용한 블록 추출 과정은 첫째, 클러터를 제거한 블록 내 어휘들에 대한 형태소 분석을 진행한다. 형태소 분석은 뜻을 가진 가장 작은 말의 단위로서, 이 단계에서는 정보추출의 기본적인 대상이 되는 블록 내 어휘들을 명사(名詞), 복합명사(複合名詞)만으로 구성한다. 품사를 정확히 결정하는 일은 최종적인 결과까지 영향을 주게 되므로 가장 주의를 요하는 과정이다. 국내에는 다양한 한글 형태소 분석기가 있기 때문에 본 연구에서는 Korean Morphological Analyzer라는 RHINO 한글 형태소 분석기를 사용한다. RHINO는 입력된 문장을 어절(띄어쓰기 단위) 별로 끊어서 각 어절의 형태소와 품사를 분석한다. 분석에 사용된 사전은 국립국어원에서 공개한 1200만 어절 규모의 한국어 현대 문어형태 분석 말뭉치를 기초로 하였으며, 말뭉치는 각 어절을 품사 분석 해 놓았기 때문에 이 자료들을 추출하여 비교적 큰 규모의 사전을 빠르게 만들 수 있다. 또한 이 프로그램은 동적사전(Dynamic Dictionary)을 사용하며, 분석 대상이 들어오면 주위의 문맥을 판단하여 가장 최적의 분석 결과를 제시한다. 단점으로는 아직 실생활에서 혹은 문서에서 나올 수 있는 모든 경우에 대해 단어를 완벽하게 대응하지 못한다는 것이다.

키워드 가중치를 이용한 블록 추출단계에서 주의사항으로 단어 간 띄어쓰기와 영어는 하나의 명사로 봐야한다. 같은 단어라도 띄어쓰기가 되어있는 것과 그렇지 않은 단어에 대해서 다른 결과 값이 나오기 때문에 본 연구에서는 임의적으로 연구자가 정리를 하였지만 이 부분의 경우 추후에 보완을 해 나가야 할 것 같다. 둘째, 키워드 스코어(빈도수)를 계산하여 결과를 도출한다. 문서에서 빈번하게 발생되는 키워드 일수록 주제와 관련성이 높은 단어일 수도 있고, 일반적으로 문서 내에서 자주 쓰이는 공통 단어일 경우도 있기 때문에, 키워드 간 변별력(辨別力)을 주기 위한 기초 작업 이라 할 수 있다. 표 1은 키워드 스코어에 대한 결과 값을 나타낸 표로서 가장 빈번하게 발생되는 키워드들로 재 정렬 한다.

<표 1> 각 키워드 별 본문여부 및 빈도수 결과

ID	Keyword	Text	Frequency
1	sewolho	O	9
2	disastrous accident	O	7
3	support	O	6

<Table 1> The result of the frequency of each keyword-specific content, and the text

결과 리스트 중에서 셋째, 키워드의 빈도수를 기준으로 최소 지지도(n=1인 항목에 대해서)를 만족하는 키워드에 대해 간소화 작업을 진행한다. 지지도가 낮은 항목은 우연히 발생할 수 있는 또는 흥미가 없는 경우일 가능성이 있기 때문에, 웹 문서 내에서 한번 씩 나온 키워드들을 제거한다. 마지막 단계로 키워드 스코어를 기준으로 각 블록에 대해 가중치를 부여한다. 이 때 가중치 값이 높으면 높을수록 웹 문서 내에서 빈번하게 발생되는 키워드가 많다는 것을 나타낸다. 하지만 주의할 점은 빈번하게 발생되는 키워드가 주제를 나타내는 중요 키워드라는 의미가 되는 것은 아니다. 가중치 부여 과정은 표 2를 통해 구할 수 있다.

