韓國컴퓨터情報學會 論文誌
第15卷 第4號, 2010. 4.

2010-15-4-1-1

주제어 가중치 기법에 의한 효율적인 블로그 검색 시스템

신 현 일*, 윤 은 일**, 류 근 호**

Efficient Blog Retrieval System by Topic-based Weighting

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

요 약

"Web 2.0"으로 불리는 새로운 세대의 웹에서, 블로그를 통하여 누구나 손쉽게 정보나 의견을 세상에 알릴 수 있게 되었고 이러한 블로그를 효과적으로 검색하기 위해서 블로그의 특성을 고려한 검색 알고리즘들이 새롭게 제안이 되고 있다. 그러나 실제 블로그 검색 시스템에 적용된 키워드 기반 검색이나 블로그간의 링크 분석을 통한 랭킹만으로는 사용자가 기대하는 성능을 발휘하지 못한다. 본 논문에서는 검색 결과를 향상시키기 위해 블로그 글과 검색어와의 연관성을 고려한 주제어 가중치 기반의 블로그 검색 시스템을 제안한다. 제안된 시스템은 블로그 글 마다 주제어(Topic)를 추출하여 색인어보다 더 높은 가중치를 부여한다. 기존 시스템과의 비교에서 제안된 방법이 실제 검색 결과에서 재현율이 향상됨을 알 수 있었다.

Abstract

In the new generation of Web, commonly called "Web 2.0", blogging has facilitated the publishing information or his/her opinion on the web. Various blog retrieval algorithms have been proposed to search for blogs more effectively. However, actually keyword-based searching or link-analysis blog ranking system cannot satisfy the user's requirement. In this paper, we suggest a topic-based weighting blog retrieval system in which the links between blog writings and searching words are considered to improve the search results. Our system extracts topics from each blog and weights them much higher than other guide words. In the comparison with other systems, we see that the proposed topic-base system has better recall rate of search results.

▸ Keyword : 블로그(Blog), 검색(Retrieval), 주제어 가중치 색인(Topic weighted Index)

• 제1저자 : 신현일, 교신저자 : 윤은일
• 투고일 : 2009. 12. 02, 심사일 : 2010. 01. 05, 게재확정일 : 2010. 03. 08.
* 충북대학교 컴퓨터공학 석사 **충북대학교 전자정보대학 컴퓨터전공 교수
※ "이 논문은 2009학년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음(This work was supported by the research grant of the Chungbuk National University in 2009)" 또한 "이 논문은 2010년 교육과학기술부로부터 지원받아 수행된 연구임" (지역거점연구단육성사업 / 충북BIT연구중심대학육성사업단)

Ⅰ. 서 론

블로그는 개인의 생각이나 관심사에 대한 정보를 일지 형태로 기록해 두는 웹사이트 이지만 타인과의 커뮤니케이션 또한 가능하다는 특징이 있다.[1][2]. 따라서 블로그 검색에서는 검색어와 관련된 같은 결과라도 그 블로그가 얼마나 더 사람들에게 인기가 있고, 사용자들의 반응이 얼마나 더 활발하게 이루어지는 페이지이냐에 따라 그 순위를 다르게 매길 수밖에 없다. 그러나 이러한 실제 검색 시스템에서 사용자가 원하는 것은 검색어와 가장 적합성이 높은 블로그 글이 검색 결과의 상단에 나올 것임을 기대할 수 있어야 한다. 그러나 인터넷에서 서비스 되고 있는 블로그 검색 시스템을 실제로 사용해보면, 검색어와 해당 블로그 글과의 관련은 낮은데도 불구하고, 블로그 자체의 권위(authority)[3]나 명성(reputation) [3]이 높다는 이유로 검색 결과의 상위에 랭크가 되는 등의 문제점이 발생된다. 본 논문에서는 이러한 문제를 해결하기 위해 주제어 가중치 기반 블로그 검색 시스템을 제안한다. 이 새로운 블로그 검색 시스템은 각 블로그 글에서 단어를 추출하여 색인할 때, 주제어를 따로 추출하여 그것에 더 높은 가중치를 부여함으로써, 블로그 글과 주제어간의 연관성을 높이고 검색 결과의 성능을 향상 시킨다.

본 논문의 2장 관련 연구에서는 블로그 검색과 일반 웹페이지 검색의 차이는 무엇이며, 기존 시스템의 문제는 무엇인지 알아보고 이를 바탕으로 3장에서 향상된 블로그 검색 시스템을 구현한다. 4장에서는 기존 시스템과 성능 평가를 하고 그 결과를 분석한다.

