뉴스 웹 페이지에서 기사 본문 추출에 관한 연구*

A Study on Extracting News Contents from News Web Pages

이용구(Yong-Gu Lee)**

초 록

웹을 통해 제공되는 뉴스 페이지의 경우 필요한 정보 뿐 아니라 많은 불필요한 정보를 담고 있다. 이러한 불필요한 정보는 뉴스를 처리하는 시스템의 성능 저하와 비효율성을 가져온다. 이 연구에서는 웹 페이지로부터 뉴스 콘텐츠를 추출하기 위해 문장과 블록에 기반한 뉴스 기사 추출 방법을 제시하였다. 또한 이들을 결합하여 최적의 성능을 가져올 수 있는 방안을 모색하였다. 실험 결과, 웹 페이지에 대해 하이퍼링크 텍스트를 제거한 후 문장을 이용한 추출 방법을 적용하였을 때 효과적이었으며, 여기에 블록을 이용한 추출 방법과 결합하였을 때 더 좋은 결과를 가져왔다. 문장을 이용한 추출 방법은 추출 재현율을 높여주는 효과가 있는 것으로 나타났다.

ABSTRACT

The news pages provided through the web contain unnecessary information. This causes low performance and inefficiency of the news processing system. In this study, news content extraction methods, which are based on sentence identification and block-level tags news web pages, was suggested. To obtain optimal performance, combinations of these methods were applied. The results showed good performance when using an extraction method which applied the sentence identification and eliminated hyperlink text from web pages. Moreover, this method showed better results when combined with the extraction method which used block-level. Extraction methods, which used sentence identification, were effective for raising the extraction recall ratio.

키워드: 웹 뉴스 기사 추출, 문장 기반 추출, 블록 기반 추출, 웹 마이닝
web news content extraction, sentence based extraction, block based extraction, web mining

---

* 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 〔인문사회 분야: KRF-2008-356-H00001〕.

** Visiting Scholar, School of Information Sciences, University of Pittsburgh (yglee@mail.sis.pitt.edu)

■ 논문접수일자 : 2009년 2월 17일 ■ 최초심사일자 : 2009년 2월 20일 ■ 게재확정일자 : 2009년 3월 2일
■ 정보관리학회지, 26(1): 305-320, 2009. 〔DOI:10.3743/KOSIM.2009.26.1.305〕

1. 서 론

현재 뉴스를 제공하는 미디어로는 전통적인 전달매체인 인쇄물과 방송 미디어가 있고, 최근에는 웹도 이에 포함된다. 웹을 통해 제공되는 뉴스는 기존 매체에서 제공되는 내용을 그대로 제공하거나 독자적인 콘텐츠를 발굴하여 제공하기도 한다. 더 나아가 기존 매체를 이용하지 않고 오직 웹을 이용하여 뉴스를 제공하는 인터넷 신문 웹 사이트도 등장하였다.

이들 매체들이 웹을 통해 뉴스를 제공할 때, 뉴스 본문과 함께 다양한 부가 정보를 제공한다. 이러한 부가 정보는 주로 웹을 이용하기 위한 메뉴, 관련기사 링크, 다양한 광고와 이미지 등을 포함한다. 대부분 이들은 뉴스 기사와 관련이 없어 웹 뉴스를 가공하거나 응용할 때 방해(잡음)가 된다. 예를 들어 뉴스 콘텐츠를 색인할 때, 뉴스 본문과 함께 이러한 잡음도 같이 색인되어 시스템의 성능 뿐 아니라 효율성도 저하시킨다.

웹 페이지를 대상으로 잡음을 제거하기 위해 많은 연구들이 있어 왔다. 웹 페이지를 검색하거나 분류하기 위해 콘텐츠를 추출하는 웹 마이닝(web mining)에서는 잡음 제거가 최종 시스템의 성능향상을 가져오는 것으로 알려졌다(Yu et al. 2003; Yi, Liu, and Li 2003).

이러한 마이닝 기법을 이용하여 일반적인 웹 페이지로부터 콘텐츠를 추출하는 연구는 많이 수행되었으나 뉴스 웹 페이지를 대상으로 기사 본문을 추출하는 연구는 드물다. 또한 전자의 연구 초점은 시스템의 최종 성능 향상에 맞추어져 있어 뉴스 추출 단계의 정확한 성능을 파악하기 힘들다. 뿐만 아니라, 뉴스 페이지의 특성을 제대로 반영하지 않고 일반적인 웹 마이닝 기법을 그대로 적용한다면 최적의 성능을 보장하기 어렵다.

웹 페이지로부터 뉴스 본문을 추출할 때 다양한 방법을 이용할 수 있다. 즉 HTML 태그를 제외한 페이지 내에 모든 텍스트를 뽑아내는 기초적인 방법에서 뉴스 기사의 본문을 정확히 뽑아내는 고급 방법까지 생각할 수 있다. 이 연구에서는 다양한 기사 추출 방법들을 제시하고 그에 따른 성능을 평가하였다. 특히, 뉴스의 특징인 텍스트(문장)와 웹 페이지의 특징인 태그의 깊이(동일 깊이에 따른 블록)를 결합하여, 뉴스 웹 페이지로부터 문장과 블록에 기반한 뉴스 기사 추출 방법을 제시하고 최적의 성능을 가져올 수 있는 방안을 제시하고자 하였다.

2. 웹 마이닝

웹 마이닝은 웹 문서나 서비스에서 필요한 정보를 자동적으로 발견하거나 추출하기 위하여 데이터 마이닝 기법을 이용한다(Etzioni 1996). 웹을 대상으로 한 어플리케이션들은 웹 마이닝 기법을 많이 이용하기에 이들 기법의 성능이 시스템 또는 어플리케이션의 최종 결과에 많은 영향을 미칠 수 있다.

일반적으로 웹 페이지 구조를 다섯 가지 주요 영역으로 구분한다. 이들 주요 영역은 헤더(header), 풋터(footer), 좌측 영역, 우측 영역, 중앙 영역이다(Cadenhead, Chen, and Cook 2008). 이 중에서 중앙 영역이 주요 콘텐츠가 위치한 영역으로, 이 영역을 구분하고 콘텐츠를 추출하기 위해 웹 페이지 분할(web page segmentation: WPS) 알고리즘을 사용한다.

WPS 알고리즘은 페이지를 정해진 길이로 나누어 분할하는 고정길이 페이지 분할 기법, 규칙 기반 구조 분석을 통한 페이지 분할 기법, HTML의 DOM 트리 기반 페이지 분할 기법, 주요 블록 레벨에 의한 분할 기법, VIPS(vision-based page segmentation) 기법 등으로 나누어 볼 수 있다.

규칙 기반 구조 분석 기법은 웹 페이지에서 XML을 이용하여 분할 규칙을 만들고 이를 통해 페이지의 주요 영역을 식별한다(Vitali, Iorio, and Campori 2004).