<표 2> 키워드 별 가중치 부여 계산 과정

ID	Keyword	Noun
10	Special laws are expected to be processed compensation, support of victims and suffering district support projects such memorial consists of three parts, comes 12 in plenary.	Special law, boat ,compensation, victim, suffering district, support, memorial consists, work, plenary, process

<Table 2> Calculation of the weight by keywords

먼저, 키워드는 '특별법(1), 배(1), 보상(1), 피해자(1), 피해지역(1), 지원(1), 추모사업(1), 일(1), 본회의(1), 처리(1)' 이며 각각의 키워드에 대한 스코어 (빈도수)는 '4,4,4,5,2,6,2,5,2,2' 이다. 위 두 개의 값을 곱하여 각각 더하며 최종 키워드 별 가중치 값은 (1*4)+(1*4)+(1*4)+(1*5)+(1*2)+(1*6)+(1*2)+(1*5)+(1*2)+(1*2)=36이 된다. 즉 임의의 블록ID '10'의 가중치는 '36'이 되는 것이다. 이러한 방식으로 나머지 블록들에 대한 가중치 값을 구할 수 있으며, 표3은 Data Type의 길이정보와 키워드(색인어) 가중치를 만족하는 값들을 표로 나타낸 것이다.

<표 3> 길이정보와 가중치를 동시에 만족하는 리스트

ID	Data Type Length Information	ID	Weight by kewords
6	193	6	94
8	113	8	77
13	112	15	53

<Table 3> List that satisfies the length and the weight information at the same time

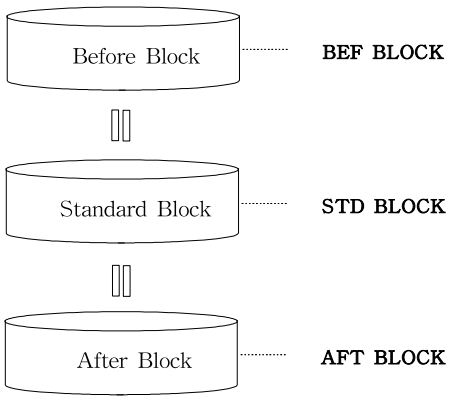
구해진 결과 값 표 3을 토대로 상위 Level 10을 범위로 지정한 후 범위 내 중복으로 만족하는 임의의 ID 값에 대해 구조 분석 단계로 넘어갈 후보 본문 블록을 추출한다. 본 논문에서 상위 범위를 10으로 지정한 것에 대한 근거는 실험 전 각각의 개별적인 특징을 가진 블로그, 카페, 뉴스에서 임의의 문서를 3개씩 추출하여 분석한 결과 Level 10을 넘어 갈수록 비 본문 영역의 블록들이 본문 영역의 블록보다 더 많이 추출됨을 발견 할 수 있었다.

3.2 구조분석단계

영역구분단계에서 추출 된 후보 본문 블록들을 토대로, 구조 분석 단계에서 각 후보 블록들의 전후(前後) 블록들에 대해 군집화(Clustering)하여 하나의 집단으로 본문요소를 확장한다.

본문은 구성상 하나의 영역으로 표현되며, 본문 전체가 연속된 문장으로 이루어져 있다고 가정한다. 이런 가정은 텍스트의 구조적 응집성(Text Coherence)으로 설명할 수 있으며, 본문 내 위치한 텍스트들은 거리상 서로 가깝게 밀집하여 위치한다.

(그림 4) 구조분석단계 군집화 과정을 도식화로 표현



(Figure 4) Represented by the structure analysis step clustering process schematized