Ⅱ. 관련 연구

2.1. 블로그 검색의 특성

웹 검색에서는 웹 문서의 중요도를 평가하는 기준으로 키워드와 웹 문서간의 유사도만을 고려하는 것이 아니라 추가적인 정보를 검색 랭킹에 활용하는 다양한 방법들이 제안되고 있다. 그 중에서도 구글 검색 엔진에 사용된 PageRank[4] 알고리즘과 같이 웹 문서간의 하이퍼링크를 통해 웹 문서간의 참조나 인용을 통해 상대적 중요도에 따라 가중치를 부여하는 링크 분석 방법이 가장 유명하며, 오늘날 대부분의 웹 검색 랭킹의 기초 이론으로 사용하기에 이르렀다. 그러나 이 방법은 실제 블로그 검색에서는 효과적이지 못하다. 그 이유는 대부분의 블로그 글 간의 하이퍼링크를 통한 참조가 일반 웹페이지에서 나타나는 것에 비해 훨씬 적기 때문이다[5]. 일반적으로 웹페이지들은 회사나 유명인사 같이 특정 대상을 직접 가리키거나 그에 상응하는 정보를 제공하는 역할을 하지만, 블로그의 경우에는 개인의 생각과 관심 정보를 기록해 놓은 곳이라는 차이가 있기 때문에 웹 문서 검색 랭킹을 적용이 어렵다. 그리하여 블로그만의 특성에 맞춘 새로운 방법들이 제안되었다. BlogRank[6] 알고리즘의 경우에는 PageRank를 블로그 특성에 맞게 수정을 하여 블로그 간의 유사성과 접속 용이성을 분석하여 하이퍼링크 랭크 기법을 사용하였고, B2Rank[7] 알고리즘은 기존의 하이퍼링크를 통한 랭킹에 추가적인 요소로 블로그 글이 올라오는 빈도수, 글에 대한 댓글 수를 일정한 시간에 맞게 수치화 하여 적용을 하였다. EigenRumor[3] 알고리즘과 이를 더 발전시킨 블로그 에고센트릭(Egocentric) 검색[8]의 경우 블로그 글과 사용자 사이의 댓글(comment)이나 블로그의 글 사이에 연결된 원격 댓글(trackback)들의 양을 고유벡터로 사용하여 블로그 자체의 매력(attractiveness)[3]과 블로그의 각 글들에 대한 사용자의 평가를 수치화하여 블로그 랭킹에 적용한 알고리즘들이다. 그 밖에도 블로그 글에 대해 사용자들의 긍정적인 반응내지는 관심으로 볼 수 있는 스크랩, 트랙백, 댓글 등의 행동들을 하이퍼링크를 대체하는 측정 요소로 사용한 포스트 랭킹 알고리즘[9]도 제안이 되었다. 이와 같이 지금까지 제안된 블로그 랭킹 알고리즘들은 모두 블로그의 권위(authority)나 명성(reputation) 등의 특성을 이용하여 측정하는 것에 초점이 맞추어져 있었다. 하지만 실제 검색 시스템에서는 블로그 자체의 매력(attractiveness)뿐만 아니라, 검색어와의 연관성 또한 검색 결과의 재현율이나 정확률을 좌우하는 중요한 요소이다. 따라서 좋은 검색 결과를 얻기 위해서는 검색어와 관련성 높으면서 권위도 높은 블로그의 글에 가중치를 주는 랭킹 알고리즘이 필요하다.

2.2. 국내의 블로그 검색 시스템

국내에서 서비스되고 있는 블로그 전문 검색 시스템으로는 올블로그[10]의 블로그 검색, 블로그앰[11]의 블로그 검색과 나루 검색[12]이 있다. 이 3가지 블로그 검색 시스템은 2.1절에서 언급한 블로그 검색 랭킹 알고리즘들의 특징과도 유사하게, 블로그의 인기도를 랭킹의 주요소로 사용하고 있다. 그러나 이러한 블로그 검색 시스템은 제한점이 있다. 검색어와의 관련성이 높지 않은 글인데도 불구하고, 블로그 자체가 인기가 높아도 상위 결과에 랭크가 되는 경우가 나타난다. 이런 결과가 발생하는 이유는 블로그 글의 주제와 상관없이 블로그 자체의 데이터양이나 사용자들과의 활발한 교류만으로 블로그의 권위(authority)가 높아지고 그것이 그대로 키워드 검색

결과 순위에 반영이 되기 때문이다. 블로그와 검색 키워드 간의 적합성을 높이기 위해서는 검색 키워드가 해당 블로그 글을 대표할 수 있는 역할을 갖고 있어야 하며, 그러기 위해서는 키워드의 대표성에 따라 다른 비중으로 색인어 랭킹이 계산되어야 한다. 따라서 본 논문에서는 이러한 제한점을 극복하고 더 나은 블로그 검색 성능을 보여주는 블로그 검색 시스템을 구현하기 위하고자 블로그에서 추출할 수 있는 색인어 중에서 블로그 글의 내용을 대표할 수 있는 주제어를 따로 추출하여, 일반 색인어보다 더 높은 가중치를 부여하는 주제어 기반 블로그 검색 시스템을 제안한다.

III. 블로그 검색 시스템 구현

블로그 검색 시스템은 그림 1과 같이 사용자 질의 서버, 데이터베이스 시스템, 웹 로봇의 세 부분으로 나눌 수 있다. 웹 로봇은 많은 블로그들이 모인 허브(hub)와도 같은 블로그 포털 사이트를 시작으로, 링크를 돌며 각 블로그를 순회하며 블로그 글들과 대응되는 블로그 페이지들을 수집하여 색인한다.