DOM을 이용한 페이지 분할은 HTML을 DOM 트리로 바꾼 후에, 광고제거나 하이퍼링크 제거 등과 같은 여러 필터링 모듈을 통해 트리에서 각각의 노드를 제거하거나 수정하여 콘텐츠를 추출한다. 이 기법은 단지 콘텐츠를 찾아내기 보다는 비 콘텐츠 요소를 제거하는데 초점을 둔다(Gupta et al. 2003).

주요 블록 레벨에 의한 분할 기법은 웹 페이지의 핵심 콘텐츠가 위치한 중앙 영역을 식별하고, 그곳에서 주요 블록을 찾아 필요한 데이터를 추출한다. 이 방법은 페이지를 분할하는 HTML 태그나 문서구조를 만들어주는 태그를 찾아내어 블록을 식별한다(Cadenhead, Chen, and Cook 2008).

특히 VIPS 기법의 경우 HTML의 트리 구조와 웹 페이지를 시각적으로 나누는 태그를 이용하여 블록이나 페이지를 분할한다(Yu et al. 2003). 또한 각각의 블록에 대해 중요도를 부여하거나 중앙영역을 찾아내어 콘텐츠를 추출한다(Song et al. 2004).

웹 페이지에서 뉴스 본문을 추출하는 연구로 Reis 외(2003)를 들 수 있다. 이 연구는 4,088개 페이지를 대상으로, 웹 페이지 간의 구조적 유사성을 측정하는 개념인 TED(tree edit distance)를 이용하여 뉴스 본문을 추출하여 87.71%의 성능을 보였다. 다만 추출 결과를 평가하기 위해 정확도와 같은 단순한 성능 평가 척도를 이용하여 이 연구와는 직접적인 비교가 불가능하다. 한국어를 대상으로 한 연구(한광록 외 2007)에서는 먼저 HTML 트리에서 하이퍼링크를 갖는 태그를 제거하고, 간단한 경험규칙을 이용해 나머지 불필요한 정보를 제거하여 기사 본문을 추출하였다. 이 연구 역시 최종 요약 성능을 보여주었지만, 구체적인 기사 본문 추출 성능은 제시하지 않았다.

3. 실험설계

3.1 실험문헌집단 구축

이 연구의 실험 대상은 뉴스 웹 페이지이므로 이를 인터넷을 통해 수집하였다. 실험 결과의 보편성을 획득하기 위해 실험문헌집단을 구축할 때 다음과 같은 기준을 적용하였다. 우선 다양한 형태의 뉴스 정보원(news source)을 통해 뉴스를 추출하였으며 주제도 다양하게 수집하였다. 또한 웹 페이지를 선정할 때 랜덤하게 수집하여 다양한 페이지가 실험대상으로 되도록 하였다. 이를 위해 많은 뉴스를 제공하는 웹 포털[1]을 이용하였으며, 그 결과 〈표 1〉을 얻었다.

1) 이 페이지(http://news.naver.com/main/presscenter/category.nhn)에 제공되는 뉴스는 각 언론사가 직접 편집한다.

〈표 1〉 실험집단 추출을 위한 정보원 및 기사 수

유 형	사이트 및 URL	기사 수
일간지	경향신문	13
	국민일보	13
	동아일보	13
	문화일보	10
	서울신문	13
	세계일보	13
	조선일보	12
	중앙일보	13
	한겨레	12
	한국일보	13
	소 계	125
방 송	KBS TV	11
	MBC TV	11
	mbn TV	13
	SBS TV	12
	WOW환경 TV	12
	YTN TV	13
	소 계	72
경제/IT	디지털타임스	13
	매일경제	12
	머니투데이	12
	서울경제	13
	아시아경제	13
	아이뉴스24	13
	이데일리	13
	전자신문	13
	지디넷코리아	13
	파이낸셜뉴스	11
	한국경제	13
	헤럴드경제	13
	소 계	153
인터넷신문	노컷뉴스	13
	미디어오늘	12
	오마이뉴스	11
	프레시안	11
	소 계	47
스포츠	스포츠서울	13
	스포츠조선	11
	일간스포츠	12
	소 계	36
매거진/ 지역	매일신문	9
	부산일보	11
	씨네21	6
	주간한국	12
	한경비즈니스	8
	소 계	46
전문지/ 영자지	조세일보	7
	코리아타임스	7
	소 계	14
블로그	tistory.com 외 5개 사이트	6
	소 계	7
합 계		498

포털의 뉴스홈 사이트로부터 506개의 뉴스 URL을 추출하였으나 만화, 동영상, 디렉터리 페이지 등 뉴스 본문을 추출할 수 없는 8개 페이지를 제외한 498개를 실험대상으로 선정하였다. 선정된 리스트의 주제 분야는 각 신문사의 헤드라인 뉴스에 해당하므로 다양하다.

선정 리스트의 하이퍼링크로부터 URL과 추후 실험에서 사용하기 위해 포털에 제시된 기사 제목을 같이 저장하였고, 해당 URL의 웹 페이지는 3일 후에 수집하였다. 그 이유는 기사 게시 후 교정과 시간에 따른 페이지의 변화를 반영하기 위함이었다. 즉, 뉴스 웹 페이지는 기사 본문이 생성된 후, 관련 기사에 대한 링크, 이용자의 댓글(comment) 등과 같은 본문 이외의 정보가 추가되는데 이러한 변화를 반영하기 위함이었다.

수집된 웹 페이지의 HTML를 파싱하기 위해 명백한 태그 편집 오류를 수정하였다. 대표적인 예로 잘못된 각주 문법을 사용하거나 한 페이지에서 〈body〉 및 〈html〉 태그를 두 번 이상 중복 하여 사용한 경우 이를 정상적인 형태의 HTML 태그로 수정하였다. 또한, 뉴스 편집을 위한 특수기호나 문자 중 정확한 한글 인코딩을 방해하는 부분을 수정하였다. 이러한 수정은 뉴스 콘텐츠에 영향을 주지 않는 범위 내에서 수행하였다.

이 연구에 적용된 기사 추출 방법의 성능을 평가하기 위해, 실험 대상 웹 페이지로부터 수

작업으로 여러 가지 정보를 추출하였다. 구체적인 추출 내용은 뉴스 기사의 제목, 부제, 기사 본문, 저자 등이다. 몇몇 기사에서는 제목과 부제를 식별하기 어려워 다음과 같은 규칙을 적용하였다. 기사 제목의 경우 해당 HTML의 〈title〉 태그에 사용된 문장, 포털에 제시된 제목과 유사한 정도, 기사 본문의 첫 문장 순으로 채택하였다. 부제의 경우 제목 다음 문장으로, 글자체나 크기 등이 기사 본문과 다른 문장들을 채택하였다.

3.2 기사 본문 추출 모형

3.2.1 기사 추출 모형의 배경

선행연구를 살펴보면, HTML 문서를 분석하거나 파싱할 때 가장 기본적으로 쓰이는 개념이 바로 문서 객체 모형(Document Object Model: DOM)이다. 이 모형은 플랫폼 또는 언어 중립적인 인터페이스로 프로그램이나 스크립트가 문서의 내용, 구조, 그리고 스타일을 동적으로 접근하거나 업데이트 하는 것을 가능하게 한다.

