韓國컴퓨터情報學會 論文誌
第16卷 第1號, 2011. 1.

2011-16-1-1-3

블로그 검색 성능 향상을 위한 주제-랭크 기법

신 현 일*, 윤 은 일*, 류 근 호*

The Topic-Rank Technique for Enhancing the Performance of Blog Retrieval

Hyeonil Shin*, Unil Yun*, Keun Ho Ryu*

요 약

1인 미디어인 블로그에 대한 관심이 증가함에 따라, 블로그 검색과 관련된 다양한 랭킹 알고리즘들이 제안되었다. 이러한 알고리즘들은 블로그가 웹 페이지와 다르게 갖는 구조적 특징에 맞게 변형되었으며, 각 블로그간의 연결이나, 댓글, 트랙백들을 통해 이루어진 상호소통 속에서 나타난 결과들을 바탕으로 블로그의 평판이나 인기도를 수치화하여 검색 시스템에 반영한다. 하지만 실제 블로그 검색에서는 블로그 자체의 랭크뿐만 아니라 검색어와 블로그 글과의 적합성과 시간 등의 요소를 복합적으로 사용하게 된다. 그런데 기존에 알려진 요소만으로는 검색 결과의 품질이 낮을 수 있다. 본 논문에서는 블로그의 주제와 관련도가 가장 높은 블로그를 찾아 낼 수 있는 주제-랭크 기법을 제안한다. 이 기법은 블로그와 블로그 글의 색인어뿐만 아니라, 블로그 글을 대표하는 주제와의 관계까지 랭킹을 매기는 방법이다. 제안된 기법을 통해 블로그 검색에서 검색어와 블로그의 연관성에 따라 랭킹을 효과적으로 부여할 수 있다. 본 논문 제안하는 주제-랭크 기법을 적용한 블로그 검색 시스템의 정확률과 적용률을 국내의 다른 블로그 검색 시스템들과 비교해 본 결과, 주제-랭크 기법을 사용한 블로그 검색 시스템의 성능이 타 시스템에 비해 더 우수함을 알 수 있었다.

▸ Keyword : 블로그, 랭크 알고리즘, 검색 엔진, 주제

Abstract

As people have heightened attention to blogs that are individual media, a variety rank algorithms was proposed for the blog search. These algorithms was modified for structural features of blogs that differ from typical web sites, and measured blogs′ reputations or popularities based on the interaction results like links, comments or trackbacks and reflected in the search system. But actual blog search systems use not only blog-ranks but also search words, a time factor and so on. Nevertheless, those might not produce desirable results. In this paper, we suggest a topic-rank technique, which can find blogs that have significant degrees of association with topics. This technique is a method which ranks the relations between blogs and indexed words

• 제1저자 : 신현일 • 교신저자 : 윤은일
• 투고일 : 2010. 07. 15, 심사일 : 2010. 09. 13, 게재확정일 : 2010. 09. 28.
* 충북대학교 전자정보대학 컴퓨터전공(Dept. of Computer Science, Chungbuk National University)
※ ″이 논문은 2010년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음 (This work was supported by the research grant of the Chungbuk National University in 2010)″.

of blog posts as well as the topics representing blog posts. The blog rankings of correlations with search words are can be effectively computed in the blog retrieval by the proposed technique. After comparing precisions and coverage ratios of our blog retrieval system which applis our proposed topic-rank technique, we know that the performance of the blog retrieval system using topic-rank technique is more effective than others.

▸ Keyword : Blog, Rank Algorithm, Search Engine, Topic

Ⅰ. 서 론

전 세계적으로 인터넷 보급률이 높아지고 인터넷 속도가 빨라지는 것에 비례하여 웹에서 접근이 가능한 정보들이 기하급수적으로 늘어나고 있다. 이런 정보의 홍수 속에서 원하는 정보를 찾기 위해서는 찾고자 하는 정보의 특성에 맞는 인터넷 정보 검색 시스템이 필요하다. 특히 우리는 웹에 올라와 있는 수많은 정보 중에서도 전 세계적으로 사람들의 최근 관심과 생각들을 보여주면서 가장 활발하게 정보의 바다를 채우고 있는 블로그라는 웹사이트 형식에 관심을 갖게 된다면 웹상에서 사람들이 나누는 정보가 무엇이며, 무슨 생각을 하고 있고, 어떠한 이야기를 나누는지를 알 수 있다[1][2]. 이러한 이야기는 현재 가장 이슈가 되는 것에 대한 주제가 될 수도 있고, 개인적 취향이나 경험에 국한 되는 것일 수도 있다. 그 중에서 다양한 주제를 폭넓게 다루는 인기 있는 블로그가 존재하기도 하고, 그에 반해 몇 가지 주제만 다루고 인기도 상대적으로 적지만 개인의 소중한 의견과 경험이 담긴 블로그도 대다수 존재할 것이다[3]. 실제로 블로그 검색과 관련된 다양한 연구 결과나 랭킹 알고리즘들이 나오게 되었다,[4][5][6][7][8][9-12] 하지만 이러한 기법들은 주로 블로그의 인기도 측정이나, 영향력 높은 블로그의 파급력에 초점이 맞춰져 있다. 또한 현재 서비스 되고 있는 블로그 전문 검색 시스템들도 주로 블로그의 인기도에 많은 가중치를 부여하여 블로그 랭킹을 매기고 있다. 이러한 시스템에서의 문제점은 특정 주제에 적합한 블로그 랭킹의 품질을 장담할 수가 없다는 것이다. 어떠한 블로그가 야구라는 주제에 대해 많은 이야기를 하고 있는지, 어떤 블로그가 영화를 중요한 소재로 삼고 있는 지에 대해 찾을 수 있는 지 등등 주제별로 특화된 블로그 랭킹에 관한 연구가 필요하다.

본 논문에서는 블로그 검색의 성능 향상을 위한 주제-랭크 기법을 제시하고, 이것을 실제 블로그 검색 시스템에 구현해 보고 성능을 평가해 볼 것이다. 2장 관련연구에서는 기존의 블로그 관련 랭킹 알고리즘들과 검색 시스템에 대해 살펴보며, 보완해야 할 점 까지 살펴볼 것이다. 그리고 3장에서는 주제-랭크 기법에 대해 설명할 것이다. 4장에서는 실제로 블로그 검색 시스템을 간단히 구현해 볼 것이며, 여기에 주제별 블로그 랭킹 알고리즘을 적용해 본다. 그리고 이를 바탕으로 4장에서 성능을 평가해 보며, 마지막 5장에서는 주제별 블로그 랭킹에 대한 결론과 향후 더 연구해나가야 할 사항에 대해 알아본다.