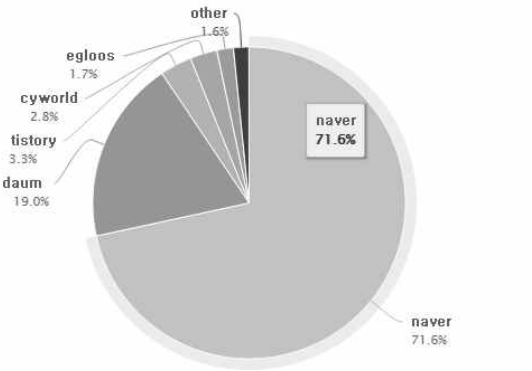
즉 특정텍스트 블록이 본문이라면 그 이웃 텍스트 블록 또한 본문이며, 이웃에 위치한 텍스트 블록과의 거리, 이웃 텍스트 블록의 특징은 특정 텍스트 블록의 본문 여부를 결정하는 중요한 역할을 한다. 군집화를 하는 첫 번째 과정은 영역구분 단계에서 추출한 후보 본문 블록을 기준으로 해당요소의 위치를 검색한다. HTML 원본 문서 내 전체 블록 요소로부터 영역구분단계에서 추출한 후보 본문 블록에 대한 문장을 연구자가 직접 검색 하면 동일한 텍스트를 갖는 요소를 찾고, 해당 요소(자식요소)에 대한 부모노드를 2차적으로 검색한 다음 추출한 부모요소가 포함하는 모든 자식요소를 출력하는 것이다. 이러한 방법은 앞서 설명한 구조적 응집성에 근거하여 영역구분단계에서 추출한 후보 본문 블록에 대한 전/후 블록과 함께 군집화를 하고, 추출된 블록들에 대한 결과 값 중 공통 된 영역이 있을 경우 해당 영역을 본문으로 확정한다.

4. 실험 및 평가

4.1 실험 데이터

분석 실험의 정확성을 위해 데이터 선별에 있어 연구자의 주관적인 의견이 개입되지 않아야 하고, 특정기간이나 사건에 영향을 받지 않아야 한다. 또한 범주간의 실험 문서의 수 즉, 표본의 개수도 동일하도록 문서를 수집해야 한다. 이를 위해 전체 문서 집단에서 무작위로 일정개수의 문서를 선별하되 추출된 문서 간 중복되는 문서가 없도록 실험 문서를 추출해야 한다. 본 연구에서는 블로그, 카페, 뉴스 레이아웃 구조의 사이트에서 직접 수집한 30개의 웹 문서 데이터 세트를 이용하여 제안한 방법의 타당성을 검증하고자 한다. 다만, 아래사항에 대해서는 예외처리(Exception) 으로 간주한다. 첫째, 블로그, 카페는 싸이월드, 다음, 이글루스와 같은 대부분의 페이지에서 연구자가 소스코드에 대한 자료 수집이 자유롭지 못하기 때문에 그림 5와 같이 상위 랭크를 점유하고 있는 블로그 중 네이버(점유율 70.1%)와 티 스토리(점유율 3.4%)로 모집단을 한정하였다.

(그림 5) 블로그 점유율



(Figure 5) Portion by the Blog http://www.blogchart.co.kr/

또한 한국어에 대한 형태소 분석이 이루어졌기 때문에 영문 사이트는 제외하고 한국어로 이루어진 사이트를 중심으로 웹 문서를 수집하였고 셋째, 웹 사이트 원본에는 기존 <script> 요소 때문에 jQuery로 구현된 클러터 제거 코드가 충돌하여 error가 발생한다. 그렇기 때문에 전처리 단계 전에 기존 코드에 대한 <script>를 사전에 제거함으로써 이와 같은 error가 발생하지 않게 미리 차단한다.

4.2 평가방법

일반적으로 정보검색 시스템의 성능을 평가하는데 쓰이는 척도인 F-Measure 값으로 평가를 진행하였다. F-Measure은 분류 문제의 평가에서 가장 광범위하게 사용되며 재현율(Recall)과 정확률(Precision), 그리고 F1의 조화 평균으로 계산된다. 이러한 방법은 선행 연구에서도 사용되었던 방법으로[15][16][17] 재현율, 정확률, F1은 식(3)과 같이 정의된다.

$$\text{재현율}(R) = \frac{|C \cap D|}{|C|} \qquad (3)$$
$$\text{정확률}(P) = \frac{|C \cap D|}{|D|}$$
$$F1 = \frac{2PR}{R+P}$$

위 식에서 C는 정답 블록들의 집합이고, D는 시스템으로부터 추출된 블록들의 집합이다. $C \cap D$는 정답 블록들 중에 시스템이 정답이라고 올바르게 예측한 블록들의 집합을 의미한다. 즉 재현율은 정답 블록들 집합 중 추출된 블록들의 비율을 말하며, 정확률은 추출된 블록들 중 정답인 블록들의 비율을 나타낸다.