내용을 추출하기 위해서는 HTML로 된 웹 페이지에 대해 일련의 분석 또는 파싱 처리를 필요로 한다. 즉, HTML 문서를 파서를 이용하여 DOM 트리를 생성해야 한 후, 이 트리의 논리적인 단위로부터 필요한 정보를 추출한다(Gupta et al. 2003). 논리적인 단위는 하이퍼링크가 될 수도 있고 특정 태그 내의 텍스트가 될 수도 있다. 그 단위가 텍스트라면 속성상 문장용 태그인 '〈P〉', 테이블 셀 태그인 '〈TD〉', 헤더 태그인 '〈H〉' 등과 함께 많이 사용한다.

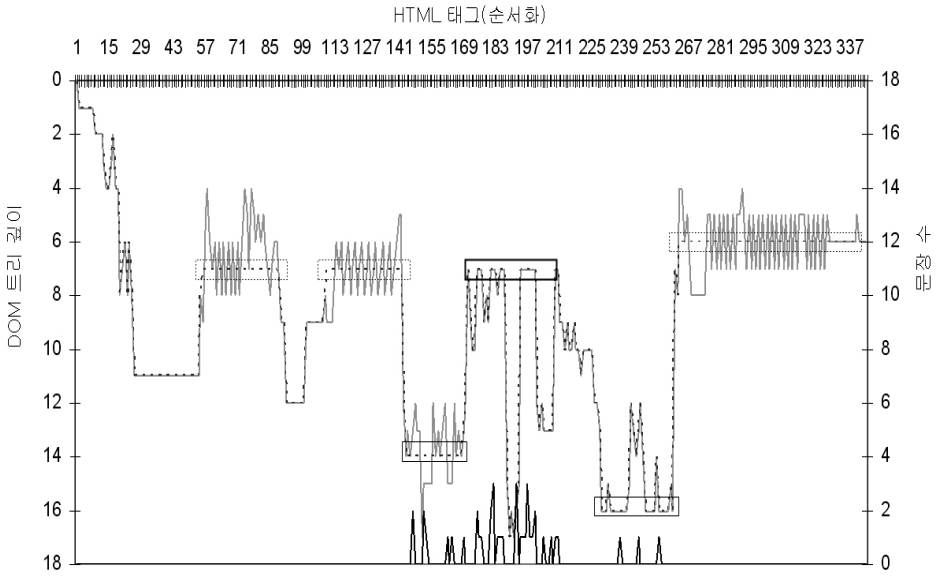
트리 구조는 루트(root) 노드로부터 자식 노드, 참고로 잎 노드로 계속해서 깊이(depth) 또는 레벨(level)을 가지면서 탐색 또는 이동할 수 있다. 마찬가지로 HTML을 파싱해 구축된 DOM 트리도 태그에 따라 깊이를 가지며, 동일한 깊이를 갖는 이웃노드들을 묶어 하나의 블록으로 만들 수 있다.

한편, 뉴스 기사의 경우 대개 제목, 본문, 저자 등의 하위 구조를 가진다. 본문의 경우 여러 문단으로 구성되어 있으며, 문단은 하나 또는 여러 문장으로 구성된다. 즉 본문 전체는 각각의 문장이 연속적으로 또는 수평적으로 나열된 형태를 띤다고 볼 수 있다. 이러한 특징은 텍스트의 구조적 응집성(text coherence)으로 설명이 가능하다.

뉴스 웹 페이지의 경우 뉴스 기사를 HTML 문서 형식으로 표현한 것이다. 이를 공간적으로 보면, 뉴스 기사의 특징으로 수평적인 형태의 문장과 HTML의 특징인 수직적인 형태의 깊이(DOM 트리)의 모습으로 표현할 수 있다. 따라서 DOM 트리에서 특정 태그의 깊이를 찾아내고, 그 태그와 동일한 깊이에 의해 묶인 블록을 생성하고 한 블록 내의 문장(수)을 계산할 수 있다. 더 나아가, 두 변수(깊이와 문장수)의 관계를 HTML의 태그라는 한 요인에 의해 나타낼 수 있다. 이러한 구조를 실례를 통해 표현해 보면 〈그림 1〉과 같다. 이 그림을 만든 HTML 문서는 실제 뉴스 웹 사이트[2]에서 추출되었다.

〈그림 1〉을 보면, x축은 HTML 문서 내의

2) http://www.msnbc.msn.com/id/28155450/



〈그림 1〉 DOM 트리의 깊이와 문장 수

태그를 순서(번호) 대로 나열하였으며, 그 태그들이 DOM 트리 내에서 어느 정도의 깊이(그림에서 위의 그래프)를 갖는지를 왼쪽의 y축을 통해 볼 수 있다. 또한 태그가 갖고 있는 문장의 수(아래의 꺽은 선)는 오른쪽 y축을 통해 볼 수 있다. 그림에서 태그들은 동일하거나 유사한 깊이를 갖는데, 이러한 블록(사각형) 중에 문장을 포함하는 것(실선)과 그렇지 않은 것(점선)이 있으며, 동일한 깊이를 갖는 것(첫 번째, 두 번째, 네 번째)도 있다. 실제 뉴스 기사는 문장들이 많이 분포하는 실선 사각형 중에 중간에 위치한 굵은 실선 사각형 안에 존재한다.

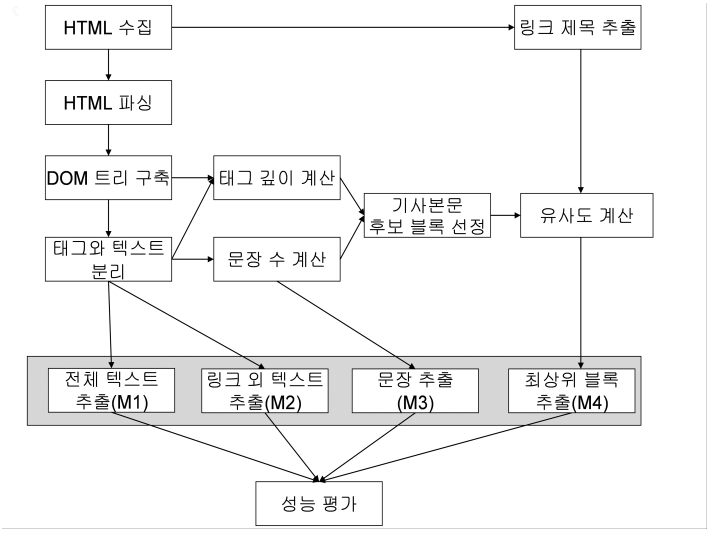
뉴스 기사 추출에서 〈그림 1〉이 갖는 의미는 다음과 같다. 깊이에 따라 묶인 블록(사각형)을 만들고 이들이 얼마나 많은 문장을 포함하는지를 계산하면, 가사의 본문이 들어 있는 블록을 발견할 수 있다. 그 이유는 웹 페이지를 디자인할 때 동일한 의미를 갖는 콘텐츠는 한 곳(블록화)에 모아놓기 때문이다.