Ⅱ. 관련 연구

일반적인 웹 문서를 위한 랭킹 알고리즘으로 유명한 PageRank[5]와 같이 웹 문서간의 링크를 분석하여 랭킹을 매기는 방법은 블로그 검색에 적합하지가 않다. 그 이유는 블로그의 글들은 일반적인 웹 문서에 비해 상대적으로 상호간의 링크 연결이 적기 때문이다.[4][13] 그리하여 이러한 문제를 보완한 블로그 랭킹 알고리즘이 몇 가지 제안되기도 하였다.[4][5][7][8][9-12] 블로그 사이트의 인기도를 측정하기 위해서 블로그의 특성 중, 블로그 글 본문에서의 하이퍼텍스트 링크뿐만 아니라 트랙백(trackback)과 댓글(reply)을 통한 링크[14]를 직접적으로 활용하거나 네티즌의 반응 정도를 계산하여 랭킹을 매긴다. 하지만 이러한 기존의 방법들은 어떤 블로그가 가장 많은 네티즌들에게 인기가 있고, 많은 참조가 되고, 관심을 갖게 되는 지에 대한 랭킹을 매길 수는 있겠지만, 특정 주제에 적합한 블로그를 찾을 때에는 이것만으로는 주제에 적합한 블로그의 순위를 매길 수 없다. 인기 있는 블로그라 해도 해당 블로그가 특정 주제에 대한 비중이나 글의 양이나 질이 떨어질 수도 있기 때문에 이 경우에는 해당 주제와의 적합성은 당연히 낮다고 봐야 할 것이다. 이렇듯 블로그를 검색하게 될 때, 찾고자 하는 주제와 가장 적합한 블로그를 찾기 위해서는 블로그의 인기도뿐만 아니라, 블로그와 주제와의 연관성을 수치화 하여 비교할 수 있는 랭킹 방법이 필요하게 되는데, 본 논문에서는 이것을 “주제-랭크 기법”이라 명명하며. 여기서 말하는 ‘주제(topic)’란 블로그의 각 글 내용을 대표하는 주제, 이야깃거리, 화제가 되는 대상 등을 뜻 한다. 블로그의 구조적 특성을 이용하여 특정 주제에 대한 블로거의 명성을 블로그 랭킹에 이용한 블로그-랭크 알고리즘[15]이 제안이 되었는데, 여기서의 주제는 블로그 태그에서 추출된 색인어라는 제약이 있다. 하지만 블로그의 태그는 블로거가 직접 입력하는 블로그의 주제, 소재 단어를 뜻하는 것으로, 블로거가 태그를 올바

르게 사용하지 않는 경우에는 큰 효과가 없다. 블로그의 주제를 올바르게 추출하기 위해서는 블로그 태그뿐만 아니라 블로그의 다양한 구조적 특성을 이용해야 한다. 때문에 블로그의 주제에 대응되는 색인어인 주제어들에 대하여, 주제어가 추출된 구조에 따라 가중치를 부여한 색인 방법[16]도 제안이 되었다. 본 연구에서는 주제어 가중치 색인뿐만 아니라 이를 토대로 블로그 주제와 블로그 간의 관련성을 중점으로 랭킹 하여, 검색어와 대응되는 주제와 가장 관련이 높은 블로그를 찾을 수 있게 한다.

실제 국내의 블로그를 검색해볼 수 있는 웹 서비스에는 "블로그앙[2])", "올블로그[1])", "나루 검색[2])" 등이 있는데, 이 중에서 블로그앙의 경우에는 블로그의 인기도에 의해 결과 랭킹을 매기고 있으며, 올블로그의 경우에는 블로그 기본 정보와 주제와의 매칭을 통해 검색 결과 랭킹을 매기고 있다. 또한 국외에서 서비스 중인 블로그 전문 검색 시스템으로는 "테크노라티(technorati.com)"와 "아이스로켓(icerocket.com)"이 있는데, 두 검색 시스템의 큰 특징은 두 서비스가 국내 블로그 검색 서비스 보다 블로그의 태그 정보를 블로그의 분류와 검색에 적극 활용하는 점에 있다. 하지만 이러한 기존 블로그 검색 시스템은 특정 주제로 블로그 검색을 했을 때, 해당 주제에 대한 비중이 약함에도 불구하고 블로그 랭크 자체가 높아지는 문제가 있다. 또한 특정한 글 하나를 검색하기 위함이 아니라, 특정 주제에 가장 적합한 블로그 자체를 찾기 위한 방법에는 적합하지 않다는 문제점이 있다. 이러한 기존 검색 시스템에서 서비스하고 있는 주제별 블로그 검색의 실제 성능에 대해서는 4장 성능 평가에서 본 논문에서 구현한 검색 시스템과의 비교를 통해 좀 더 자세히 알아볼 것이다

III. 주제-랭크 기법

기존 블로그 검색 시스템에서의 주제별 블로그 랭킹은 블로그 사이트 자체의 인기도와 블로그 글 각각의 글에 대한 색인어의 빈도수와 글이 올라온 날짜를 계산하여 최종 블로그 검색 결과 랭킹을 매기는 방법을 사용한다. 본 논문은 여기에 추가적으로 블로그 글과 주제와의 관계 그래프도 검색 랭킹 결과에 반영하기 위해 주제-랭크 기법을 제안하며, 특정 주제와 대응되는 색인어인 주제어 t와 관련된 블로그 b의 주제-랭크(TRank(b,t)) 계산 방법은 다음과 같다.

$$TRank(b,t) = BTopic(b,t)*c + k + \frac{\sum_{i=1}^{k} score_i}{k}$$

여기서 BTopic(b,t)은 블로그 b에서의 주제어 t의 빈도수로, 블로그 b에서 수집된 글 각각에서 추출된 주제어 t가 수집된 누적 수치이다. k는 블로그 b에서 주제어 t와 관련된 글의 수이며, $\sum_{i=1}^{k} score_i$/k는 주제어 t와 관련된 블로그 b의 글들의 평균 score 값이며, score는 로그 용어가중치 공식인 "1 + log식인f)"로 산출하여 용어가중치식인f)가 1인 단어의 지나치게 낮은 영향력을 보충하고, 이f가 높은 단어의 지나친 영향력을 낮춘 블로그 글 별 용어가중치이다. BTopic(b,t)은 다음과 같이 블로그 사이트와 블로그 사이트 내에 수집된 글에서 추출된 각 글들의 주제어들 간의 관계를 <그림 1>과 같은 그래프로 나타낼 수 있으며 이러한 관계는 검색 시스템에서 색인 과정에서 데이터로 추출하여 처리 할 수 있다. 마지막으로 상수 c는 BTopic(b,t)이 TRank(b,t)에 반영되는 정도를 조절하기 위해 필요하며, 본 연구에서 구현한 블로그 검색 시스템에서는 c=0.5로 적용하였다.