일반적으로 n개의 클래스를 가진 다중 클래스 분류 평가 문제에서의 최종 F-Measure 값은 각 클래스별로 F-Measure 값을 측정한 후, 데이터 개수나 중요도에 따라 클래스 값의 가중치 합을 구하는 방식으로 계산된다. 본 논문에서의 분류 영역인 텍스트 블록의 본문 여부 결정 문제 또한 본문/비본문의 2-클래스를 가진 다중 클래스 분류 문제이므로 본문/비본문의 각 F-Measure 값을 계산한 후, 클래스 별 데이터의 개수에 따라 가중치 합을 계산할 수 있다. 그러나 다중 클래스 분류 문제이더라도 데이터의 분포가 한쪽으로 치우친 경우에는 전체 F-Measure 값이 데이터가 많이 포함된 쪽의 F-Measure 값에 의해 좌우 될 수 있기 때문에 더 중요한 본문을 분류하지 못하였음에도 불구하고 전체적으로는 비교적 높은 정확도를 가진다고 판단될 수 있다. 이에 따라 본문 영역에 대한 정확한 분류 여부를 평가하기 위해 비본문의 분류 F-Measure는 무시하고, 본문에 대한 F-Measure 만을 측정하여 2.2절에 설명한 기존 기법과 비교, 분석하고 타당성을 검증한다.

4.3 성능평가

본 논문에서는 2.2절에서 설명한 텍스트 블록의 단어/링크 밀도에 의한 본문 분류 방법에 대해 4.2절 평가방법에 설명한 것과 같이 재현율(R), 정확률(P), F1 값으로 비교하였다. 먼저, 오류 유형별로 분류해 본 결과 제목을 본문과 함께 추출하는 경우가 가장 많았고, 본문 사이에 끼어있는 광고가 함께 추출 되는 경우가 두 번째로 발생하였다. 본문 내 광고 추출 오류는 광고가 본문과 별도의 위치에 있는 것이 아니라 본문 중간에 나타나는 경우에 발생하기 때문이므로 블록 영역의 텍스트 길이가 1음절 단어일 때 사전에 제거해야 한다. 그 다음으로는 본문 전체가 추출되지 않거나, 본문 일부가 추출되지 않는 경우인데, 이럴 경우에는 본문의 텍스트가 비 본문영역 보다 적을 때 많이 발생된다. 그리고 댓글이 추출되는 경우가 그 뒤를 이었다. 기타 오류 유형을 살펴보면 저작권 표시와 같은 전혀 상관없는 콘텐츠가 추출된 경우로 나타났다. 이처럼 같은 Template의 웹 페이지라 할지라도 포함하는 본문의 텍스트 정도에 따라 정확도에 차이를 보이기 때문에 높은 결과 값을 얻기 위해서는 다양한 웹 페이지의 데이터 분석이 필요한 것이다.

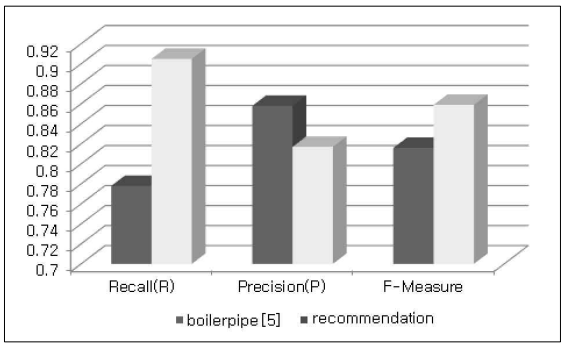
표 4와 그림 6은 연구자가 임의로 추출한 데이터 30세트에 대해 2.2절 즉, 웹 문서의 본문 추출을 위한 공개 시스템으로 현재까지 가장 성능이 좋다고 알려진 방법과 제안한 시스템의 본문 분류 결과를 나타낸다.