실선 사각형에 대해 실제 웹 페이지의 내용을 통해 설명하면, 첫 번째 실선 사각형은 관련 뉴스 기사에 대한 내용이며, 두 번째는 기사 본문, 세 번째는 스폰서 링크인 광고에 해당한다. 특히 기사 본문에 해당하는 굵은 실선 사각형 내에서 두 개의 큰 계곡을 볼 수 있는 있는데, 이는 기사 본문 내에 존재하는 광고와 기사와 관련된 동영상에 해당한다.

이 연구에서는 뉴스 웹 페이지의 이러한 두 가지 특징을 이용하여 기사 본문을 추출하고자 하였다.

3.2.2 기사 추출 방법

이 연구에서는 기본적인 4 가지 기사 추출 방법을 이용하였다. 웹 페이지로부터 뉴스 본문을 추출하기 위해 〈그림 2〉와 같은 실험 과정을 거쳤다.



〈그림 2〉 전체 시스템 과정 및 방법

먼저 실험문헌집단을 구축하기 위해 포털로부터 뉴스 URL과 링크 텍스트를 추출하였다. 이렇게 추출된 리스트로부터 HTML 페이지를 수집하였다.

수집된 HTML에 대해 파싱을 거쳐 DOM 트리를 구축하였다. 이때 파싱이 불가능한 HTML에 대해서는 실험 대상에서 제외하였다.

DOM 트리의 모든 노드를 순회하면서 해당 노드가 태그에 해당하는지, 아니면 텍스트에 해당하는지를 체크하고 이를 이용하여 페이지 내에 쓰인 태그와 텍스트를 분리하였다. 또한 각각의 태그가 DOM 트리에서 얼마나 깊이를 갖는지 계산하였다. 더 나아가 각각의 텍스트에서 어느 정도 문장을 포함하는지도 계산하였다.

다음 단계로 전 단계에서 계산된 정보(깊이와 문장 수)를 이용하여 기사가 출현 가능한 후보 블록을 선정하였다. 즉 DOM 트리의 동일한 깊이에 출현한 문장 수의 합계를 산출하고, 그 값이 1 이상인 깊이를 후보로 등록하였다.

앞서 HTML를 수집할 때 두 가지 종류의 뉴스 제목을 수집하였다. 하나는 포털에서 해당 뉴스 웹 페이지로 연결할 수 있게 제공된 하이퍼링크의 텍스트이고, 다른 하나는 해당 페이지에서 추출한 원제목이다. 전자의 경우 대부분 포털 화면의 공간적인 제약으로 인해 원제목의 앞부분을 중심으로 압축하여 제공된다.

구체적인 기사 추출 방법은 다음과 같다. 먼저 페이지 내에 모든 텍스트를 뽑아내는 기초적인 방법(M1)으로, 이 방법에서는 HTML 태그와 스크립트를 제외한 모든 텍스트를 추출하였다. 따라서 이 방법에서는 하이퍼링크 텍스트를 포함하였다. 태그와 텍스트를 분리하는 단계에서 텍스트만 대상으로 작업을 수행하였다.

일반적인 뉴스 웹 페이지는 많은 하이퍼링크 태그를 가지고 있다. 이들은 대부분 관련 뉴스 기사로 연결이나 광고, 메뉴 등에 해당한다. 간혹 뉴스 기사 본문에 중요 단어에 대해 하이퍼링크가 부여하는 경우 있긴 하나 이들의 수는

전체 하이퍼링크나 기사 본문의 양에 비하면 매우 적다.

두 번째 기사 추출 방법(M2)에서는 이러한 속성을 사용하였다. 즉 웹 페이지에서 하이퍼링크 태그 전체를 제거하고 나머지 텍스트만 뽑아내는 방법을 사용하였다(한광록 외 2007). 만약 뉴스 웹 페이지가 이용자의 댓글과 같은 일반 텍스트가 존재하지 않고 기사 본문과 관련 기사로의 링크, 광고, 메뉴 등만 존재하는 단순한 형태의 페이지라면 하이퍼링크 관련 정보만 제거하더라도 뉴스 기사 본문과 유사한 텍스트를 얻을 수 있을 것이다.

세 번째 방법은 문장에 기반한 기사 추출 방법(M3)이다. 즉 웹 페이지의 텍스트에 대해 문장 여부를 식별한 후, 문장으로 이루어진 텍스트만 추출하였다. 일반적으로 문장은 그 자체로 충분한 의미를 구성하는 단어들의 집합으로 정의할 수 있지만, 이 연구에서는 마침표(.), 물음표(?), 느낌표(!) 등의 문장 부호를 찍는 단위를 문장으로 처리하였다. 즉, 이들 부호가 뒤에 공란(‘ ’), 줄 바꿈 문자 또는 새로운 태그를 만나는 경우, 이를 문장으로 식별하였다. 태그와 텍스트를 분리한 후, 텍스트에 대해 이 방법을 적용하여 문장을 식별하였다.

네 번째 방법으로 기사 본문 후보로 선정된 블록들과 포털의 하이퍼링크에서 추출한 제목 간의 유사도를 측정하여 가장 높은 유사도 값을 가지는 블록을 기사 본문으로 선정하는 방법(M4)이다. 기사에 댓글이 달리면 많은 텍스트를 포함할 수 있으므로 이들과 기사 본문을 구별하기 위한 특별한 방법이 필요하다. 이 연구에서는 이러한 문제를 해결하기 위해 본문을 요약적으로 표현하는 제목을 이용하였다. 즉, 웹 페이지가 다양한 후보 블록을 가질 때 하나의 블록을 기사 본문으로 선택하기 위해 제목과의 유사도를 이용하였다.

3.3 평가 방법

제안한 여러 방법의 뉴스 본문 추출 성능을 파악하기 위해, 각각의 웹 페이지에서 기사 본문을 수작업으로 추출하였다.

시스템이 추출한 결과와 수작업 결과를 비교 평가하기 위해 일반 정보검색 시스템의 성능 평가에 쓰이는 두 척도인 정확률과 재현율을 응용하여 적용하였다. 이러한 방법은 기존의 선행연구에서도 적용하였는데, 이들 연구에서는 단순척도만 이용하였다(Lin and Ho 2002; Debnath, Mitra, and Giles 2005).