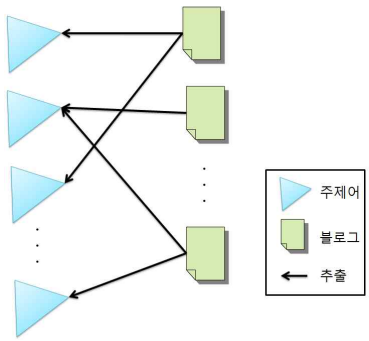

그림 1. 주제-블로그 관계 그래프
Fig. 1. Topic-Blog Relation Graph.

IV. 블로그 검색 시스템의 구현

4.1 구현 환경 및 시스템의 전체 구성

주제별 블로그 랭킹의 검색 결과 성능 평가를 위해서, 간단한 블로그 검색 시스템을 구현해 보았다. 블로그 검색 시스템은 웹크롤러, 데이터베이스 시스템, 사용자 질의 서버 세 가지

1) http://www.blogyam.co.kr
2) http://www.allbog.net
3) http://www.naroo.com

로 나누어 구현을 하였다. 웹크롤러는 블로그 웹페이지를 돌면서, 블로그 글과 블로그 사이트 데이터를 수집하여 데이터베이스 시스템에 색인하게 된다. 시스템 개발 언어는 Python 2.5.2를 사용하였으며, 색인은 관계형 데이터베이스 시스템인 MySQL 4.1을 이용하여 저장하였다. 사용자 질의 서버는 Apache 2를 MS 도우7 RC 환경에서 구동하였다. 웹크롤러가 추출한 블로그 정보(블로그 이름, 블로그 주소, 블로그 RSS주소)는 데이터베이스에 BlogList 릴레이션으로 저장이 된다. 그리고 블로그의 글에서 글의 주제어를 추출하여 주제-블로그 그래프가 Blog-Topic 릴레이션에 저장이 되며, 블로그 글에 관한 정보(주소, 블로그ID, 날짜, 제목)는 PageList에 저장이 되며, 블로그 글과 블로그 글 본문에서 추출된 색인어들과의 관계 그래프는 PageWord에 Sc- ore와 함께 저장이 된다. 색인어 목록은 WordList 릴레이션에 저장이 된다. 각 블로그 글에서 주제어는 글의 제목이나 글 태그에서 1차적으로 추출하였으며, 해당 글을 인용한 타 블로그 글에서 인용 시에 링크에 건 앵커 텍스트(Anchor Text)[17]에서 주제어를 추출하여, 추가적인 외부인용 주제어를 추출하였다. 또한 마지막으로 검색 시스템의 색인 과정이 다 끝난 후, 저장된 전체 글 페이지의 총 개수와 특정 색인어를 포함하고 있는 문헌의 수를 세어서 상수가 17인 피벗 역문헌빈도(pidf)[18]을 계산 하였다.

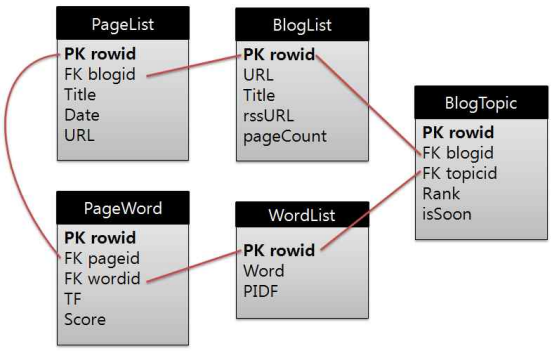

그림 2. 데이터베이스 스키마
Fig. 2. Database Scheme.

이 방법은 기존 역문헌빈도 가중치 기법보다 중간 빈도 용어에 대한 중요도를 더 부여함으로서, 역문헌빈도 계산의 전체적 성능 더 뛰어나다. 피벗 역문헌빈도로 모든 색인어의 용어가중치를 계산 한 것을 기반으로 가장 용어가중치가 높은 상위 2.13%의 색인어를 글의 주제어로 등록하여, 블로그 제목과 태그에서만 주제어가 추출되는 시스템의 단점을 보완하였다. 중심점이 17인 피벗 역문헌빈도 공식은 다음과 같다.

$$pidf(k) = \log_2 \frac{N}{|d_k - 17| + 1}$$

여기에서 N은 색인된 전체 블로그의 수이며, dk는 색인어 k와 관련된 블로그 페이지의 수를 나타낸다. 이 외에도 문맥상 추출 등, 더욱 더 정확하고 신뢰성 있는 주제어 추출 방법이 [19] 존재 하지만, 고수준의 자연어 처리 기술을 요구하기 때문에, 본 연구에서는 글 제목과 태그를 기반으로 주제어를 추출하고, 추출된 주제어를 통해 다시 글 본문에서 재추출하여 주제어를 추가하는 방식을 이용하였다. 이 중에서 태그를 수집하는 방법은 간단하다. 웹로봇은 블로그 페이지의 소스에서 하이퍼텍스트 링크를 거는 'a'태그 중에서 속성이 rel="tag" 인 것을 찾아서 그 부분만 추출하면 된다. rel은 링크를 거는 페이지와의 링크를 당하는 페이지 사이의 관계(relationship)를 표현하는 속성인데, tag로 명명하면 태그를 통해 서로 연결 한다는 뜻이다. 이러한 표기 방법은 마이크로포맷(MicroFormats) 표준에 맞추어 대부분의 블로그 사이트에 공통적으로 적용되어 있다.[20] 또한 블로그 글의 본문은 태그로 감싸는 방식이 많이 쓰이는 데, 이 또한 마이크로포맷의 일종인 hAtom 패턴에 따른 것이다.

4.2 블로그 데이터 추출 및 색인

Python 언어로 구현한 웹크롤러의 핵심 코드는 <표 1>과 같다. 웹로봇인 crawl 모듈은 웹페이지 탐색의 시작 주소 목록과 깊이 우선 탐색의 깊이를 인수로 받는다. 입력 받은 웹페이지들이 저장된 페이지 목록인 pages에서 page의 URL주소를 하나씩 가져와서, urllib2 라이브러리를 이용하여 TCP/IP 인터넷 접속을 하여 페이지 소스를 다운로드 받은 후, 소스를 분석하기 쉽게 하기 위해, Beautif- ulSoup API를 통해 HTML 구조화를 거치게 되면, 실제 색인 작성에 적합한 페이지 인지를 다시 분석하게 되고, 블로그 유형도 분석하게 된다. page-Analysis 모듈은 블로그 유형에 맞게 해당 페이지의 제목, 날짜, 본문의 주제어를 추출해 내어 데이터베이스에 저장을 하게 된다. 이 때 추출된 주제어는 색인에 적합한 색인어로 변환하게 되고, 색인어 별로 랭킹 점수를 매겨서 점수와 함께 저장을 한다.