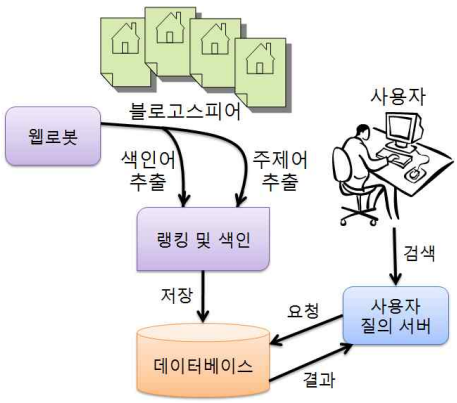

그림 1. 블로그 검색 시스템 구성도
Fig. 1. Blog Retrieval System Diagram

웹 로봇은 블로그 글들 중에서 수집해야 할 대상을 선별한 후, 블로그 글의 고유주소, 제목, 색인어 등을 추출하여 색인 모듈에 넘기고 자기 자신이 재귀적으로 다른 페이지들을 지속적으로 방문할 수 있게 하이퍼링크들을 따로 추출하여 메모리에 저장을 한다. 데이터베이스 시스템은 웹 로봇이 색인 모듈에 페이지 소스를 넘겨서 추출된 정보들과 랭킹 함수에 의해 계산된 점수를 웹 서버에 데이터베이스 형태로 저장을 하고 관리하는 역할을 한다. 마지막으로 사용자 질의 서버는 사용자가 입력한 검색어를 질의어로 변환한 다음 데이터베이스에서의 검색 결과 목록을 랭킹 점수의 내림차순으로 정렬하여 웹페이지에 출력하는 역할을 한다.

3.1 웹 로봇과 색인

웹 로봇(Web Robot)은 실제 웹사이트들을 링크를 타고 돌면서 페이지 정보를 수집하여 검색 엔진의 색인과정에서 사용할 메타 데이터들을 수집 한다. 블로그 검색 시스템에서의 웹 로봇의 작동 원리는 그림 2와 같다. 최초에 입력된 URL 주소를 시작으로 해당 페이지에 링크된 다른 웹 주소들을 깊이 우선 탐색(depth-first search)을 하면서 URL 주소들을 수집하는 과정이 필요하고, 이렇게 수집한 웹페이지 중에서 아직 색인되지 않은 웹페이지만을 선별 한 후에, 페이지 소스를 파싱(parsing)하여 블로그 페이지의 구조를 분석하고 각 구조별로 필요한 데이터를 추출하게 된다[13]. 이 때 글 본문과 제목, 블로그 태그 중에서 중요도가 높은 색인어를 해당 블로그 글의 주제어로 추출하고, 본문의 나머지 부분에서는 중요도가 상대적으로 낮은 색인어들을 추출한다. 또한 날짜나 제목, 주소와도 같은 블로그 글의 기본정보도 따로 추출하여 데이터베이스에 저장하며, 마지막으로 하이퍼링크에서 새로 방문할 웹 주소들을 추출하여 페이지 목록에 저장하고 이미 방문한 페이지 주소는 목록에서 지워가며 다시 시작 한다. 일련의 과정들은 더 이상 읽어 들일 페이지 주소가 없을 때까지 재귀적으로 반복한다.