구체적으로 이 실험에서 사용한 성능 평가 척도로 추출 재현율, 추출 정확률, F1 등이고 이들의 공식을 살펴보면 다음과 같다.

$$\text{추출재현율}(R) = \frac{\text{추출된 적합 단어 수}}{\text{적합 단어 총수}}$$

$$\text{추출정확률}(P) = \frac{\text{추출된 적합 단어 수}}{\text{추출된 단어 총수}}$$

$$F = \frac{2PR}{P+R}$$

평가의 기본이 되는 단위는 단어로 하였다. 즉 수작업 평가 집단에서 출현한 단어와 실험 결과에서 출현한 단어의 일치 여부로 평가하였다. 그 이유는 기사 본문은 단어로 이루어져 있으며, 이들 단어가 가공처리 되어 다른 어플리케이션에서 색인어 또는 자질로써 의미를 갖기

때문이다. 따라서 공란이나 편집용 특수 문자의 경우 무의미하므로 평가에서 제외하였다.

또한 각각의 웹 페이지에 대해서는 위의 단순 척도를 적용하였지만, 전체 실험 문헌집단(498건)에 대해서는 각 방법별로 복합척도인 마이크로평균 F1(microaverage F1: MicroAvg_F1)과 매크로평균F1(macroaverage F1: MacroAvg_F1)를 사용하였다. 또한 마이크로평균 F1을 구하기 위해 필요한 마이크로평균 정확률(microaverage precision: MicroAvg_Pr)과 마이크로 평균 재현율(microaverage recall: MicroAvg_Re)을 제시하였다(Sebastiani 2002). 다만 제안한 방법의 결과를 보다 쉽게 이해하기 위해 마이크로 평균 F1을 중심으로 평가하였다.

4. 실험 결과 분석

4.1 기사 추출 방법의 성능

먼저 실험문헌집단의 특징을 살펴보기 위해 이들이 얼마나 많은 단어를 포함하고 있는지 텍스트 관련 통계를 조사하였다. 그 결과 〈표 2〉를 얻었다.

〈표 2〉를 살펴보면, 태그 및 관련 정보를 제외한 HTML 페이지의 전체 텍스트의 단어는 348,704개이며, 이중 기사 본문은 151,369개로 전체 단어의 43.41%를 차지하였다. 또한 링크가 걸려 있는 단어와 나머지 단어는 128,596개와 68,738개로 36.88%와 19.71%를 차지하였다. 흥미로운 점은 기사 본문에 출현한 단어의 수와 하이퍼링크에 출현한 단어의 수가 큰 차이(6.53%)를 보이지 않는다는 것이다. 색인의 관점에서 보면, 이것은 기사 본문과 그 외의 텍스트를 같이 색인할 경우 많은 양의 불필요한 정보를 색인하게 되어 시스템의 효율성을 저하시킬 수 있음을 의미한다. 또한 뉴스 웹 페이지로부터 기사 본문을 추출할 때에는 하이퍼링크 내의 텍스트만 제거하여도 불필요한 정보를 많이 줄일 수 있음을 의미한다.

이 연구에서는 앞서 설명한 기본적인 기사 추출 방법을 적용하였다. 각각의 방법 내용을 요약하면, 뉴스 웹 페이지로부터 전체 텍스트를 추출하는 방법, 하이퍼링크 텍스트 이외의 텍스트를 추출하는 방법, 문장으로 식별된 텍스트를 추출하는 방법 그리고 DOM 트리의 깊이로 블록화한 후 이 블록들과 기사 제목의 유사도를 이용한 방법이다.

이 연구에서는 제목과 블록간의 유사도를 비교하기 위하여 N-gram 색인과 벡터공간 검색 모형을 결합하였다(Cavnar 1995). 이 모형에서 제목은 질의로, 블록은 문헌으로 표현하였다. 구체적인 시스템의 처리 과정은 다음과 같다.

① 실험과정에서 추출한 전체 텍스트를 읽어

〈표 2〉 실험문헌집단의 단어 통계

	전체 텍스트	기사 본문	링크 텍스트	기타 텍스트
합계	348,704	151,369	128,596	68,739
평균	700.21	303.95	258.22	138.03
비율(%)	100.00	43.41	36.88	19.71

3-gram 사전을 구축하였다. 이때 각각의 3-gram에 대해 IDF를 계산하였다.

② DOM 트리 깊이에 따른 텍스트를 추출하고 이들 블록에 대해 3-gram을 생성하였다. 또한 이들을 코사인 정규화된 로그 TF · IDF를 가중치로 갖는 3-gram 벡터를 표현하였다.

③ 제목에 대해 블록과 마찬가지로 처리하여 코사인 정규화된 로그 TF · IDF를 가중치로 갖는 3-gram 벡터를 표현하였다.

④ 코사인 유사계수 공식에 의해 두 벡터 간의 유사도를 산출하고 각각의 블록을 순위화하여 그 결과를 제공하였다.

이 실험에서 문헌 표현으로 N-gram 방법을 적용한 이유는 형태소 분석이나 불용어 제거와 같은 언어학적 처리를 필요로 하지 않고 특정 언어나 주제 분야에 의존적이 아니라는 특성을 갖기 때문이다(정영미 2005). 즉 웹 페이지는 여러 언어가 함께 사용될 확률이 높아 이들 다중 언어를 N-gram으로 처리하는 것이 용이하기 때문이다.

기본적인 기사 추출 방법에 따른 성능평가 결과는 〈표 3〉에 제시하였다.

기사 추출 방법의 결과를 마이크로평균 F1 값 중심으로 살펴보면, 가장 좋은 성능을 보인 방법은 문장 식별을 통해 추출한 방법(M3)이며 다음으로 페이지 내에서 하이퍼링크 텍스트를 제외한 방법(M2) 순으로 나타났다.

실험 결과에 기사 본문이 얼마나 출현하는지를 나타내는 마이크로평균 재현율의 경우, 전체 텍스트를 추출한 방법(M1)이 가장 좋은 성능을 보였고, 방법(M2)이 그 다음으로 좋은 성능을 보였다. M1의 경우는 전체 텍스트를 추출하기 때문에 모든 기사 본문이 추출되어 이 척도 값이 100%에 달하였다. M2의 경우, 기사 본문 내에 포함된 하이퍼링크 텍스트가 제거되어 척도 값이 100%에 못 미쳤다.

여기서 주목할 점은 M3의 마이크로평균 재현율이 98.06%로 같은 척도에서 더 좋은 성능을 가져온 M1과 M2에 비해 크게 낮지 않다는 것이다. 즉 뉴스 본문이 문장으로 구성된 텍스트이므로 웹 페이지에서도 문장을 식별하고 이를 추출하여도 대부분의 본문이 추출되는 것을 알 수 있다. 다만 간단한 문장 식별 규칙을 적용하여 그 성능이 완벽하지 못했다. 대표적인 예로 HTML 태그가 문장의 중간에 들어가 앞뒤로 문장이 나뉘거나 기사 본문 사이에 나타나는 중간제목의 누락을 들 수 있다.

마이크로평균 정확률은 마이크로평균 재현율과 정반대의 결과를 보였는데, 이는 M3이나 M4가 기사 본문 이외의 불필요한 정보를 제거하는 기능을 하고 있기 때문인 것으로 보인다.