표 1. 블로그 검색을 위한 웹크롤러 모듈
Table 1. WebCrawler Module for Blog Search.

Python code : crawl Module

```
def crawl(self, pages, depth=3):
for i in range(depth):
newpages=set()
isSeedUrl=1 #히트 페이지 설정
for page in pages:
try: #히트페이지 일 경우, urlFilter모듈도 수행
if isSeedUrl==1:
filteredUrl=self.urlFilter(page)
if filteredUrl!="0":
try:
# 페이지 접속 시도를 10초 이내로 제한
socket.setdefaulttimeout(10)
c=urllib2.urlopen(filteredUrl)
except urllib2.URLError, e: continue
filteredUrl=c.geturl() #실제 주소
blogpage=self.pageFilter(filteredUrl)
if blogpage!="0":
soup=BeautifulSoup(c.read())
offset=self.isBlog(blogpage,soup)
bid=self.addBlog(blogpage,soup,page)
if bid==0: continue
self.pageAnalysis(blogpage,soupbid,,offset)
else: continue
else:
# 히트페이지가 아닐 경우, urlFilter모듈은 생략
try:
socket.setdefaulttimeout(10)
c=urllib2.urlopen(page)
page=c.geturl()
except urllib2.URLError, e: continue
blogpage=self.pageFilter(page)
if blogpage!="0":
soup=BeautifulSoup(c.read())
offset=self.isBlog(blogpage,soup)
bid=self.addBlog(blogpage,soup,page)
if bid==0: continue
self.pageAnalysis(blogpage,soupbid,,offset)
self.dbcommit()
if blogpage!="0":
links=soup('a') # 링크 추출
for link in links:
if ('href' in dict(link.attrs)):
#절대주소로 변환
url=urljoin(page,link['href'])
filteredUrl=self.urlFilter(url)
if filteredUrl=="0": continue
newpages.add(filteredUrl)
except Exception, e: continue
isSeedUrl=0
pages=newpages
```

주제어 및 색인어 추출 시에는 간단하게 구현된 별도의 형태소 분석기 모듈을 사용한다. 그리고 페이지 소스에서 링크 부분만을 추출하여(soup('a')) pages에 추가하고, page들을 다시 순차적으로 깊이 우선 탐색을 반복하면서 다시 데이터와 링크 주소를 저장하는 방식으로 재귀적으로 반복을 하게 된다. pageAnalysis 모듈에서 실제 분석하여 저장하는 데이터들은 블로그 제목(title), 주소(URL)와 날짜(date) 등 페이지 정보를 추출하며, 페이지별로 추출한 주제어들은 블로그 페이지에 나타난 블로그 태그, 블로그 제목, 앵커 텍스트(다른 블로그 글에서 해당 블로그 페이지를 링크했을 때, 링크에 대한 설명인 앵커 텍스트)에서 명사 위주로 단어를 추출하여 블로그의 주제어로 누적 개수와 함께 BlogTopic 릴레이션에 저장하게 되며, 글 본문 내용에서 색인어를 추출하고 빈도수를 세어 Word List 릴레이션에 저장하였다.

4.3 사용자 질의 서버

마지막으로 사용자 질의 서버는 사용자가 자신이 입력한 검색어에 가장 적합한 블로그를 결과로 볼 수 있게 한다. 입력받은 검색어를 서버는 데이터베이스 질의에 치환하여 실제 데이터베이스에서 검색을 하게 되고, 찾아낸 결과는 다시 웹사이트에 출력 형태로 변환되어 웹페이지의 결과로 나타나게 된다. 실제 작동 방식을 한 예로 들자면, 검색어로 '영화'를 입력하고 실행을 하면 질의 서버는 검색어와 매칭이 되는 색인어를 DB의 색인어 릴레이션인 WordList에서 찾아본 후, 해당 색인어가 존재할 경우에만 색인어의 ID를 받아오게 되고, 이 색인어 ID와 관련된 블로그들의 ID를 BlogTopic 릴레이션에서 찾게 된다. BlogTopic에 저장된 블로그와 주제어간의 관계에서는 주제어의 누적수를 저장하게 되는 데, 이 수치에 의해 블로그의 랭킹(rank)이 매겨 지게 된다. 이 순서에 따라 각 블로그의 글들에 대해서 Page Word 릴레이션에서 또 한 번 색인어와 관련된 글들을 랭킹 함수에 의해 산출된 score의 내림차순으로 가져와 검색 결과에서 각 블로그마다 글 목록을 보여주게 된다. 질의 서버에서 검색 과정에서 이루어지는 SQL 문들을 정리해 보면 <표 2>와 같다. 또한 <그림 3>은 '영화'를 검색어로 입력했을 때의 검색 결과 화면이다. 검색어와 가장 적합한 상위 블로그와 해당 블로그에서 '영화'관련 글 목록들이 나오게 된다. 블로그마다 블로그 주소, 블로그 제목이 뜨게 되고, 블로그 글은 각각 글 제목, 글 날짜, 글 미리보기(첫 문장)가 나타나게 되며, 글은 글 URL주소가 링크로 걸리게 된다. 글 목록 순서는 글에 나타난 색인어 별로 매겨진 랭킹 함수에 비례한 빈도수 수치의 내림차순이다. 검색 결과의 정확성에 랭킹을 매기기 위한 방법과 점수를 매기기 위한 일반적인 지표들에는 내용기반 랭킹이나 유입링크를 사용하는 방법이 있다.[6] 이중에서 내용기반 랭킹 방법은 단어빈도, 문서 내 위치, 단어 거리등을 이용하여 랭킹을 매기는 방법이며, 유입링크를 사용하는 방법은 Page- Rank[21] 알고리즘처럼 페이지 간의 참조를 통해 인기페이지에 가중치를 부여하는 방식이나, 링크 자체에 설명이 붙은 텍스트(AnchorTexts)를 검색의 정확도의 지표로 활용하는 방법이다. 단순한 내용 기반 랭킹은 가장 일반적인 랭킹 방법인데, 블로그 태그와 제목과 함께

본문에서의 품사나 격 정보를 통해서 글의 주제를 추출할 수 있다. 그리고 블로그 글 사이의 하이퍼링크 정보는 링크에 붙여진 텍스트(AnchorTexts)에서 추출한 주제어를 해당 블로그의 주제-랭킹에는 부분적인 개념으로 적용이 가능하다.

표 2. 질의 서버에서 요청하는 SQL 쿼리 문의 예
Table 2. Example of SQL Queries requested by Question Server.