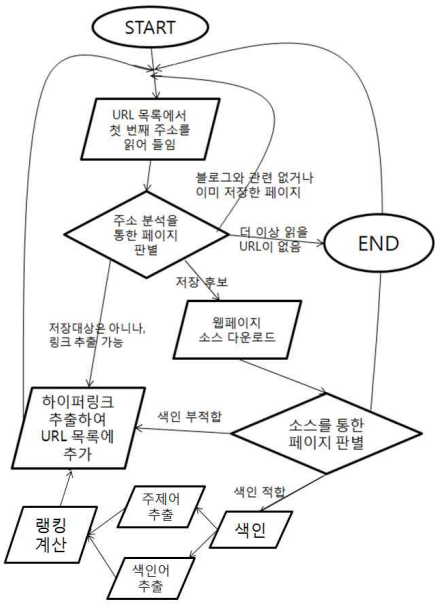

그림 2. 웹 로봇의 작동 원리
Fig. 2. Web Robot Mechanism

주제어 추출을 위한 쉬운 방법은 글의 제목을 이용하는 것이다. 하지만 글의 제목만으로는 글의 주제어를 알 수 없는 경우가 존재하고, 글의 제목과 맞지 않게 더 중요한 주제어가 본문에 존재 할 수 있으므로, 제목만으로는 주제어를 추출하는데 제한점이 있다. 보다 정확한 방법은 글의 본문 또한 문맥에 의해서 어떤 색인어가 글의 주제어가 되는지를 판별해내는 기법을 도입하는 것인데, 이런 기법을 사용하기 위해서는 자연어의 구문적, 의미적 처리 방법[14]이 추가적으로 요구된다. 그와 반대로 처리속도에서 보다 효과적인 방법으로 페이지 형식을 분석하는 방법의 경우, 본문에서의 밑줄이나 굵은 표시로 강조 된 구절에서 주제어를 추출하거나, 격 이론을 이용하여 특정 접속부사로 시작하는 구문을 통해 주제어를 추출할 수 있다[15]. 본 연구에서도 문맥에 의한 방법이 아닌 페이지의 형식에 의한 방법을 통한 주제어 추출방식을 사용하였다. 즉 주제어 추출 방법 중에서 후자 방식을 이용하여, 블로그 글의 제목과 블로그 태그와 글 본문에서 중요하게 표시된 구절에서 블로그 글의 주제어를 추출하는 방법을 사용하였다. 여기서 블로그 태그란 블로그 글 작성자가 글을 자신의 블로그에 올릴 때, 자신이 직접 해당 글의 주제나 소재, 분류에 알맞은 단어를 쉼표 구분으로 입력해 놓는 것을 말한다. 블로그 태그는 전 세계 웹페이지에 공통적으로 적용할 수 있게 규정한 마이크로포맷(microformats)[16]에 따라 블로그 글 본문 하단이나 상단에 rel="tag" 형태의 속성이 들어간 하이퍼링크로 붙여진다. 블로그 태그는 글쓴이(blogger)가 스스로 자신의 글의 주제, 소재 또는 분류와 관련 있는 단어를 저장하지만 오용의 여지가 있다. 따라서 본 연구에서는 블로그 태그를 주제어 추출에 사용하되, 그 비중을 낮게 적용하였다.

```
// 최초 히트페이지를 pages에 저장한 후 모듈 실행
// 웹 로봇 모듈이 실행될 때마다,
// pages에 새로운 URL들이 추가됨
// 웹 로봇 모듈은 pages가 다 비워질 때까지 반복

Module WebRobot(pages) {
 foreach url in pages
   //블로그 글인지 아닌지 판별
   IsBlog = analysis url address
     //(1:맞음 또는 모름, 2:아니지만 링크추출대상,
     // 3:블로그와 관련 없거나 이미 저장된 페이지)
   if IsBlog=3 : goto next url
   if IsBlog=1 (
  Download page by HTTP-protocol(url)
  WhatKind = analysis page //블로그 1차 판별
  //(블로그유형, 1:네이버, 2:다음, 3:이글루스, 4:티스토리,
      // 5:텍스트큐브, 6:워드프레스, 7:기타 블로그, 8:아님)
    if WhatKind=8: 다음 url로 웹 로봇 이동
  //pageIndex 모듈을 불러와서 데이터 추출 및 색인
     pageIndex(다운로드 소스, url, WhatKind) )
   Save Hyperlinks into pages
    // 링크 추출은 깊이 우선 탐색으로 함 }
```

그림 3. 웹 로봇 모듈
Fig. 3. Web Robot Module

실제 구현한 웹 로봇 모듈은 그림 3과 같다. 웹 로봇이 방문한 페이지가 블로그 페이지인지 아닌지는 페이지 주소의 도메인을 통해 식별이 가능할 수 있다. 그러나 개인이나 사업자가 따로 등록한 도메인일 경우에는 웹페이지 소스를 직접 분석해서 판단해야 한다. 워드프레스나 텍스트큐브처럼 특정한 블로그 웹사이트 제작 도구를 통해 만들어진 블로그의 경우에는 웹페이지 소스의 메타데이터를 통해 판별이 가능하며, 꼭 그러한 경우가 아니라도 메타데이터의 주요 정보[17]를 통해 대부분의 블로그가 판별이 가능하다. 또한 블로그가 어떤 유형인지, 네이버 블로그인지 아니면 이글루스 블로그인지에 대한 정보도 블로그 주소나 웹페이지 HTML 소스의 메타데이터로 대부분 구별이 가능하다. 블로그 구별이 완료되면 해당 정보는 색인 모듈(pageIndex)로 웹페이지 주소와 웹페이지 소스와 함께 넘겨진다. 웹 로봇에 의해 호출된 색인 모듈(pageIndex)은 데이터베이스에 블로그 글 정보와 색인어, 색인어별 랭킹 점수 등을 저장하게 되며 이에 대한 실제 구현은 그림 4에 기술되어 있다. 색인 모듈은 각 블로그 유형에 맞게 블로그 페이지 구조를 분석하여, 제목 부분에서는 제목을 추출하고 블로그 태그나 날짜, 본문 그리고 본문의 특정 강조 부분 등을 각각 추출한다. 또한 추출된 데이터는 데이터베이스에 저장될 위치에 맞게 가공이 되어 실제 데이터베이스 질의어에 의해 저장 및 갱신이 이루어지게 된다. 그리고 본문 글의 내용을 대표하는 주제어를 글의 제목, 태그, 다른 중요 구문 등에서 추출하여 각각의 가중치에 맞게 색인어 별로 계산된 후 최종적으로 색인이 된다.

```
// src : 다운로드 된 블로그 페이지 소스
// url : 페이지의 URL 주소
// kind : 블로그 유형의 offset
Module pageIndex(src, url, kind) {
 init a dictionary of (guide word : count), G
  structure analysis for each kind
  abstract topics and Title from the page title
  foreach topic in topics
       t = 1 * title topic weight
```

```
        append (topic : t) to G
    abstract Date
    in matter sentences, // 중요 구문
        abstract topics & its frequency
            foreach topic in topics
                t = frequency * each topic weight
append (topic : t) to G
    in the others, //글 본문의 나머지 부분
        abstract guide words & its frequency
            foreach guide word in guide words
                g = frequency * guide word weight
                append (guide word : g) to G
    foreach (g : c) in G
        calculate g's score by ranking function
        insert g's score of this page into database
        insert other page information & relation  }
```