〈표 3〉 기본적인 기사 추출 방법의 성능

	MicroAvg_Pr	MicroAvg_Re	MicroAvg_F1	MacroAvg_F1
M1	43.41	100.00	60.54	57.20
M2	68.72	99.93	81.44	79.01
M3	73.67	98.06	84.13	81.66
M4	89.67	55.64	68.67	63.02

실험 결과에서 M4가 가장 낮은 성능을 보인 원인을 찾기 위해 문헌별 마이크로평균 F1의 값을 〈그림 3〉에 제시하였다.

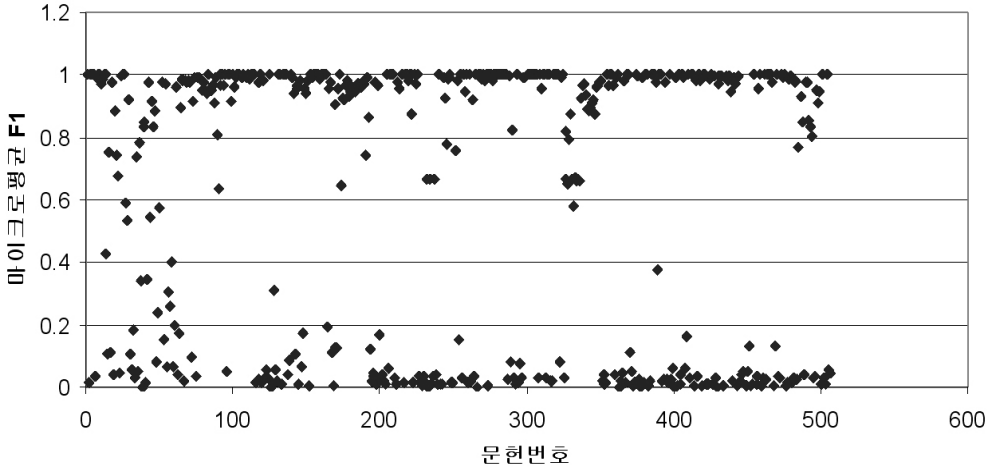
〈그림 3〉에서 보는 바와 같이, 문헌에 따라 마이크로평균 F1값이 심하게 양극화된 현상을 보였다. 이는 하나의 조건(여기서는 기사 본문과 제목의 유사도)에 만족하면 매우 좋은 결과를 가져왔으나 그렇지 않은 경우 매우 나쁜 결과를 가져오는 것으로 보인다. 조건을 만족하지 못하는 주된 이유는 두세 단어로 된 짧은 제목에서 기인하는데, 제목의 핵심 단어가 본문보다는 관련 기사 하이퍼링크 텍스트나 관련 댓글에 반복적으로 출현하여 이들 텍스트와 높은 유사도 값을 가지기 때문이다.

4.2 기사 추출 방법의 결합

이 연구의 주된 초점은 웹 페이지에서 문장 식별과 태그의 깊이에 따른 블록화를 통해 뉴스 본문을 추출하는 것이다. 따라서 두 요인을 결합하여 어떠한 성능을 가져오는지를 알아보기 위해 기본적인 방법들을 결합하였다. 먼저 M3과 M4에서 쓰인 두 요인(문장 수와 깊이에 의한 블록화)을 결합하였으며, 이들 기본 방법들에 대해 하이퍼링크 텍스트를 제거하는 M2를 결합하였다. 마지막으로는 기본 방법 3가지 모두를 결합하였다.

두 요인을 결합하기 위해 M3에 M4를 결합하였다. M4의 경우, 두 단계로 이루어져 있는데, 하나는 동일한 깊이에 따른 블록화이고 다른 하나는 블록과 제목의 유사도 계산이다. 따라서 M4를 두 단계를 나누어 결합하였다. 먼저 MM는 단지 M3에 블록화만 적용하였다. 즉 전체 문장 중에 블록 별로 문장 수를 계산하여 최대 문장 수를 가지는 블록의 텍스트를 기사 본문으로 추출하였다. 다음 M34는 MM에 더해 최대 문장을 가지는 블록을 제목과 비교하여 유사도를 산출하고 유사도 값이 0.0 이상이면 해당 블록을 기사 본문으로 추출하였다.

M4에 M3을 결합하였는데, 이 방법(M43)은



〈그림 3〉 M4에 의한 문헌별 마이크로 F1 값

먼저 블록별로 텍스트를 추출하고 제목과의 유사도를 산출(M4)한 후 문장 수를 계산하였다. 즉 유사도 값이 최대인 블록에 대해서 문장 수를 계산하여 한 개 이상의 문장이 출현하면 그 블록을 기사 본문으로 추출하였다.

이와 같이 두 가지 요인을 결합한 방법들에 의해 생성된 결과의 성능은 〈표 4〉와 같다.

결과를 보면, M34가 마이크로평균 F1이 83.26%로 가장 좋은 성능을 보였으며, 다음으로 최대 문장 수를 가지는 블록을 기사 본문을 추출한 MM이 좋았다. M43은 가장 낮은 성능을 보였다. M3에 M4를 적용하는 것이 반대로 적용한 방법보다 더 좋은 성능을 가져왔다. 그 이유는 〈표 3〉에서 보듯이 M3의 성능, 특히 마이크로평균 재현율이 매우 우수하기 때문인 것으로 보인다. 이는 추출 성능을 높이고자 여러 방법을 결합할 때 더 좋은 성능을 보이는 방법을 먼저 적용해야 한다는 것을 보여준다.

전체적인 평가척도인 마이크로평균 F1에서 보면 여전히 두 요인을 결합한 방법 모두 M3보다 낮은 성능을 보인다.

다음으로 M3과 M4에 M2를 결합하여 실험하였다. 먼저, M3에 M2를 결합한 M32는 페이지에 나타난 전체 문장 중에 하이퍼링크 텍스트에서 추출한 문장을 제거한 나머지를 기사 본문으로 추출하였다. 또한 M4에 M2를 결합하였는데, M42는 하이퍼링크 텍스트를 제거하고 블록별로 제목과 유사도를 산출하여 최상위 블록을 기사 본문으로 추출하였다. 이들 방법의 결과를 보면 〈표 5〉와 같다.

M2를 적용한 M3과 M4 모두 상당히 높은 향상율(각각 6.99%, 28.37%)을 보였다. 특히 M42의 경우 마이크로평균 재현율에서 매우 큰 향상이 이루어졌다. 이렇게 M2를 적용하였을 때 성능이 향상되는 주된 이유는 하이퍼링크 텍스트 중에 문장으로 식별되기 쉬운 형태(예로 '?'와 '!'로 끝맺음)로 된 텍스트가 제외되면서 상대적으로 기사 본문이 많이 추출되었기 때문이다.

M32의 경우 M3보다 4.95% 향상된 것으로 나타났다. 이는 하이퍼링크 텍스트를 제거하고 문장을 식별하여 기사 본문을 추출하는 M32가 우수한 성능을 보여 비교적 간단하면서 효율적인 방법임을 입증하였다.