1) 검색어 '영화'를 DB 색인어에서 찾기
select rowid from WordList where word='영화';
#결과 : '영화'의 색인어 ID (22795)

2) 색인어 '영화'와 관련된 블로그 찾기
select blogid, srank from BlogTopic
where topicid=22795 order by srank desc;
#결과: srank 내림차순의 블로그 리스트

3) 각 블로그별로 색인어와 관련된 글 찾기
(아래는 블로그 ID가 x인 블로그의 경우임)
select pageid, score from PageWord
where wordid=22795 and blogid=x
order by score desc;
#결과 : score 내림차순의 블로그 글 목록

4) 검색 결과에 보여 줄 블로그 정보를 가져옴
select url,author from BlogList where rowid=x;

5) 검색 결과에 보여 줄 글(ID가 y)의 정보를 가져옴
select title,date,url from PageList where rowid=y;

그림 3. 검색 결과 화면
Fig. 3. Search Results.

V. 성능 평가

검색 엔진의 성능을 평가하기에 앞서, 좋은 검색엔진은 많은 정보를 가지고 있어야 하며, 데이터베이스 내용 갱신 주기가 빨라야 하고, 검색 속도가 빨라야 한다. 또한 사용자가 편리하게 사용할 수 있게 검색 옵션이 쉬우면서도 다양해야 하며, 원하는 정보를 검색 결과 상위 리스트에서 찾을 수 있어야 한다. 본 연구에서 성능 평가를 위해 수집한 블로그 사이트는 총 5,660개 이며, 각 블로그에서 수집된 페이지는 44,106개였다. 색인어는 총 894,649개가 추출 되었다. 블로그 정보를 수집할 때, RSS주소도 수집을 해 놓기 때문에, 초기에 웹로봇이 블로고스피어(Blogosphere)를 항해하면서 각 블로그의 RSS주소를 많이 수집해 두기만 한다면, 차후에는 새로운 글에 대한 데이터 쉽게 축적 할 수 있다. 검색 옵션은 단일 검색어 검색 및 복수 검색어의 AND 검색이 가능하게 구현하였고, 사용자가 원하는 정보를 검색 결과 상위 리스트에서 찾을 수 있게, 검색어와 가장 관련이 높은 순서로 결과를 출력하도록 하였다. 검색 속도와 검색 결과 정확도에 대한 성능 평가결과는 다음과 같다.

5.1 검색 속도 측정

사용자 질의 서버를 로컬에서 접속하여 검색을 수행했을 때, 서버가 반응하여 결과를 출력하는 시간을 측정하였다. 측정 도구로는 Microsoft사의 Web Application Stress Tool 1.0[3]을 사용하였다. 5분동안 5종류의 검색 쿼리를 반복 수행하여 서버가 검색 결과의 첫 Byte를 클라이언트 PC에 까지 전송하는데 걸리는 시간(TTFB)과 검색 결과 화면에 대한 모든 Byte를 전송 완료했을 때의 시간(TTLB)를 측정하여 평균을 구하였다. 반복 수행 횟수는 검색 시스템의 반응에 따라 달랐는데, 본 연구의 경우에는 5분 동안의 총 1278회 수행 결과를 측정한 것이다. 측정수치는 측정을 한 클라이언트 PC네트워크 상태, 측정 시간에 따라 달라질 수가 있지만, 다른 검색 서버의 평균 시간과 비교하여 블로그 검색 시스템이 갖추어야 할 검색 속도로 참고할 수 있다.

표 3. 검색 속도 비교
Table 3. Search Runtimes Comparison.

	본 연구	블로그얌	올 블로그	나루 검색
TTFB	0.157	1.543	0.130	0.333
TTLB	0.234	1.543	2.765	2.403
검색 옵션	블로그 검색	블로그 검색	블로그 검색	기본 설정

단위 : 초

4)http://www.microsoft.com/downloads/details.aspx?FamilyID=e2c0585a-062a-439e-a67d-75a89aa36495

<표 3>을 보면, 비록 본 연구에서 구현한 블로그 검색 시스템은 다른 시스템과 달리 정식 서비스를 하고 있는 것이 아니며, 로컬에서 1대의 PC만 접속이 된 상태에서 측정한 검색 속도이긴 하나, 일반적인 검색 시스템에서 기대되는 검색 속도를 갖추었다는 것은 확인이 가능했다. 즉 본 연구에서 구현한 검색 시스템이 사용자가 검색 서비스에 기대하는 최소한의 검색 속도를 만족시킬 수 있는 성능인지를 알아보기 위해, 기존 시스템들과 비교해 보았는데, <표 3>과 같은 결과라면 충분한 검색 속도 성능을 기대할 수 있다고 여겨진다.

5.2 주제별 정확률 및 적용률 측정

검색 결과로 검색어와 얼마나 적합한 결과가 나왔는지에 대해서 정보검색 학문의 정통적인 방법인 정확률(Precision)[22]을 이용하여 평가해 보았다. 또한 재현율의 경우에는 미리 적합한 페이지들을 준비해 놓고 비교해 보아야 하는데, 웹로봇에 의해 임의로 저장된 2만 건의 블로그 페이지 중에서 특정 검색어와 관련 있는 페이지를 미리 분류해 놓는 것이 쉽지 않기에, 재현율 대신에 사용자 지향적 척도라 할 수 있는 적용률(Coverage ratio)도 적용하였다. 정확률이 검색된 문헌들 가운데 관련 있는 페이지의 비율이라고 한다면, 적용률은 사용자가 알고 있는 관련 문헌들 중 실제 검색된 관련 페이지의 비율이라고 할 수 있다.

$$\text{재현율} = \frac{\text{검색된 결과 중 실제로 관련이 있는 문헌 수}}{\text{관련 있는 문헌의 총 개수}}$$

$$\text{정확률} = \frac{\text{검색된 결과 중 실제로 관련이 있는 문헌 수}}{\text{검색된 문헌의 총 개수}}$$

$$\text{적용률} = \frac{\text{검색된 결과 중 사용자가 알고 있는 관련 문헌 수}}{\text{사용자가 알고 있는 관련 문헌의 총 개수}}$$

정확률의 경우엔 검색 결과만 갖고 분석을 하면 쉽게 얻을 수 있다. 하지만 적용률의 경우에는 검색 하는 사람이 검색하기 전에 알고 있는 검색어 관련 블로그들의 리스트가 있어야 한다. 이 리스트는 저장된 블로그들을 하나하나 직접 수작업으로 성능 평가에 사용될 검색어에 맞게 분류를 하여 적용률 측정에 이용하였다. 본 연구에서 구현한 블로그 검색 시스템에서 표 4와 같이 야구, 영화, 사진, 맛집, 음악 등 다섯 가지의 주제와 관련된 블로그를 검색해본 결과, 검색 결과의 정확률 평균은 96.8%가 나왔으며, 적용률은 평균 83.4%가 나왔다.