그림 4. 페이지 색인 모듈
Fig. 4. Page Indexing Module

색인된 데이터들은 GwordList, GwordInfo, UrlList 테이블에 각각 저장이 되고, 이들은 데이터베이스 관리 시스템에서 관리한다. GwordList 테이블에는 색인어가 Gword라는 이름의 필드에 저장이 되며, GwordInfo 테이블에는 실제 블로그 페이지에서 추출된 각 색인어의 랭킹 점수가 Score 필드에 저장이 되고, 추출된 페이지와 색인어의 식별자(ID)가 UrlId와 GwordId라는 이름의 외래키로 저장이 된다. 블로그 페이지들은 UrlList 테이블에 저장이 되며, 페이지의 기본 정보인 제목, 주요 주제어, 날짜가 각각의 필드(Title, Topics, Date)에 저장이 된다.

3.2 랭킹 함수 및 사용자 질의 서버

사용자 질의 서버에서는 입력 받은 검색어를 GwordList 테이블에서 찾아내어 색인어의 식별자를 역 인덱스로 받아온다. 그리고 GwordInfo 테이블에서 해당 인덱스와 관련된 URL 주소 식별자를 페이지 별 색인어 랭킹 점수의 내림차순으로 가져온 후 마지막으로 UrlList 테이블에서 실제 페이지 정보를 가져와서 질의 결과로 보여주게 된다. 검색 결과로는 블로그 글의 제목, 날짜, 페이지링크 뿐만 아니라 글과 관련된 주요 주제어를 함께 출력하여 사용자가 자신이 원하는 블로그 글을 쉽게 찾을 수 있도록 인터페이스를 구현하였다. 글과 관련된 주제어들은 블로그 페이지에 나타난 블로그 태그, 블로그 제목, 블로그 본문에서의 강조 구문(소제목, 밑줄, 굵은 폰트 등)과 다른 웹페이지를 참조하는 하이퍼링크에 대한 설명인 앵커텍스트(anchortext)에서 단어를 추출하여 색인어로 저장하였으며, 각 색인어별로 가중치에 비례한 용어 빈도(df)를 적용하고, 페이지에서의 해당 색인어 k의 비중인 rate(k)와 df(k)를 각각 GwordList 테이블에 저장하였다. 또한 데이터 수집을 위한 웹 로봇의 탐색이 완료 된 후에, 저장된 전체 페이지의 총 수와 색인어 k를 포함하고 있는 문헌의 수를 계산하여 역문헌빈도(idf)[18]를 산출하고, 최종적으로 색인어 k의 랭킹 점수(score)를 산출하여 저장을 한다. 특정 페이지에서의 색인어 k의 랭킹 점수를 나타내는 score(k)는 식score(k)=df(k)*rate(k)*idf(k)에 의해 계산이 되며, 색인어 k의 가중치 df(k)와 페이지에서의 비중 rate(k)와 역문헌빈도 idf(k)의 자세한 계산 방법은 그림 5와 같다.

score(k) = df(k) * rate(k) * idf(k)

df(k) = Wtitle + Wtag
+ Wanchortext*Fanchortext
+ Wheadline*Fheadline + Wbold*Fbold
+ Wunderline*Funderline + Wbody*Fbody

rate(k) = df(k) / (페이지 전체 색인어의 df 합)

$idf(k) = \log_2 N - \log_2 dk + 1$

df(k) : 색인어 k의 용어 빈도
Wx : x 위치에서의 색인어 가중치
(i.e. Wtitle : 제목에서 추출된 색인어의 가중치)
Fx : x 위치에서의 색인어 빈도수
(i.e. Fbody : 본문에서 추출된 색인어의 빈도수)
N : DB에 저장된 전체 페이지 수
dk : 색인어 k와 관련된 페이지 수

그림 5. 랭킹 함수
Fig. 5. Ranking Function

실제로 사용자 질의 서버에 검색어 g가 입력된다고 가정하면, 검색어 g는 데이터베이스 저장 및 검색에 적합한 명사 형태로 색인어화 되고, 데이터베이스의 색인어 테이블 GwordList에서 매칭 되는 색인어를 찾아서 해당 색인어가 존재할 경우에만 색인어의 ID인 gID를 GwordInfo 테이블에서 받아온 후, gID와 연결되어 있는 블로그 페이지의 식별자인 bID 모두를 UrlList 테이블에서 받아오는 과정을 거치게 된다. 이 때 페이지 식별자마다 가지고 있는 색인어의 랭킹 점수가 높게 나타난 순서로 페이지 식별자 목록을 받게 되며, 이 것을 기반으로 UrlList 테이블에서 해당 식별자와 매칭 되는 페이지의 정보를 받아 온 후, 목록 순서대로 사용자 질의 서버로 다시 넘겨, 검색 결과를 웹에 출력한다. 구현한 사용자 질의 서버의 모습은 그림 6과 같은 형태이다.