마지막으로 M2, M3, M4를 모두 결합하였다. 결합 순서로는 각각의 M3과 M4에 M2를 먼저 수행하고 나머지 방법을 결합하였다. 그 결과

〈표 4〉 두 요인에 의한 결합 방법의 성능

	MicroAvg_Pr	MicroAvg_Re	MicroAvg_F1	MacroAvg_F1
MM	91.37	71.90	80.47	64.62
M34	79.42	87.48	83.26	79.40
M43	78.52	56.29	65.57	58.89

〈표 5〉 하이퍼링크 텍스트 제외 기법을 결합한 방법의 성능

	MicroPr	MicroRe	MicroAvg_F1	MacroAvg_F1
M32	86.37	91.96	89.08	88.20
M42	80.52	88.18	84.18	80.97

〈표 6〉을 얻었다.

〈표 6〉을 보면, M3에 다른 방법을 결합한 방법이 더 좋은 성능을 보였다. 이는 〈표 4〉의 결과에서 기술한 설명과 동일한 이유이다. 특히 M324의 경우 마이크로평균 F1이 89.35%로 전체 실험을 통해 가장 좋은 성능을 보였다. 이는 M32의 마이크로평균 F1보다 약간 높다. 다만 이 정도의 성능 차이라면 이 방법에 포함된 제목과 블록간의 유사도 계산 과정이 시스템 자원을 많이 요구하므로 효율성이 떨어진다고 볼 수 있다.

4.3 매체별 비교 및 종합

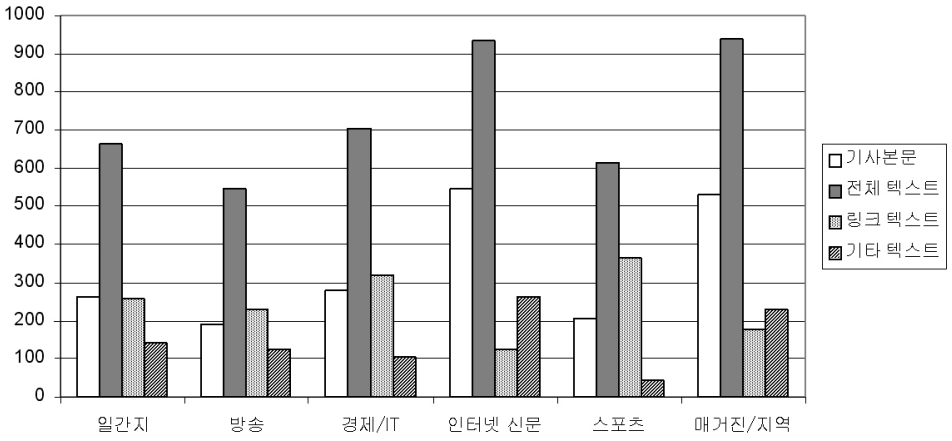
매체별 성능을 비교하기 전에 〈표 1〉에서 제시된 매체별 텍스트 크기를 살펴보면, 〈그림 4〉와 같았다. 다만 전문지/영자지와 블로그의 경우, 수집된 HTML의 수가 적어 표본으로써 대표성을 갖지 못하므로 분석에서 제외하였다.

〈그림 4〉에서 기사 본문을 비교해 보면, 인터넷 신문과 매거진/지역 매체가 비교적 긴 기사 본문을 가졌으며 이들은 기타 텍스트도 비교적 길었다. 아마도 이들 매체들이 이용자의 댓글과 같은 추가 정보를 뉴스 웹 페이지에 많이 담고 있는 것으로 보인다. 반대로 방송과 스포츠 신문 매체의 경우 상대적으로 짧은 기사 본문을 가졌으며 특히 스포츠 신문의 기타 텍스트는 가장 짧았다.

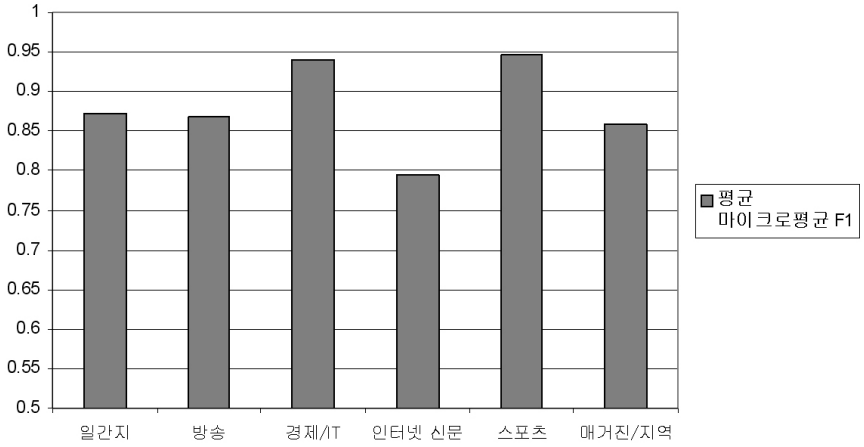
가장 좋은 성능을 보인 기사 추출 방법(M324)의 결과를 매체별로 비교하여 〈그림 5〉를 얻었다. 가장 좋은 성능을 보인 매체는 스포츠와 경제/IT 분야 매체이었다. M324가 문장에 기반

〈표 6〉 세 가지 결합 방법의 성능

	MicroPr	MicroRe	MicroAvg__F1	MacroAvg__F1
M324	86.88	91.96	89.35	89.37
M423	85.85	59.70	70.43	64.66



〈그림 4〉 매체별 텍스트 통계



〈그림 5〉 매체별 기사 추출 성능

하여 기사를 추출하므로 적합 텍스트에 속하는 기사 본문이 상대적으로 많고, 부적합 텍스트에 해당하는 기타 텍스트가 적은 매체에서 좋은 성능을 보였다. 반대로 인터넷 신문과 매거진/지역 매체의 경우 저하된 성능을 보였다. 이들 매체에서는 부적합 텍스트인 기타 텍스트가 가장 길었다. 따라서 앞으로 기타 텍스트에 대한 보다 효율적인 처리가 필요한 것으로 보인다.

지금까지의 실험을 종합적으로 살펴 보면 다음과 같다.

먼저, 뉴스 웹 페이지에서 문장을 식별하고 이를 이용하여 기사 본문을 추출하는 것은 성능 평가에서 추출 재현율의 향상을 가져오는 것으로 드러났다. 이는 뉴스가 문장으로 이루어져 있고 대부분의 웹 페이지에서 뉴스 본문이 차지하는 비율이 높기 때문이다. 앞의 실험에서는 문장 식별 기능을 먼저 이용한 방법들의 마이크로평균 재현율이 평균 88.27%로 블록을 이용한 방법(64.95%) 보다 23.32%가 높았다. 이러한 모습은 하이퍼링크 텍스트 제거도 같은 역할을 하는 것으로 나타났다.