표 4. 정확률 및 적용률 측정 결과
Table 4. Evaluation Results of Precision and Coverage Ratio.

검색어	정확률	적용률
야구	66/68 (97%)	15/16 (93%)
영화	91/100 (91%)	42/48 (87.5%)
사진	115/116(99%)	34/38(89.5%)
맛집	30/31(96.8%)	9/13(69.2%)
음악	61/61(100%)	21/27(77.8%)
평균	96.8%	83.4%

따라서 본 시스템은 검색어를 주제로 한 블로그 글을 찾는 것에 대한 검색 결과 관련도가 높게 구현되었다고 볼 수 있다. 하지만 객관적인 성능평가를 위해서는 타 시스템과 동일한 조건에서 비교를 해야 정확하다고 할 수 있을 것이다. 그리하여 다음 절에서 우리는 타 시스템과의 성능 비교 결과를 분석해 보았다.

5.3 타 시스템과의 비교

이 절에서는 2.3절에서 살펴본 국내 블로그 검색 시스템과 검색 결과를 2010년 6월 19~20일이라는 일자를 기준으로 비교해 보았다. 특정 검색어와 관련된 블로그 글 검색이 아닌, 특정 주제와 가장 적합한 블로그 검색을 찾는 것으로 기준을 삼았다. “블로그앙”이나 “나루”, “올블로그”의 경우 블로그만을 찾는 검색 옵션이 존재하거나, 검색 결과에 블로그 리스트가 출력된다. 실험 방법은 특정 주제를 검색어로 하여 검색 했을 때, 결과로 나오는 블로그 중에서 상위 10개가 정말 해당 주제와 연관성이 높은 블로그인지에 대한 결과를 비교 하였다. <표 5, 6, 7, 8>에서 각 블로그마다 주제에 따른 연관성을 알아보기 위해 각 블로그의 기본 정보와 블로그 글의 양을 이용하여 연관성을 측정하였다. 블로그의 기본 정보로부터 해당 주제가 그 블로그와 연관성이 높은 주제라고 할 수 있는지를 판단하기 위해, 블로그 제목이나 설명, 카테고리, 페이지에 표시된 상위 태그 등에서 해당 주제와 직접적인 관련이 있는 단어가 2회 이상 존재 할 때 해당 블로그가 그 주제에 대한 연광성이 있다고 판단하였다. 또 한, 각 블로그의 글의 양과 비율을 측정하여 “상”, “중”, “하”로 구분하였는데 이는 해당 주제가 글의 제목이나 태그, 소제목 등에 존재 할 경우, 그 주제와 직접적인 관련이 있는 글로 보고, 해당 글들이 블로그에 50개 이상 존재 하고 해당 글이 차지하는 비율이 20%이상일 경우에는 글 양을 ”상“으로 판별, 해당 연관 글 수가 20개 이상이고 차지하는 비율이 20% 이상이면 ”중“, 그 이하는 ”하“로 판정 하였다. 하지만 블로그 기본 정보에서 해당 주제와 연관성

이 있는 것으로 판별이 되어도 블로그 자체 글의 양이 너무 적거나, 기본 정보와의 연관성도 없고 글의 수도 보통 이하 일 경우에는 해당 블로그와 주제는 서로 적합하지 않은 블로그로 평가하였다. <표 5>에서 보이듯이, "야구"를 검색어로 하여 야구라는 주제와 관련된 블로그 검색 결과 10개 중에서, 1번 블로그의 경우 연관글의 양이 "상"이며, 블로그 정보와 "야구"와의 연관성도 있는 것(O)으로 판별이 되었기 때문에 1번은 "야구"와 관련성이 높은 블로그로 최종 판별이 되어, 블로그 주소가 표에서 굵게 표시 되었다. <표 5>의 3번처럼 블로그 정보에서 주제와의 직접적인 연관을 찾을 수 없는 경우에도 연관 글의 양에서 측정된 연관성이 "상"일 경우에는 해당 블로그는 그 주제와 연관성이 높은 블로그로 최종 판별이 된다. 하지만 <표 5>의 8번 블로그처럼 블로그 정보와의 연관성이 없고(X), 연관 글의 양 또한 보통 이하일 경우에는 이 블로그는 해당 주에와의 연관성이 낮은 것으로 최종 판별이 된다. 이렇게 하여 검색 결과 순위 상위 10개의 블로그에 대한 "야구" 주제와의 연관성을 각각 판별한 결과는 <표 5>의 가장 아래의 행에서 나타나 있듯이 10개 중에서 9개 , 즉 90%가 연관성이 높은 블로그, 즉 해당 주제와의 적합성이 높은 블로그가 된다. 검색어를 "야구"로 입력했을 때 나오는 결과로 측정한 결과 본 연구에서 구현한 시스템의 결과는 적합성이 90%로 높게 나온 것이다.

표 5. 상위 결과 10 블로그와 "야구" 주제와의 적합성 - 본 연구
Table 5. Compatibility of Top 10 Blogs Results with Topic, "baseball" - This Study

	블로그	블로그 정보	연관글
1	twooutsowhat.tistory.com	O	상
2	yagoora.textcube.com	O	상
3	doomhammer.co.kr	X	상
4	hitting.kr	O	상
5	conodont.egloos.com	X	상
6	hyunby1986.tistory.com	X	상
7	mlbspecial.net	O	상
8	seo345.egloos.com	X	중
9	yaguyagu.egloos.com	O	중
10	pinkegobox.egloos.com	O	상
적합한 블로그 수		9/10 (90%)	

표 6. 상위 결과 10개 적합성 - 블로그앎 블로그검색
Table 6. Compatibility of Top 10 Blogs Results with Topic, "baseball" - Blogyam

	블로그	블로그 정보	연관글
1	glwwf.egloos.com	X	중
2	doomhammer.co.kr	O	상
3	blog.naver.com/generlst	O	상
4	natsue.egloos.com	X	중
5	blog.naver.com/cocogus	X	중
6	tokusatsu.egloos.com	X	상
7	blog.naver.com/jjun8204	O	하
8	blog.naver.com/hcr333	X	중
9	blog.naver.com/coriyan	X	하
10	blog.naver.com/acchaa	O	상
적합한 블로그 수		4/10 (40%)	

표 7. 상위 결과 10개 적합성 - 나루 검색
Table 7. Compatibility of Top 10 Blogs Results with Topic, "baseball" - Naroo