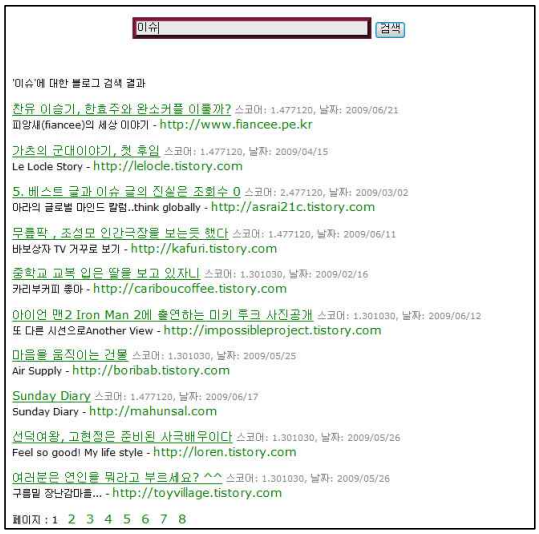

그림 6. 검색 결과 화면
fig. 6. Result Display

IV. 성능 평가

4.1 실험 환경

본 논문에서 구현한 블로그 주제어 검색 시스템의 실험 환경은 표 1과 같다.

표 1. 실험 환경
Table 1. Experiment Environment

H/W	CPU	Pentium IV 2.6Ghz
	RAM	DDR2 2G Bytes
	HDD	250G Bytes
S/W	OS	MS Windows XP Pro.
	DBMS	MySQL 4.01
	Web Server	Apache 2
	Language	Python 2.6.2
Network	Bandwidth	100M

4.2 실험 방법

각 블로그 검색 시스템간의 성능 비교는 각각의 검색 결과의 적합성에 대한 척도를 통해 비교할 수 있다. 일반적으로 많이 쓰이는 척도에는 정확률(precision)과 재현율(recall)이 있는데, 여기서 재현율이란 적합 글이 얼마나 많이 검색되었는가를 나타내며, 재현율을 측정하기 위해서는 이미 알려진 적합 글들을 집합으로 미리 정해 놓아야 한다. 반면에 정확률은 검색된 글들이 얼마나 적합한가를 나타낸다. 정확률을 통해 합한가를 나타낸성능을 비교하는 경우에는 각 합한가를 나타낸 결과가 많이 다를 경우, 어느 글이 적합 글이고 어느 글이 부적합 글인지에 대한 판단을 계속 해야 하는데, 그 기준이 사용자타낸글에 대한 수용 또는 거절타낸판단 합한으므로 객관적으로 정할 수가 없다. 본 연구에서는 재현율의 비교를 통해 각 블로그 검색 시스템간의 성능을 평가하였다. 사전에 특정 검색어와 적합한 블로그 글 집합을 작성해 두고, 그것을 바탕으로 각 를 나타낸검색 결과의 재현율 측정에 사용하였다. 비교 대상 시스템으로는 올블로그[11]와 나루낸검색[12] 검색 시스템을 선택하였으며, 재현율 측정을 위한 적합 글 집합은 국내 블로그 포털 사이트인 이글루스[19]에서 10일 동안 올라온 글 중에서 글의 주제가 영화 제목 “왓치맨”인 글을 모두 수집한 결과 총 24개의 블로그 글의 목록을 만들 수 있었으며 내용은 표 2와 같다. 각 검색 시스템에서 “왓치맨”을 검색하여 나온 결과의 블로그 글 목록 중에서 표 2의 목록과 일치하는 글의 개수를 각각 측정하였다.

표 2. 영화 “왓치맨”과 관련된 이글루스 블로그 글 목록
Table 2. Egloos.com blog Writings about a movie, "The Watchman"

1	http://wallflower.egloos.com/1880907
2	http://swooya.egloos.com/4875206
3	http://atonal.egloos.com/1880720
4	http://carbuncl.egloos.com/1880710
5	http://halzz.egloos.com/1880663
6	http://ioros.egloos.com/1401802
7	http://wishsong.egloos.com/4085628
8	http://tomino.egloos.com/4085582
9	http://lobo.egloos.com/4874204
10	http://job314.egloos.com/2256890
11	http://tunajuice.egloos.com/1332580
12	http://cyvaster.egloos.com/4873937
13	http://netyhobby.egloos.com/4873932
14	http://onesuck.egloos.com/2293341
15	http://malcolmx.egloos.com/2256545
16	http://okajun.egloos.com/4085011
17	http://darkberry.egloos.com/4225625
18	http://kratos71.egloos.com/1400436
19	http://atonal.egloos.com/1880038
20	http://likeion.egloos.com/2256128
21	http://atrakush.egloos.com/4084624
22	http://primemover.egloos.com/4815895
23	http://draco21.egloos.com/1331577
24	http://bundo.egloos.com/1400120