둘째, 블록화와 더 나아가 제목과의 유사도를 이용한 방법들은 추출 정확률을 높이는데 도움을 주는 것으로 나타났다. 전체적으로 이 방법들은 성능 향상에 도움이 되지 않으나 다른 방법과 결합하였을 때 성능향상에 도움이 되는 것으로 나타났다.

셋째, 뉴스 본문 추출 과정에서는 하이퍼링크 텍스트 제거는 간단하면서 매우 중요한 요소로 보인다. 이는 실험문헌집단 전체를 놓고 보았을 때 이들이 차지하는 비율이 매우 높음을 통해 간접적으로 알 수 있다. 단어를 기준으로 전체 텍스트의 36.88%에 해당하는 하이퍼링크 텍스트의 제거는 재현율과 정확률을 모두 향상 시키는 것으로 나타났다. 정말 간단한 경험규칙으로 상당히 높은 성능을 얻을 수 있음을 보여 주었다.

넷째, M32와 M324가 가장 좋은 마이크로평균 재현율 91.96%를 보였는데 이는 전체 기사 본문에서 8.04%만을 누락하였다는 뜻이며, 마이크로평균 정확률은 평균 86.63%로 이 방법들이 추출한 기사 본문에서 13.37%만이 오류

에 해당한다는 것을 의미한다.

마지막으로, 매체별 기사 추출 성능 비교 분석을 통해 기타 텍스트에 대한 보다 효율적인 처리가 성능 향상을 가져올 것으로 보인다. 앞서 설명한 것처럼 문장이나 제목과의 유사도를 통한 추출 방법으로는 기타 텍스트를 따로 분리하기엔 어려움이 있다. 따라서 일반적으로 뉴스 웹 페이지를 디자인할 때 기사 본문을 앞에 위치시키고 기타 텍스트를 뒤에 위치시키는 경험규칙을 적용하여 기타 텍스트를 분리해냄으로써 충분히 성능향상이 가능할 것으로 보인다.

기타 실험 과정에 나타난 고려사항은 다음과 같다. 실험 데이터 구축과정에서 수집된 제목 중 수작업으로 수집한 제목을 이용하여 유사도 실험을 하였으나 기존의 제목을 이용한 실험과 그리 큰 차이를 보이지 않았다. 또한 실험문헌집단의 51.5%만이 기사 제목을 〈title〉 태그를 사용하여 입력하여 이 실험에서는 이용하기 어려웠다.

5. 결 론

이 연구에서는 웹 페이지로부터 뉴스 본문을 추출하기 위해 문장과 블록에 기반한 뉴스 기사 추출 방법을 제시하였고, 추출 재현율과 추출 정확률을 이용하여 실험 결과를 평가하였다. 실험문헌집단은 여러 뉴스 웹사이트로부터 다양한 주제와 형태를 갖는 웹 페이지 498개를 수집하였으며, 성능 평가를 위해 각각의 페이지에서 수작업으로 기사 본문을 추출하였다. 실험 결과 다음과 같은 결과를 얻었다.

첫째, 웹 페이지로부터 뉴스 본문을 추출할 때 페이지 내의 문장을 식별하여 추출하는 방법이 재현율을 높여 효과적인 방법임을 보여주었다. 또한 하이퍼링크 텍스트의 제거 방법도 같은 기능을 하는 것으로 나타났다. 따라서 이들의 결합을 통해 효율적이면서 효과적인 시스템을 구축할 수 있을 것으로 보인다.

둘째, DOM 트리의 동일 깊이를 이용한 블록화는 미미한 추출 정확률 성능향상을 가져왔다. 이 방법은 추후 좀 더 심도 있는 연구를 통해 성능을 끌어올릴 수 있도록 최적화 과정이 필요한 것으로 보인다.

셋째, 기본적인 기사 추출 방법보다 이들을 결합한 방법이 더 좋은 성능을 가져오는 것으로 나타났다. 이때 각 방법 간의 결합 순서는 더 좋은 성능을 보인 방법을 먼저 적용하여야 한다.

이 연구에서는 뉴스 웹 페이지에 대해 문장과 블록이라는 두 가지 요소로 접근하였다. 또한 이 요소들을 이용하여 기사 본문을 추출하는 방법과 하이퍼링크 텍스트를 제거하는 간단한 경험규칙을 결합하여 최적의 추출 성능을 가져오도록 하였다. 이 연구의 제한점으로는 한글로 된 문헌만을 대상으로 실험하였으므로 다른 언어로 된 뉴스 사이트에 대해 적용해 볼 가치가 있으며 실험 대상 문헌의 규모도 좀 더 크게 할 필요가 있다. 또한 추후에 기존의 웹 페이지 분할 기법을 결합하여 보다 최적화된 성능을 가져오는지 알아볼 필요가 있다.

참 고 문 헌

정영미. 2005. 『정보검색연구』. 서울: 구미무역 출판부.

한광록, 선복근, 유형선. 2007. 웹 뉴스의 기사 추출과 요약. 『한국 컴퓨터정보학회 논문집』, 12(5): 1-10.

Cadenhead, Tyrone, Jinlin Chen, and Terry Cook. 2008. "Improving web information indexing and retrieval based on center block duplication detection." *International Journal of Innovative Computing and Applications*, 1(3): 194-204.

Debnath, Sandip, Prasenjit Mitra, and C. Lee Giles. 2005. "Automatic extraction of informative blocks from webpages." *Proceedings of the 2005 ACM Symposium on Applied Computing*, 1722-1726.

Etzioni, Oren. 1996. "The world wide web: Quagmire or gold mine." *Communications of the ACM*, 39(11): 65-68.

Gupta, S., K. Kaiser, D. Neistadt, and P. Grimm. 2003. "DOM-based content extraction of HTML documents." *Proceedings of the 12th International Conference on World Wide Web*, 249- 256.

Lin, Shian-Hua and Jan-Ming Ho. 2002. "Discovering informative content blocks from web documents." *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 588-593.

Reis, Davi Castro, Paulo Golgher, Altigran Silva, and Alberto Leaender. 2003. "Automatic web news extraction using tree edit distance." *Proceedings of the 13th International Conference on World Wide Web*, 502-511.

Sebastiani, Fabrizio. 2002. "Machine learning in automated text categorization." *ACM Computing Surveys*, 34(1): 1-47.

Song, Ruihua, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. 2004. "Learning block importance models for web pages." *Proceedings of the 13th International Conference on World Wide Web*, 203-111.

Vitali, Fabio, Angelo Di Iorio, and Elisa Ventura Campori. 2004. "Rule-Based Structural Analysis of Web Pages." *Document Analysis Systems VI*, 425-437.

Yi, Lan, Bing Liu, and Xiaoli Li. 2003. "Eliminating noisy information in Web pages for data mining." *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and data Mining*, 296-305.

Yu, Shipeng, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. "Improving pseudo-relevance feedback in web information retrieval using web page segmentation." *Proceedings of the 12th International Conference on World Wide Web*, 11-18.