	블로그	블로그 정보	연관글
1	yagoora.textcube.com	O	상
2	kjdspecial.egloos.com	X	중
3	mlbspecial.net	O	상
4	blog.naver.com/dhp1225	O	상
5	quixote80.textcube.com	X	중
6	mecklen.egloos.com	X	하
7	dearsanta.tistory.com	O	상
8	blog.naver.com/pcrang01	O	상
9	greenzaku.egloos.com	O	중
10	masakhee.egloos.com	X	상
적합한 블로그 수		7/10 (70%)	

"블로그앎"의 경우는 40%밖에 되지 않는데 그 이유는, 검색어에 적합한 블로그를 보여주기 보다는, 블로그 자체의 블로그앎 랭킹이 높은 순서로 보여주기 때문이었다. "올블로그"의 경우에는 60%였는데, 상위 10개 블로그 모두 블로그 제목이나 설명이 야구와 관련된 블로그였지만, 콘텐츠의 축적된 양이 너무나 적은 블로그들이 40%나 되었다. 이 블로그들은 상위 결과라고 하기에는 부적합하다. "나루"검색의 경우에는 70%라는 보통 수준의 성능이었다. 이와 같은 방법으로 다른 검색 키워드들의 결과를 측정하여, <표 9>와 같은 최종 결과를 얻을 수 있었다.

표 8. 상위 결과 10개 적합성 - 올블로그 블로그검색
Table 8. Compatibility of Top 10 Blogs Results with Topic, "baseball" - Allbog

	블로그	블로그 정보	연관글
1	twooutsowhat.tistory.com	O	상
2	blog.monawa.com/kbo09	O	중
3	edge27.egloos.com	O	상
4	hitting.kr	O	상
5	blog.naver.com/garcia14	O	하
6	blog.naver.com/haleejjang	O	중
7	irjj.egloos.com	O	중
8	yagoosarang.tistory.com	O	하
9	yagoora.textcube.com	O	상
10	sports24.tistory.com	O	하
적합한 블로그 수		6/10 (60%)	

표 9. 5가지 주제에 대한 시스템별 상위 10개 결과의 적합성 비교
Table 9. Compatibility of Top 10 Blogs Results with 5 Topics

검색어	본 연구	블로그얌	나루	올블로그
야구	0.9	0.4	0.7	0.6
영화	0.9	0.7	0.8	0.8
사진	0.9	0.5	1	0.44
맛집	0.7	0.6	0.6	0.5
음악	0.9	0.7	0.9	0.5
평균	0.86	0.58	0.8	0.57

<표 9>과 같은 검색어 5가지 (야구, 영화, 사진, 맛집, 음악)를 검색어로 하여 검색한 결과 중 상위 10개 블로그 결과를 각 시스템 별로 적합성을 비교해 본 결과 평균 수치가 86%인 본 연구 결과가 가장 높았고, 그 다음으로는 나루 검색이 80%였으며, 블로그얌과 올블로그의 블로그 검색은 이보다 낮은 수치가 나타났다. 본 연구에서 구현한 블로그 검색 시스템에서는 검색 결과의 상위 10개 중 8~9개는 해당 주제와 적합한 블로그로 기대할 수 있으며, 다양한 주제에서 동일한 품질의 결과를 기대하기 위해서는 DB에 축적된 블로그의 수도 중요하지만, 무엇보다도 블로그 당 수집된 글의 수가 가장 중요하다. 하지만 이번 성능평가에서는 블로그 당 평균 글 페이지 수가 8개에 불과 했으며, 그 편차가 심하였다. 만약 기간을 충분히 두고 블로그별로 좀 더 충분한 양의 글들을 수집하여 평균 50개 이상의 글들을 수집하게 된다면 더욱 더 높은 품질의 결과를 다양한 주제에서 기대할 수 있을 것이다. 또한 이 경우에는 블로그에서 해당 주제가 차지하는 상대적 비중을 수치화하여 주제-랭킹 함수에 추가적으로 반영할 수가 있다. 또한 주제에 대응되는 색인어인 주제어에 대한 추출 방법을 제목, 태그, 본문에서의 중요도, 앵커 텍스트 외에도 더 효율적이고 정확한 방법을 사용하게 된다면 성능이나 수집 속도를 더 끌어 올릴 수 있을 것이다. 그리고 블로그 글들을 글쓴이가 자체 분류할 때 사용하는 "카테고리"에 대한 검색 시스템의 활용이 나[23][24], 태그에서의 불필요한 용어 제거 등의 문제도 더 나은 블로그 주제 추출을 위한 과제로 남아 있다.

VI. 결 론

블로그 검색에서의 가장 중요하게 고려해야 할 것은 각 블로그가 가지고 있는 글에서 주제어를 뽑아내어 그 블로그가 자주 다루는 주제를 순서화 하여 블로그 검색 결과에 반영해야 한다는 점이다. 블로그가 자주 다루가 인기가 있는 주제는 블로그의 태그 구름[4]에 나타나기도 한다. 하지만 본 연구에서는 블로그의 글의 제목과 태그, 앵커 텍스트(AnchorTexts)에서 주제어를 1차 추출하였고, 블로그 글 본문에서 추출한 색인어들의 피벗 역문헌빈도를 가중치가 높은 용어들을 2차적인 주제어로 추출하였다. 이렇게 블로그별로 주제어를 따로 색인화 해두어, 블로그 검색 시, 해당 주제어와 가장 적합한 블로그로 랭킹을 매길 수 있다. 또한 블로그는 고유의 RSS주소가 있기 때문에, 이것을 잘 활용한다면 웹로봇이 추후 업데이트를 위해 재방문 할 필요가 없이 RSS를 통해 최신 데이터를 수집할 수 있어서, 한번 수집된 주소들을 바탕으로 지속적인 데이터 축적이 용이하고, 이렇게 축적된 데이터가 블로그별로 그 양이 충분하게 된다면 각 블로그에서 자주 다루는 주제들을 정확히 추출할 수가 있게 되며, 이 주제어들은 블로그를 검색할 때 검색 결과로서 단순히 블로그 페이지 리스트를 보여주는 것이 아닌, 해당 주제와 관련성이 높고, 해당 주제를 비중 있게 다룬 블로그를 상위 결과로 찾을 수 있는 블로그 검색 시스템을 구축할 수 있다.

5) 태그 구름(Tag Cloud)은 메타 데이터에서 얻어진 태그들을 분석하여 중요도나 인기도등을 고려하여 시각적으로 늘어놓아 웹 사이트에 표시하는 것으로, 관련된 글과 바로 접근할 수 있는 링크 기능까지 포함되어 있다.[25]

참고문헌

[1] Kumar, R., Novak, P., Raghavan, S. and Tomkins, A, "Structure and evolution of the Blogspace," Communication of the ACM, Vol. 47, No. 12, 2004.