표 3. 본 연구의 검색 결과
Table 3. Experiment of this work

1	http://cyvaster.egloos.com/4873937
2	http://tunajuice.egloos.com/1332580
3	http://job314.egloos.com/2256890
4	http://lobo.egloos.com/4874204
5	http://tomino.egloos.com/4085582
6	http://wishsong.egloos.com/4085628
7	http://ioros.egloos.com/1401802
8	http://halzz.egloos.com/1880663
9	http://swooya.egloos.com/4875206
10	http://carbuncl.egloos.com/1880710
11	http://atonal.egloos.com/1880720
12	http://swooya.egloos.com/4875206
13	http://dracc21.egloos.com/1331577
결과	13/24

표 4. 올블로그의 검색 결과
Table 4. Experiment of "allblog.com"

1	http://swooya.egloos.com/4875206
2	http://atonal.egloos.com/1880720
3	http://carbuncl.egloos.com/1880710
4	http://wishsong.egloos.com/4085628
5	http://job314.egloos.com/2256890
6	http://onesuck.egloos.com/2293341
7	http://malcolmx.egloos.com/2256545
8	http://atonal.egloos.com/1880038
9	http://likeion.egloos.com/2256128
10	http://atrakush.egloos.com/4084624
11	http://primemover.egloos.com/4815895
12	http://dracc21.egloos.com/1331577
결과	12/24

표 5. 나루 검색의 "왓치맨" 검색 결과
Table 5. Experiment of "naroo.com"

1	http://wallflower.egloos.com/1880907
2	http://carbuncl.egloos.com/1880710
3	http://tomino.egloos.com/4085582
4	http://job314.egloos.com/2256890
5	http://okajun.egloos.com/4085011
6	http://primemover.egloos.com/4815895
결과	6/24

4.3 실험 결과 분석

각 시스템의 실험 결과 중 본 연구에서 구현한 검색 시스템은 표 3의 결과와 같이 적합 글 집합의 24개 중에서 13개를 찾아내어 가장 높은 재현율을 보여주었고, 올블로그도 표 4와 같이 검색 결과 중 12개가 일치하였다. 하지만 나루 검색의 경우 표 5에서 볼 수 있듯이 검색 결과 중 6개만 일치 하였다. 각 결과의 재현율을 백분율로 계산하여 비교해 보면 그림 7과 같으며, 성능 비교를 통해 본 연구에서 구현한 주제어 기반 블로그 검색 시스템의 성능이 가장 우수함을 알 수 있다.

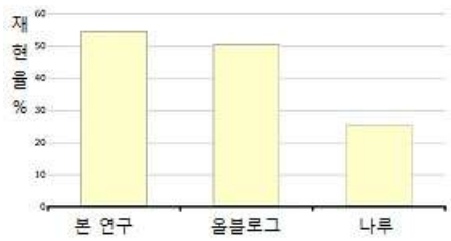

그림 7. 재현율 평가 실험 결과
Fig 7. Recall Ratio Experiment Result

올블로그의 블로그 검색과 나루 블로그 검색의 경우 블로그 글의 제목과 본문에서의 색인어의 빈도수와 블로그 자체의 권위 랭킹만을 검색 결과에 반영한 것에 반해, 주제어 기반 블로그 검색은 블로그 글의 구조를 분석하여 보다 중요하게 사용된 어휘에 더 높은 가중치를 부여 하여 색인하고 랭킹에 적용한 차이가 있으며 이로 인해 검색 결과의 적합성에 차이가 발생하였다.

표 6. 검색 시간 비교 (단위: 초)
Table 6. Retrieval Time Comparison

키워드	나루	올블로그	본 연구
'영화'	1.67	1.76	1.31
'여행'	1.93	1.51	1.98
'선물'	1.78	2.26	1.32
평균	**1.79**	**1.84**	**1.54**

검색 시스템의 중요한 성능 중의 하나인 검색 시간에 대한 세 시스템간의 비교는 표 6의 결과처럼, 모두 1~2초의 시간이 측정되었다. 검색시간은 각 시스템 별 검색 시간 측정은 표 2와 같이 각각의 키워드로 검색했을 때 걸린 시간의 평균을 구하였다.

주제어만을 이용하여 검색을 하는 본 연구의 검색 시스템을 실제 서비스되고 있는 블로그 검색 시스템과의 성능을 비교해 본 결과, 본 논문에서 구현한 주제어에 더 높은 가중치를 부여하는 검색 시스템이 블로그 검색에 더 효과적임이 입증되었다.