<표 4> 기존시스템과 제안한 시스템과의 성능비교

		boilerpipe[5]	Proposed Method
Data Set	Recall(R)	0.778	0.905
	Precision(P)	0.858	0.817
	F-Measure	0.816	0.859

<Table 4> List that satisfies the length and the weight information at the same time

(그림 6) 데이터 세트에 대한 성능 비교



(Figure 6) Comparing the performance of the data set

각 표와 그래프로 나타난 실험 결과에서 볼 수 있듯이 제안한 시스템이 boilerpipe에 비하여 F-Measure가 약 4.3%의 성능 향상 효과를 보였다. 또한 두 개의 데이터 세트에서 정확도의 큰 차이를 보이지 않음으로써 제안한 방법을 이용하여 데이터에 민감하지 않은 본문 추출이 가능함을 알 수 있게 되었다.

5. 결론

최근 들어 웹을 통하여 새롭게 생성되는 정보의 양이 급속도록 증가하면서 웹으로부터 유용한 정보를 추출하는데 관심이 모아 지고 있다. 그로인해 웹 문서가 대량으로 생산되고 정보추출의 대상도 자연스럽게 웹 문서로 이동하게 되었다. 웹 문서의 경우 일반문서와는 달리 그 내용이나 형식이 매우 자유로울 뿐만 아니라 빠르게 변화하고 있기 때문에 개발자, 또는 저작자는 이에 능동적으로 대처해야한다.

이런 환경의 변화를 반영하여 최근 정보추출에 관한 연구는 기계학습, 지식관리, 에이전트, 자료융합, 정보수집 등의 기술과 결합하여 적용가능한 혹은 학습 가능한 정보추출 분야로 발전해가고 있다. 그래서 웹 문서에서 정보를 표현하는 가장 중요한 영역인 본문영역을 자동 추출하는 것은 특정 형식에 얽매이지 않는 다양하고 자유로운 형식으로 작성된 웹 문서들을 효과적으로 빠른 시간 안에 재가공할 수 있는 대표적인 방법으로 폭넓게 사용된다. 즉, 방대한 웹 문서들로부터 문서의 핵심 내용이라고 할 수 있는 본문 내용을 추출하는 것은, 웹 문서를 이해하고 분석하여 서비스하기 위한 가장 기초적이면서도 중요한 단계이므로 이를 위하여 본 논문에서는 수집한 웹 문서에 대해 영역구분단계와 구조분석단계를 걸쳐 본문영역을 추출하는 방법을 제안함으로서 통계적인 방법과 지식 기반 분류 방법의 장점을 모아 대량의 문서를 신속하고 정확하게 렌더링 할 수 있게 된다. 향후에는 다양한 데이터 세트를 이용한 검증과 함께 최근에 제안된 다른 방법들과의 성능을 비교하여 타당성을 검증해야 할 것이고, 특히 본 논문에서 제안한 방법이 다양한 언어에서도 제한되지 않은 웹 문서 환경에 적합함을 보다 명확히 검증하기 위해 한국어뿐만 아니라 다른 언어의 문서 집합을 수집하고 이를 이용한 실험과 검증이 필요할 것이다.

References

[1] J.M. Lim, S.J. Jang, M.Y. Kim, J. H. Lee, “2014 Status of Utilization of Internet,” Korea Internet Agency, 2014

[2] Deng C., Shipeng Y., Ji-Rong W., Wei-Ying M., “VIPS: a Vision-based Page Segmentation Algorithm,” Microsoft Technical Report(MSR-TR-2003-79), 2003.