[2] Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C., "Topic sentiment mixture: modeling facets and opinions in weblogs," In Proceedings of the 16th international Conference on World Wide Web, WWW '07. ACM, New York, pp. 171-180, 2007.

[3] Chris Anderson. "The long tail : why the future of business is selling less of more," New York : Hyperion, 2006.

[4] Won-Seok H, Young-Joo D, Duck-Ho B. and Sang-Wook K, "Post Ranking Algorithms in Blog Environment," Proc. of the KIISE Korea Computer Congress 2008, Vol. 35, No. 1(C), pp. 189-193, 2008 June.

[5] Apostolos Kritikopoulos, Martha Sideri, and Iraklis Varlamis. "Blogrank: ranking weblogs based on connectivity and similarity features," AAA-IDEA '06, pp. 8, 2006.

[6] Jung-Hoon Kim, Tae-Bok Yoon, Kun-Su Kim, Jee-Hyong Lee, "Trackback-Rank: An Effective Ranking Algorithm for the Blog Search," IITA, vol. 3, pp. 503-507, 2008.

[7] Kangmiao Liu, Guang Qiu, Jiajun Bu, Chun Chen, "Ranking Using Multi-features in Blog Search," Advances in Multimedia Information Processing - PCM 2007, pp. 714-723, 2007.

[8] Junghoon K. Taebok Y. and Jeehyong L, "The Blog-Rank algorithm for the effective blog search," Proc. of the 35th KIISE Fall Conference, Vol. 35, No. 2(A), pp. 93-94, 2008 October.

[9] Y. Wu and B.L. Tseng, "Important Weblog Identification and Hot Story Summarization," In Proceedings of AAAI Computational Approaches to Analyzing Weblogs, pp. 221-227, 2006.

[10] Won-Seok Hwang, Sang-Wook Kim, Duck-Ho Bae, Young-Joo Do, "Post Ranking Algorithms in Blog Environment," Future Generation Communication and Networking Symposia, International Conference on, vol. 2, pp. 64-67, 2008.

[11] Jie Shen, Yan Zhu, Hui Zhang, Chen Chen, Rongshuang Sun, Fayan Xu, "A Content-Based Alg- orithm for Blog Ranking," International Conference on Internet Computing in Science and Engineering, pp. 19-22, 2008.

[12] Ko Fujimura, Takafumi Inoue, Masayuki Sugisaki, "The EigenRumor Algorithm for Ranking Blogs," WWW 2005 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2005.

[13] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S Yu, "Identifying the Influential Bloggers in a Community," In WSDM '08: Proceedings of the international conference on Web search and web data mining, pp. 207-218, 2008.

[14] Herring, S. C., Kouper I., Paolillo J. C., Scheidt, L. A., Tyworth M, Welsch P., Wright E. & Yu N., "Conversations in the Blogosphere: An Analysis "From the Bottom Up," PHICSS-38, 2005.

[15] Junghoon K. Taebok Y. and Jeehyong L, "The Effective Blog Search Algorithm based on the Structural Features in the Blogspace," Journal of KIISE : Software and Applications, Vol. 36, No. 7, pp. 580-589, 2009 July.

[61] Hyeonil S., Unil Y. and Keun H. R., "Efficient Blog Retrieval System by Topic-based Weighting," Journal of the Korea Society of Computer and Information, Vol. 15, No. 4, pp. 1-9, 2010 April.

[17] Dou, Z., Song, R., Nie, J., and Wen, J., "Using Anchor Texts with Their Hyperlink Structure for Web Search," In Proceedings of the 32nd international ACM SIGIR '09., New York, pp. 227-234, 2009.

[18] Jae-Yun L., "A Study on the Pivoted Inverse Document Frequency Weighting Method," Journal of the Korea Society for Information Management, Vol. 20, No. 4, pp. 233-248, 2003 December.

[19] Seung-Shik K., Hagyu L., So-Hyun S., Gi-Choi H. and Byung-Joo M., "Term Weighting Method by Postposition and Compound Noun Recognition," Proc. of the 28th KIISE Fall Conference, Vol. 28, No. 2, pp. 196-198, 2001 October.

[20] Ben Adida, "hGRDDL: Bridging microformats and RDFa,"

Web Semantics : Science, Services and Agents on the World Wide Web, Vol. 6, No. 1, pp. 54-60, 2008.

[21] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The Pagerank Citation Ranking: Bringing Order to the Web," Technical report, Stanford Digital Library Technologies Project, 1998.

[22] A. Borodin and R. Gareth, S. Jeffrey and T. Panatiotis, "Link Analysis Ranking - Algorithms, Theory, and Experiments," ACM Trans. on Internet Technology, Vol. 5, No. 1, pp. 231-297, 2005.

[23] Bok-Keun S. and Da-Hyun W. and Kwang-Rok H., "A Study on Paper Retrieval System based on OWL Ontology," Journal of The Korea Society of Computer and Information, Vol. 14, No. 2, pp. 169-180, 2009 February.

[24] Unil Y. Hyeonil S. and Keun H. R., "Intelligent Retrieval System for Finding Important Travel Information," Journal of The Korea Society of Computer and Information, Vol. 14, No. 11, pp. 113-121, 2009 November.

[25] Dunam K., Kangpyo L. and Hyoung-Joo K., "Improved Tag Selection for Tag-cloud using the Dynamic Characteristics of Tag Co-occurrence," Journal of KIISE : Computing Practices and Letters, Vo. 15, No. 6, pp. 405-413, 2009 June.

저 자 소 개

신 현 일

2009. 8: 충북대학교 컴퓨터공학 학사.

2009. 9 - 현재: 충북대학교 컴퓨터공학 석사

관심분야: 데이터마이닝, 정보검색, 데이터베이스

Email: shoner@chungbuk.ac.kr

윤 은 일

1997: 고려대학교 이학석사.

1997 - 2006: 한국통신 멀티미디어 연구소 전임/선임연구원.

2005: Texas A&M Univ. 공학박사

2005 - 2006: Texas A&M Univ. 포스닥연구원.

2006 - 2007: 한국전자통신연구원, 선임연구원.

2007 - 현재: 충북대학교 전자정보대학 컴퓨터전공 조교수

관심분야: 데이터마이닝, 정보검색, 데이터베이스

Email: yunei@chungbuk.ac.kr

류 근 호

1976: 숭실대학교 공학사.

1980: 연세대학교 공학석사.

1980 - 1983: 한국전자통신연구원 연구원.

1983 - 1986: 한국방송통신대학교 조교수.

1988: 연세대학교 공학박사

1986 - 현재: 충북대학교 전자정보대학 컴퓨터전공 교수

관심분야: 데이터베이스, 데이터마이닝, 바이오인포매틱스

Email: khryu@dblab.chungbuk.ac.kr