Ⅵ. 결 론

본 논문에서는 실제로 주제어를 추출하여 검색 랭킹에 중요도를 부여하는 방식의 블로그 검색 시스템을 구현하였다. 이 시스템은 블로그 글의 제목이나 블로그 태그, 굵은 폰트, 앵커텍스트나 소제목 같이 중요도가 높은 위치에서 주제어를 추출하여 본문에서 추출된 색인어보다 더 높은 가중치를 부여하여 랭킹에 반영하였고, 다른 시스템과 비교 평가해 본 결과, 재현율이 가장 우수하다는 결론을 내릴 수 있었다.

추후 연구로, 기존에 제안된 블로그 권위 랭킹 알고리즘을 이용하여 블로그 랭킹에 사용자 반응이나 블로그 권위 요소와 글의 최신성(freshness)에 대한 요소를 랭킹 함수에 추가하고 주제어 추출 방법도 효율적으로 개선하는 것에 대한 연구를 진행할 계획이다.

참고문헌

[1] R. Kumar, P. Novak, S. Raghavan and A. Tomkins, "Structure and evolution of the Blogspace," Communication of the ACM, Vol. 47, No. 12, pp. 35-39, December 2004.

[2] Q. Mei, X. Ling, M. Wondra, H. Su and C. Zhai, "Topic sentiment mixture: modeling facts and opinions in weblogs," Proceedings of the 16th international conference on World Wide Web, pp.171-180, Banff, Alberta, Canada, May 2007.

[3] K. Fujimura and N. Tanimoto, "The EigenRumor Algorithm for Calculating Contributions in Cyberspace Communities," Trusting Agents, LNAI 3577, pp. 59 - 74, 2005.

[4] Taher H. Haveliwala, "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 4, pp. 784-796, 2003.

[5] N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Identifying the influential bloggers in a community," WSDM'08, pp. 207-218, 2008

[6] A. Kritikopoulos, M. Sideri and I. Varlamis, "BlogRank: Ranking Weblogs Based on Connectivity and Similarity Features," AAA-IDEA'06, Vol. 198, No. 8, Pisa, Italy, 2006.

[7] M. A. Tayebi, S. M. Hashemi and A. Mohades, "B2Rank: An Algorithm for Ranking Blogs BasedN on Behavioral Features," Web Intelligence, pp.104-107, 2007.

[8] 정윤재, 이동만, "블로그 공간 상 에고센트릭 검색의 검색 시간 향상을 위한 권위 추정 방법," 한국정보과학회 추계 학술발표논문집, 제 13권, 제 2호, 2006년 11월.

[9] 황원석, 도영주, 배덕호, 김상욱, "블로그 환경을 위한 포스트 랭킹 알고리즘," 한국정보과학회 종합학술대회 논문집, 제 35권, 제 1호, 189-193쪽, 2008년 6월.

[10] 올블로그, *http://www.allblog.net*

[11] 블로그얌, *http://www.blogyam.co.kr*

[12] 나루 블로그 검색, *http://www.naaroo.com*

[13] 윤은일, 신현일, 류근호, "중요 여행 정보를 찾기 위한 지능 검색 시스템," 한국컴퓨터정보학회논문지, 제 14권, 제 11호, 113-122쪽, 2009년 11월.

[14] 문유진, "정보 검색을 위한 숫자의 해석에 관한 구문적, 의미적 판별 기법," 한국컴퓨터정보학회논문지, 제 14권 제 8호, 55-71쪽, 2009년 8월.

[15] 장성호, 강승식, "주제어 기반 문서 클러스터링 알고리즘," 한국정보과학회 봄 학술발표논문집, 제 29권, 제 2호, 469-471쪽, 2002년 4월.

[16] B. Adida, "hGRDDL: Bridging microformats and RDFa," Web Semantics Sci Serv Agents on WWW, Vol. 6, No. 1, pp. 54-60, February 2008.

[17] 선복근, 위다현, 한광록, "OWL 온톨로지를 기반으로 하는 논문 검색 시스템에 관한 연구," 한국컴퓨터정보학회논문지, 제 14권, 제 2호, 169-180쪽, 2009년 2월

[18] 이재윤, "피벗 역문헌빈도 가중치 기법에 대한 연구," 한국정보관리학회 정보관리학회지, 제 20권, 제 4호, 233-248쪽, 2003년 12월.

[19] 이글루스, *http://www.egloos.com*

저 자 소 개

신 현 일

2009. 8 :
충북대학교 컴퓨터공학 학사.
2009. 9 - 현재 :
충북대학교 컴퓨터공학 석사
관심분야 : 데이터마이닝, 정보검색, 데이터베이스

윤 은 일

1997: 고려대학교 이학석사.
1997 - 2006 :
한국통신 멀티미디어연구소 전임/선임 연구원.
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

류 근 호

1976 : 숭실대학교학교 공학사.
1997 : 연세대학교 공학석사.
1980 - 1983 :
한국전자통신연구원 연구원.
1983 - 1986 :
한국방송통신대학교 조교수.
1988: 연세대학교 공학박사
1986 - 현재:
충북대학교 전자정보대학 컴퓨터전공 교수
관심분야 : 데이터베이스, 데이터마이닝, 바이오인포매틱스