[3] Suhit G., Gail E. K., Peter G., Michael F. C., Justin S., “Automating Content Extraction of HTML Documents,” World Wide Web, vol.8, Issue2, pp.179-224, 2005.

[4] Jeff P., Dan R., “Extracting Article Text from the Web with Maximum Subsequence Segmentation,” The 18th international conference on World wide web, pp.971-980, 2009.

[5] Stefan E., “A lightweight and efficient tool for clcaning Web pages”, The 6th International Conference on Language Resources and Evaluation, 2008.

[6] Christian K., Peter F., Wolfgang N., “Boilerplate Detection using Shallow Text Features,” The third ACM

international conference on Web search and data mining, pp.441-450, 2010.

[7] Jian F., Ping L., Suk Hwan L., Sam L., Parag J., Jerry L., "Article Clipper-A System for Web Article Extraction," 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.743-746, 2011.

[8] Tim W., William H. H., Jiawei H., "CETR-Content Extraction via Tag Ratios," 19th international conference on World wide web, pp.971-980, 2010.

[9] Jung-chan Yun, Sung-dae Yun, "Design of personalized Web mining using association rules ", Journal of Korea multimedia society, Vol. 11-11, pp.1566~1574, 2008.

[10] Hyung-woo Lee, Tae-su Kim, "Research of knowledge inference algorithm with associated mining method based on Ontology", Journal of Korea multimedia society, Vol. 11-11, pp.1601~1614, 2008.

[11] Tomaz K., Evaluating Text Extraction Algorithms. [Online]. Available: http://tomazkovacic.com/blog/ (downloaded 2012, Jul.)

[12] W3C Recommendation. (1999, Dec. 24). HTML 4.01 Specification [Online]. Available: http://www.w3.org/TR/html401/ (downloaded 2012, Jul.)

[13] Ju-gil Hong, Eun-young Shin, Jue-il Lee, Won-Seok Lee, "Automatic Hierarchical Classification of news articles using association rules", Journal of Korea multimedia society, Vol. 14-6, pp.730~741, 2011.

[14] Won-moon Song, Woo-seung Kim, Mung-won Kim, "HTML document, extraction using the context of the surrounding text blocks", Journal of Korean Institute of Information Scientists and Engineers : Software and Applications, Vol. 40-3, pp.155~163, 2013.

[15] S.-H. Lin, J.-M. Ho, Discobering Informative Content Blocks from Web Documents. Proc. of 8th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, 2002.

[16] Young-gu Lee, "Study on the article text extraction from news web page", Journal of Korea Society for Information Management, Vol. 26, pp.305-320, 2009.

[17] L. Bing, Y. Wang, Y. Zhang, Primary Content Extraction with Mountain Model. Proc. 8th IEEE CIT, 2008.

전 현 지

2012년 : 충주대학교 컴퓨터과학과 졸업(공학사)
2015년 : 서울과학기술대학교 산업대학원 컴퓨터공학 전공 (공학석사-재학중)

2013년~현 재: 서울과학기술대학교 산업대학원 컴퓨터 공학과 재학중
관심분야 : HTML, Data Mining, 정보보호(Personal Information) 등

고 찬

1974년2월: 경희대학교 기계공학과 (공학사)
1991년2월: 경희대학교 대학원 전자공학과 (공학박사)
2008년 2월 : 서울대학교 대학원 기술정책 (경제학박사)
1974년 10월~1978년 2월 : 해군장교 복무 (해군대학 및 초계함 승조)
1987년 12월 : 정보처리기술사 (한국산업인력공단)
2005년 12월~현 : 핀란드, 헬싱키메트로폴리아대학교 강의겸임교수
2013년 12월~현 : UAE, 아부다비대학교 강의겸임교수
1983년 10월~현 : 서울과학기술대학교 교수
관심분야: ICT 경제분석 및 정책, 그래픽스/게임제작, 문화산업 정책, 경영정보시스템